



agriculture

Special Issue Reprint

Digital Innovations in Agriculture

Series II

Edited by
Gniewko Niedbała, Sebastian Kujawa, Magdalena Piekutowska
and Tomasz Wojciechowski

mdpi.com/journal/agriculture



**Digital Innovations in
Agriculture—Series II**

Digital Innovations in Agriculture—Series II

Guest Editors

Gniewko Niedbała

Sebastian Kujawa

Magdalena Piekutowska

Tomasz Wojciechowski



Basel • Beijing • Wuhan • Barcelona • Belgrade • Novi Sad • Cluj • Manchester

Guest Editors

Gniewko Niedbała
Poznań University of Life
Sciences
Poznań
Poland

Sebastian Kujawa
Poznań University of Life
Sciences
Poznań
Poland

Magdalena Piekutowska
Pomeranian University in
Słupsk
Słupsk
Poland

Tomasz Wojciechowski
Poznań University of Life
Sciences
Poznań
Poland

Editorial Office

MDPI AG
Grosspeteranlage 5
4052 Basel, Switzerland

This is a reprint of the Special Issue, published open access by the journal *Agriculture* (ISSN 2077-0472), freely accessible at: https://www.mdpi.com/journal/agriculture/special_issues/7M19PZA2KY.

For citation purposes, cite each article independently as indicated on the article page online and as indicated below:

Lastname, A.A.; Lastname, B.B. Article Title. <i>Journal Name</i> Year , Volume Number, Page Range.
--

ISBN 978-3-7258-2527-1 (Hbk)

ISBN 978-3-7258-2528-8 (PDF)

<https://doi.org/10.3390/books978-3-7258-2528-8>

© 2024 by the authors. Articles in this book are Open Access and distributed under the Creative Commons Attribution (CC BY) license. The book as a whole is distributed by MDPI under the terms and conditions of the Creative Commons Attribution-NonCommercial-NoDerivs (CC BY-NC-ND) license (<https://creativecommons.org/licenses/by-nc-nd/4.0/>).

Contents

Preface	vii
Gniewko Niedbała, Sebastian Kujawa, Magdalena Piekutowska and Tomasz Wojciechowski Exploring Digital Innovations in Agriculture: A Pathway to Sustainable Food Production and Resource Management Reprinted from: <i>Agriculture</i> 2024, 14, 1630, https://doi.org/10.3390/agriculture14091630	1
Xinyuan He, Zhixiong Zeng, Yanhua Liu, Enli Lyu, Jingjing Xia, Feiren Wang and Yizhi Luo An Internet of Things-Based Cluster System for Monitoring Lactating Sows' Feed and Water Intake Reprinted from: <i>Agriculture</i> 2024, 14, 848, https://doi.org/10.3390/agriculture14060848	5
Ana Isabel Caicedo Camayo, Martin Alexander Chaves Muñoz and Juan Carlos Corrales ApIsoT: An IoT Function Aggregation Mechanism for Detecting <i>Varroa</i> Infestation in <i>Apis mellifera</i> Species Reprinted from: <i>Agriculture</i> 2024, 14, 846, https://doi.org/10.3390/agriculture14060846	25
Senlin Guan, Kimiyasu Takahashi, Shunichiro Watanabe and Katsunori Tanaka Unmanned Aerial Vehicle-Based Techniques for Monitoring and Prevention of Invasive Apple Snails (<i>Pomacea canaliculata</i>) in Rice Paddy Fields Reprinted from: <i>Agriculture</i> 2024, 14, 299, https://doi.org/10.3390/agriculture14020299	48
Rui Zhang, Mingwei Yao, Zijie Qiu, Lizhuo Zhang, Wei Li and Yue Shen Wheat Teacher: A One-Stage Anchor-Based Semi-Supervised Wheat Head Detector Utilizing Pseudo-Labeling and Consistency Regularization Methods Reprinted from: <i>Agriculture</i> 2024, 14, 327, https://doi.org/10.3390/agriculture14020327	64
Alper Taner, Mahtem Teweldemedhin Mengstu, Kemal Çağatay Selvi, Hüseyin Duran, İbrahim Gür and Nicoleta Ungureanu Apple Varieties Classification Using Deep Features and Machine Learning Reprinted from: <i>Agriculture</i> 2024, 14, 252, https://doi.org/10.3390/agriculture14020252	85
Berk Benlioğlu, Fatih Demirel, Aras Türkoğlu, Kamil Haliloğlu, Hamdi Özaktan, Sebastian Kujawa, et al. Insights into Drought Tolerance of Tetraploid Wheat Genotypes in the Germination Stage Using Machine Learning Algorithms Reprinted from: <i>Agriculture</i> 2024, 14, 206, https://doi.org/10.3390/agriculture14020206	99
Feng Xiao, Haibin Wang, Yueqin Xu and Zhen Shi A Lightweight Detection Method for Blueberry Fruit Maturity Based on an Improved YOLOv5 Algorithm Reprinted from: <i>Agriculture</i> 2024, 14, 36, https://doi.org/10.3390/agriculture14010036	117
Jarosław Kurek, Gniewko Niedbała, Bartosz Świdorski, Tomasz Wojciechowski, Izabella Antoniuk, Magdalena Piekutowska, et al. Prediction of Potato (<i>Solanum tuberosum</i> L.) Yield Based on Machine Learning Methods Reprinted from: <i>Agriculture</i> 2023, 13, 2259, https://doi.org/10.3390/agriculture13122259	136
Myung Hwan Na, Wan Hyun Cho, Sang Kyoong Kim and In Seop Na The Development of a Weight Prediction System for Pigs Using Raspberry Pi Reprinted from: <i>Agriculture</i> 2023, 13, 2027, https://doi.org/10.3390/agriculture13102027	161

Alexis Barrios-Ulloa, Alejandro Cama-Pinto, Emiro De-la-Hoz-Franco, Raúl Ramírez-Velarde and Dora Cama-Pinto Modeling of Path Loss for Radio Wave Propagation in Wireless Sensor Networks in Cassava Crops Using Machine Learning Reprinted from: <i>Agriculture</i> 2023 , <i>13</i> , 2046, https://doi.org/10.3390/agriculture13112046	173
Siti Nurul Afiah Mohd Johari, Siti Khairunniza-Bejo, Abdul Rashid Mohamed Shariff, Nur Azuan Husin, Mohamed Mazmira Mohd Masri and Noorhazwani Kamarudin Detection of Bagworm Infestation Area in Oil Palm Plantation Based on UAV Remote Sensing Using Machine Learning Approach Reprinted from: <i>Agriculture</i> 2023 , <i>13</i> , 1886, https://doi.org/10.3390/agriculture13101886	188
Ali Siddique, Sun Jingqi, Kung Jui Hou, Mang I. Vai, Sio Hang Pun and Muhammad Azhar Iqbal SpikoPoniC: A Low-Cost Spiking Neuromorphic Computer for Smart Aquaponics Reprinted from: <i>Agriculture</i> 2023 , <i>13</i> , 2057, https://doi.org/10.3390/agriculture13112057	209
Anant Sujatanagarjuna, Shohreh Kia, Dominique Briechle and Benjamin Leiding MushR: A Smart, Automated, and Scalable Indoor Harvesting System for Gourmet Mushrooms Reprinted from: <i>Agriculture</i> 2023 , <i>13</i> , 1533, https://doi.org/10.3390/agriculture13081533	234
Daqing Wu, Rong Yan, Hongtao Jin and Fengmao Cai An Adaptive Nutcracker Optimization Approach for Distribution of Fresh Agricultural Products with Dynamic Demands Reprinted from: <i>Agriculture</i> 2023 , <i>13</i> , 1430, https://doi.org/10.3390/agriculture13071430	250

Preface

The Special Issue “Digital Innovations in Agriculture-Series II” delves into the rapidly evolving landscape of digital technologies in agriculture, spotlighting the cutting-edge advancements that are transforming traditional farming practices. This series showcases a wide array of topics including precision agriculture, Internet of Things (IoT), artificial intelligence (AI), big data analytics, and automated systems, all designed to enhance the efficiency, sustainability, and resilience of agricultural operations.

The second series of this Special Issue emphasizes the role of these technologies in tackling global challenges such as food security, climate change, and resource optimization. AI-driven decision-support systems, for instance, enable farmers to make real-time, data-informed choices, improving crop yields and minimizing environmental impact. IoT technologies facilitate seamless monitoring of soil conditions, weather patterns, and crop health, thus enhancing resource management and reducing waste. In addition, machine learning models and remote sensing technologies allow for early disease detection, predictive analytics, and precision pest control, significantly reducing reliance on chemical inputs.

This reprint brings together interdisciplinary research efforts that aim to bridge the gap between digital innovation and practical farming solutions. It offers a platform for discussing both theoretical advancements and real-world applications, making it a valuable resource for researchers, agronomists, and industry professionals seeking to understand and implement next-generation agricultural technologies. By addressing key areas such as automation, sensor technologies, and the role of robotics in farming, the series highlights the potential for digital innovations to revolutionize global agriculture and contribute to more sustainable food systems.

Gniewko Niedbała, Sebastian Kujawa, Magdalena Piekutowska, and Tomasz Wojciechowski

Guest Editors



Exploring Digital Innovations in Agriculture: A Pathway to Sustainable Food Production and Resource Management

Gniewko Niedbala ^{1,*}, Sebastian Kujawa ¹, Magdalena Piekutowska ^{2,*} and Tomasz Wojciechowski ¹

¹ Department of Biosystems Engineering, Faculty of Environmental and Mechanical Engineering, Poznań University of Life Sciences, Wojska Polskiego 50, 60-627 Poznań, Poland; sebastian.kujawa@up.poznan.pl (S.K.); tomasz.wojciechowski@up.poznan.pl (T.W.)

² Department of Botany and Plant Protection, Institute of Biology, Pomeranian University in Słupsk, 22b Partyzantów St., 76-200 Słupsk, Poland

* Correspondence: gniewko.niedbala@up.poznan.pl (G.N.); magdalena.piekutowska@upsl.edu.pl (M.P.)

Today's agriculture faces numerous challenges due to climate change, a growing population and the need to increase food productivity. In response to these issues, digital innovation is becoming an essential tool that can significantly improve the efficiency of the agricultural sector. The digitization of agriculture, which includes technologies such as the Internet of Things (IoT), machine learning, drones and advanced monitoring systems, offers new opportunities that can help improve production processes, resource management and environmental protection.

This SI titled "Digital Innovations in Agriculture-Series II" presents a collection of studies that illustrate the diversity of applications of innovative technologies in agriculture. The articles in this Special Issue explore such areas as the accurate monitoring of plant and animal health, the optimization of production processes and the development of intelligent management systems. Through the use of modern analytical and diagnostic tools, these studies show how innovation can lead to a more informed and efficient use of resources [1,2].

The opportunities created by these technologies not only increase production efficiency, but also contribute to biodiversity conservation and ecosystem sustainability. In addition, data analysis methods and artificial intelligence-based approaches enable better forecasting and risk management, which is critical in the face of a changing climate and increasing extreme events. This Special Issue represents an important step toward understanding the impact of digital technologies on the future of agriculture and agri-food sustainability [3–5].

We hope that the research collected in it will inspire both researchers and industry practitioners to further explore the potential of innovative solutions in agriculture and their practical application in daily operations.

The perishable nature of commodities in the agri-food supply chain poses challenges for producers and marketers. Using an ecological routing model for the distribution of fresh agricultural products using an adaptive hybrid nutcracker optimization algorithm (AH-NOA) allows for increased population diversity. Thanks to k-means clustering, the algorithm effectively overcomes local optima, leading to significant reductions in costs and CO₂ emissions. Studies have indicated that AH-NOA significantly improves global search efficiency, making this method promising. In addition, MushR's gourmet mushroom breeding automation system achieved 91.7% accuracy in identifying ready-to-harvest mushrooms using Raspberry Pi-based harvesting mechanics. This approach shows how technologies can revolutionize the harvesting process in food production [6,7].

Detecting oil palm pests using UAVs provides a modern solution for agriculture. The use of drones along with a multispectral camera allows for the quick and accurate analysis of infestation rates. Studies have shown a 100% F1 score in classifying healthy and low levels of infestation, demonstrating the effectiveness of this method. Rapid response

Citation: Niedbala, G.; Kujawa, S.; Piekutowska, M.; Wojciechowski, T. Exploring Digital Innovations in Agriculture: A Pathway to Sustainable Food Production and Resource Management. *Agriculture* **2024**, *14*, 1630. <https://doi.org/10.3390/agriculture14091630>

Received: 14 September 2024

Accepted: 17 September 2024

Published: 17 September 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

through the use of UAVs can save time and resources compared to traditional detection methods. This type of technology is key in crop management, increasing the efficiency of palm oil monitoring. The results show the importance of introducing innovative solutions in agricultural production [8].

The models used to forecast potato yields in Poland showed significant potential by analyzing an integrated data set from 36 fields. Three models were developed: non-satellite, satellite and hybrid, and the best result emerged from the hybrid model, achieving a MAPE of 5.85%. Advanced data analysis techniques, such as PCA and outlier detection, helped improve forecasting performance, and the hybrid model combining data from all sources yielded the best results. This approach to yield analysis can significantly improve forecasting accuracy in agriculture. The results of this research suggest that machine learning plays a key role in improving crop management. Ultimately, innovative yield forecasting tools have the potential to optimize agricultural production [9].

A detection system for recognizing blueberry fruit maturity, based on the improved YOLOv5 algorithm, achieved impressive results, such as an average recall of 92.0% and an average precision of 91.5%. The use of the ShuffleNet module and Convolutional Blocks of Attention (CBAM) contributed to better feature fusion, and the system's speed is 67.1 frames per second. The method, with its lower computational requirements, is more suitable for use in edge devices. A comparison with other algorithms, such as SSD and Faster R-CNN, showed better results with a lighter model weight. This method has potential for applications in intelligent fruit detection systems in orchards, which can support modern agriculture. These achievements show how technology can revolutionize fruit picking [10].

Evaluating the differential revision responses of multiple wheat genotypes to water stress reveals significant differences in tolerance and plant growth. Studies that used polyethylene glycol at different concentrations applied machine learning techniques to effectively predict key growth parameters. The best results in predicting water absorption and germination rates were obtained using the elastic-net model. In addition, an analysis of the performance of classification techniques showed that a Gaussian process classifier was best for estimating root length. The results of this research are crucial for wheat genetics and breeding in the face of a changing climate. The application of new analytical techniques indicates the possibility of crop improvement under harsh conditions [11].

The use of computer vision technology to classify apple varieties has yielded exceptional results, achieving an accuracy as high as 97.48% using the DenseNet201 model. The implementation of CNN architectures and transfer learning in fruit classification confirmed the effectiveness of this method. Further analysis of the deep features extracted from the model allowed the use of traditional classifiers such as SVM and MLP, which contributed to an even higher accuracy of 99.77%. This study also considered the impact of dimensionality reduction on classification performance. These achievements underscore the importance of artificial intelligence in improving classification quality in an agricultural product. The findings point to the need for further research in apple variety diversity, which could enhance analytical capabilities [12].

Wheat Teacher is a wheat ear detector that uses semi-supervised methods and employs pseudo-tagging and consistency regularization. Experiments conducted on a set of GWHD2021 showed that Wheat Teacher achieved a mAP_{0.5} score of 92.8%, using only 20% of the labeled data. Such a result outperformed the results of two full-surveillance-based detection models that were trained on 100% of the labeled data, with a difference of no more than 1%. In addition, Wheat Teacher improved mAP_{0.5} by 2.1%, achieving 37.7% across different proportions of labeled data usage. This research underscores Wheat Teacher's significant potential in effective wheat ear detection, which can contribute to better breeding strategies. The system indicates the potential for applications of semi-supervised technology in the context of increasing efficiency in wheat crop management [13].

The introduction of a mobile pig weight prediction system, based on Raspberry Pi and 3D camera data analysis, made it possible to automate the weighing process. The

system achieved an RMSE of about 10.702, confirming its high precision. The use of 3D technology contributes to reducing labor costs in animal husbandry. With effective weight measurement, this study illustrates the innovation of measurement methods in animal production. The results of this research are applicable to increasing productivity in animal husbandry. Such systems can significantly improve production efficiency in the animal industry [14].

In addition, drone technologies in insecticide application in young rice orchards show the importance of precision agriculture. The introduction of a prescription map allows chemicals to be concentrated in areas with the highest loss, reducing their use at the same time. Studies have shown that the use of drones increases the efficiency of crop protection, leading to a lower percentage of rice damage compared to control areas. This approach contributes to the development of modern crop protection methods. The results of this research are a step toward more sustainable agricultural practices that can benefit the environment. The use of drones in crop protection opens up new opportunities for more efficient agricultural production [15].

Studies on radio signal propagation in sensor networks have shown that different models do not respond adequately to signal attenuation in dense vegetation. Comparative analyses with actual measurements in a tapioca field in Colombia have shown significant differences in network design. Machine learning techniques, including random forests and K-NN, significantly improved the prediction of signal attenuation. The final findings may be useful in planning new wireless networks in agriculture. This study is a step toward improving communication technologies in the agricultural sector. The development of such technologies has a major impact on the efficiency of agricultural resource management [16].

The use of a neuromorphic computer to estimate the weight and length of fish in smart aquaponic systems has demonstrated the ability to classify more than 84 million samples in just one second. The SpikoPoniC system, based on this research, has a 3369-fold acceleration compared to traditional computers. This enables the intelligent monitoring of aquaponics, which is crucial for future commercial solutions. This study highlights the importance of hardware optimization in aquaculture applications. Thus, it becomes clear how modern technologies can revolutionize the monitoring of aquatic resources. Innovative approaches to fish monitoring have great potential in aquaponics systems [17].

Faced with declining populations of *Apis mellifera* bees, which are crucial for pollination, the authors of [18] present an IoT system for beekeepers in developing countries. The use of low-cost devices to monitor hive parameters, such as temperature and humidity, allows the early detection of problems. The efficiency of the system is enhanced by processing data using an advanced machine learning model. This study also takes into account the analysis of energy consumption and network traffic. Ultimately, the proposed solution improves the accuracy of mite detection. Conclusions address the role of technology in pollinator protection and food security [18].

A system for monitoring the feed and water intake of sows provides information on their health and performance in real time. The use of open technologies and an electronic feeder consisting of a data processing program based on Spark and Flink brings significant benefits. The system allows 20,000 sows to be monitored simultaneously, which improves the management of their breeding. Studies have confirmed that with optimization, the system achieves a TPS of 6399 pcs/s for 10,000 sows. The results indicate the technology's potential in intensive agriculture. Such solutions can significantly contribute to improving efficiency in livestock production [19].

Today's agriculture is facing challenges from a growing population and climate change, requiring increased efficiency in food production. The digitization of agriculture, through the use of technologies such as IoT, drones and machine learning, offers solutions to increase the efficiency and sustainability of the agribusiness sector. The research included in "Digital Innovations in Agriculture-Series II" demonstrates the wide range of applications of innovative techniques that can lead to a more sustainable use of resources and better crop and livestock management. Modern analytical and predictive technologies play a key

role in protecting the environment and increasing production efficiency, which is crucial in the context of a changing climate. The development of these technologies indicates the significant impact of digitalization on the future of agriculture and the potential in optimizing production processes.

Author Contributions: All authors (G.N., S.K., M.P. and T.W.) contributed equally to the development of this Editorial. All authors have read and agreed to the published version of the manuscript.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Sachithra, V.; Subhashini, L.D.C.S. How Artificial Intelligence Uses to Achieve the Agriculture Sustainability: Systematic Review. *Artif. Intell. Agric.* **2023**, *8*, 46–59. [CrossRef]
2. Falola, P.B.; Adeniyi, A.E.; Madamidola, O.A.; Awotunde, J.B.; Olukiran, O.A.; Akinola, S.O. Artificial Intelligence in Agriculture: The Potential for Efficiency and Sustainability, with Ethical Considerations. In *Exploring Ethical Dimensions of Environmental Sustainability and Use of AI*; Kannan, H., Rodriguez, R., Paprika, Z., Ade-Ibijola, A., Eds.; IGI Global: Hershey, PA, USA, 2024; pp. 307–329. [CrossRef]
3. Niedbała, G.; Nowakowski, K.; Rudowicz-Nawrocka, J.; Piekutowska, M.; Weres, J.; Tomczak, R.J.; Tyksiński, T.; Álvarez Pinto, A. Multicriteria Prediction and Simulation of Winter Wheat Yield Using Extended Qualitative and Quantitative Data Based on Artificial Neural Networks. *Appl. Sci.* **2019**, *9*, 2773. [CrossRef]
4. Niedbała, G.; Kurasiak-Popowska, D.; Stuper-Szablewska, K.; Nawracała, J. Application of Artificial Neural Networks to Analyze the Concentration of Ferulic Acid, Deoxynivalenol, and Nivalenol in Winter Wheat Grain. *Agriculture* **2020**, *10*, 127. [CrossRef]
5. Cirocki, R.; Gołębiwska, B. Changes in the Profitability of Production of Industrial Potatoes in Poland—A Case Study. *Ann. Polish Assoc. Agric. Agribus. Econ.* **2019**, *21*, 19–28. [CrossRef]
6. Wu, D.; Yan, R.; Jin, H.; Cai, F. An Adaptive Nutcracker Optimization Approach for Distribution of Fresh Agricultural Products with Dynamic Demands. *Agriculture* **2023**, *13*, 1430. [CrossRef]
7. Sujatanagarjuna, A.; Kia, S.; Briechele, D.F.; Leiding, B. MushR: A Smart, Automated, and Scalable Indoor Harvesting System for Gourmet Mushrooms. *Agriculture* **2023**, *13*, 1533. [CrossRef]
8. Johari, S.N.A.M.; Khairunniza-Bejo, S.; Shariff, A.R.M.; Husin, N.A.; Masri, M.M.M.; Kamarudin, N. Detection of Bagworm Infestation Area in Oil Palm Plantation Based on UAV Remote Sensing Using Machine Learning Approach. *Agriculture* **2023**, *13*, 1886. [CrossRef]
9. Kurek, J.; Niedbała, G.; Wojciechowski, T.; Świdarski, B.; Antoniuk, I.; Piekutowska, M.; Kruk, M.; Bobran, K. Prediction of Potato (*Solanum tuberosum* L.) Yield Based on Machine Learning Methods. *Agriculture* **2023**, *13*, 2259. [CrossRef]
10. Xiao, F.; Wang, H.; Xu, Y.; Shi, Z. A Lightweight Detection Method for Blueberry Fruit Maturity Based on an Improved YOLOv5 Algorithm. *Agriculture* **2023**, *14*, 36. [CrossRef]
11. Benlioğlu, B.; Demirel, F.; Türkoğlu, A.; Haliloğlu, K.; Özaktan, H.; Kujawa, S.; Piekutowska, M.; Wojciechowski, T.; Niedbała, G. Insights into Drought Tolerance of Tetraploid Wheat Genotypes in the Germination Stage Using Machine Learning Algorithms. *Agriculture* **2024**, *14*, 206. [CrossRef]
12. Taner, A.; Mengstu, M.T.; Selvi, K.Ç.; Duran, H.; Gür, İ.; Ungureanu, N. Apple Varieties Classification Using Deep Features and Machine Learning. *Agriculture* **2024**, *14*, 252. [CrossRef]
13. Zhang, R.; Yao, M.; Qiu, Z.; Zhang, L.; Li, W.; Shen, Y. Wheat Teacher: A One-Stage Anchor-Based Semi-Supervised Wheat Head Detector Utilizing Pseudo-Labeling and Consistency Regularization Methods. *Agriculture* **2024**, *14*, 327. [CrossRef]
14. Na, M.H.; Cho, W.H.; Kim, S.K.; Na, I.S. The Development of a Weight Prediction System for Pigs Using Raspberry Pi. *Agriculture* **2023**, *13*, 2027. [CrossRef]
15. Guan, S.; Takahashi, K.; Watanabe, S.; Tanaka, K. Unmanned Aerial Vehicle-Based Techniques for Monitoring and Prevention of Invasive Apple Snails (*Pomacea canaliculata*) in Rice Paddy Fields. *Agriculture* **2024**, *14*, 299. [CrossRef]
16. Barrios-Ulloa, A.; Cama-Pinto, A.; De-la-Hoz-Franco, E.; Ramirez-Velarde, R.; Cama-Pinto, D. Modeling of Path Loss for Radio Wave Propagation in Wireless Sensor Networks in Cassava Crops Using Machine Learning. *Agriculture* **2023**, *13*, 2046. [CrossRef]
17. Siddique, A.; Sun, J.; Hou, K.J.; Vai, M.I.; Pun, S.H.; Iqbal, M.A. SpikoPoniC: A Low-Cost Spiking Neuromorphic Computer for Smart Aquaponics. *Agriculture* **2023**, *13*, 2057. [CrossRef]
18. Camayo, A.I.C.; Muñoz, M.A.C.; Corrales, J.C. ApIsoT: An IoT Function Aggregation Mechanism for Detecting Varroa Infestation in *Apis mellifera* Species. *Agriculture* **2024**, *14*, 846. [CrossRef]
19. He, X.; Zeng, Z.; Liu, Y.; Lyu, E.; Xia, J.; Wang, F.; Luo, Y. An Internet of Things-Based Cluster System for Monitoring Lactating Sows' Feed and Water Intake. *Agriculture* **2024**, *14*, 848. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.



Article

An Internet of Things-Based Cluster System for Monitoring Lactating Sows' Feed and Water Intake

Xinyuan He ¹, Zhixiong Zeng ^{1,2}, Yanhua Liu ¹, Enli Lyu ^{1,3,*}, Jingjing Xia ^{1,4}, Feiren Wang ^{4,5} and Yizhi Luo ^{3,6}

¹ College of Engineering, South China Agricultural University, Guangzhou 510642, China; 20231168006@stu.scau.edu.cn (X.H.); zhixzeng@scau.edu.cn (Z.Z.); cynthial@scau.edu.cn (Y.L.); xiajingjing@gdmec.edu.cn (J.X.)

² Maoming Branch, Guangdong Laboratory for Lingnan Modern Agriculture, Guangzhou 510642, China

³ State Key Laboratory of Swine and Poultry Breeding Industry, Guangzhou 510645, China; luoyizhi@gdaas.cn

⁴ Schools of Automobile, Guangdong Mechanical and Electrical Polytechnic, Guangzhou 510550, China; 2020010018@gdmec.edu.cn

⁵ Guangzhou Jiaen Technology Co., Ltd., Guangzhou 510642, China

⁶ Institute of Facility Agriculture, Guangdong Academy of Agricultural Sciences, Guangzhou 510640, China

* Correspondence: enlily@scau.edu.cn; Tel.: +86-20-8528-2860

Abstract: Acquiring real-time feeding information for monitoring lactating sows and their feeding requirements is a challenging task. Real-time data represent an important input for numerous tasks, such as disease monitoring, nutritional regulation, and feeding modeling. However, concurrently monitoring large numbers of sows and processing the real-time information for modeling is challenging using existing platforms. In this paper, we describe the design and development of a system that monitors and processes sows' feed and water consumption in real time. The system was custom-developed using open-source networking technologies. The system consists of three components: an electronic sow feeder connected to a central controller via a CAN network, an MQTT service cluster, and a data processing program. The MQTT service cluster uses Netty to develop a single service node, and it uses Zookeeper and Redis to complete node registration, discovery, and scheduling. The data processing program is based on Spark and Flink. We conducted comparative testing of three common codecs (Java Serializer, Marshalling, and Protostuff) to further speed up data transmission. The results of the experiment show that, with three service nodes, the system can concurrently monitor up to 20,000 sows. Moreover, the system achieves optimal performance when monitoring 10,000 sows at the same time, with a TPS of 6399 pcs/s and an RT of 643 ms.

Keywords: lactating sow; intelligent feeding; cluster system; IoT platform

Citation: He, X.; Zeng, Z.; Liu, Y.; Lyu, E.; Xia, J.; Wang, F.; Luo, Y. An Internet of Things-Based Cluster System for Monitoring Lactating Sows' Feed and Water Intake. *Agriculture* **2024**, *14*, 848. <https://doi.org/10.3390/agriculture14060848>

Academic Editors:
Gniewko Niedbala,
Sebastian Kujawa, Tomasz
Wojciechowski and
Magdalena Piekutowska

Received: 15 April 2024
Revised: 24 May 2024
Accepted: 27 May 2024
Published: 28 May 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The Internet of Things (IoT) is applied in various industries, including smart manufacturing [1], smart cities [2], smart healthcare [3], precision agriculture [4], and smart livestock farming [5], through various applications involving machine-to-machine and machine-to-human interactions [6]. Intelligent machines continuously send data to cloud servers via the internet. The data are analyzed and presented as per the requirements of the end-users [7]. The IoT has many applications in pig farming. Monitoring the parameters related to lactating sows is one of the most vital aspects [8,9]. The feed intake and water consumption of lactating sows are required in real time for precision feeding in pig farms. These data are vital for improving feeding practices and enabling the timely prediction of diseases [10,11].

Electronic sow feeders (ESFs) are commonly used to record the feed information of lactating sows [12]. ESF systems are used during the lactation period to control the delivery of feed to individual sows in a farrowing house. Depending on the system, ESFs may also record data for individual feed intake and the timing of eating events, and the data can be

rapidly transmitted back to cloud servers through the network [13]. With the help of ESFs and information technology, modern precision feeding systems for lactating sows have demonstrated their ability to meet individual requirements more efficiently [14]. Ma et al. devised an intelligent feeding equipment and network service platform, which can provide a convenient and intelligent feeding method as well as multiple areas of pig management. The platform realizes the recording of the daily feeding amount of pigs from multiple areas [15]. Chen et al. developed a precision feeding system consisting of intelligent feeders, environmental monitoring devices, a wireless data transmission module, and a remote monitoring terminal. The Alibaba Cloud server was connected using a long-range radio (LoRa) module and a general packet radio service (GPRS) module to achieve the long-distance transmission of feed and water intake data [16]. Although the above systems have realized the collection and management of pig feeding information, there is little mention of the number of devices connected to the systems and the implementation details of data transmission.

With the intelligent development of pig farming devices, higher demands have been placed on the stability and concurrency of pig farm IoT systems [17,18]. There are currently some commercial IoT platforms, such as ThingSpeak cloud platform, Microsoft Azure, and other platforms, that can support edge device access [19]. However, in terms of actual use on pig farms, commercial IoT platforms face issues such as cumbersome configuration, difficulty in flexible customization, and high costs [20]. With the application of technologies such as clusters, distributed systems, and microservices in animal husbandry, centralized servers are shifting towards distributed servers [21]. This shift provides new solutions for the development of high-performance IoT systems. A previous study [22] used microservices and cloud paradigms to implement a highly scalable livestock IoT platform. The microservice architecture of a distributed IoT system discussed in detail in [23] consists of a group of microservices that communicate with each other synchronously or asynchronously. Another study [24] involved the design of a sensor network based on a microkernel operating system. Based on the idea of modular design, the functional modules and control modules were organically combined so as to realize the monitoring and collection of farmland and livestock management information.

The aim of this study was to design an Internet of Things-based cluster system for lactating sows utilizing electronic sow feeders, a message queue telemetry transport (MQTT) service cluster, and data processing programs to collect and monitor the feed and water intake of lactating sows in real time. The MQTT server developed based on Netty provides connection services for ESFs. In order to improve the concurrency and stability of the system, an MQTT server cluster suitable for real-time applications in pig farms was built based on Zookeeper. A scheduling server was added to ensure the reasonable distribution of MQTT service nodes. Moreover, this system also includes a data processing platform that can monitor and record feed intake, water consumption, and ESF operating status. Our main objectives were (1) to design and develop a stable ESF and central controller for monitoring lactating sows, (2) to develop a high-concurrency MQTT server cluster to collect real-time values of central controllers attached to the server node, (3) to develop a data processing platform for recording and tallying the collected values at the pig farm, and (4) to utilize a user-friendly interactive interface for displaying real-time information. The present study included the design, implementation, and verification of the system.

The details of this study are presented in Section 4. Section 1 provides a brief description of related work reported in the literature and outlines the objectives for the present study. Section 2 offers technical details regarding the development of the system, including the description of the system architecture, a brief overview of the design of the ESF and central controller, the MQTT server cluster, the data processing program, and the data transmission security. The experimental results, accompanied by case study data, are discussed in Section 3. Section 5 provides a summary, along with conclusions, and outlines the future scope of the study.

2. Materials and Methods

2.1. The Architecture of the System

The system's architecture (Figure 1) includes three significant components: an electronic sow feeder (Figure 1a) connected to a central controller (Figure 1b) via a CAN network, an MQTT service cluster, and a data processing program. An electronic sow feeder was designed to monitor the feeding status of lactating sows. It was installed above each stall in the experiment and connected to a touch rod (which initiated feed and water discharge upon the sow's touch), a feed discharge pipe, and a water discharge pipe (Figure 1d) inside the feed trough. A central controller controls all electronic sow feeders within a unit through the controller area network (CAN). This is because the network signal is poor in the remote environment of the pig farm. The use of CAN lines can ensure the stable connection of equipment. In addition, we developed a PDA to set parameters for the electronic feeder through a QR code (Figure 1c). The electronic sow feeder records the daily feed and water intake, and the central controller transmits the data to the MQTT service cluster and data processing program running in the cloud through the MQTT protocol.

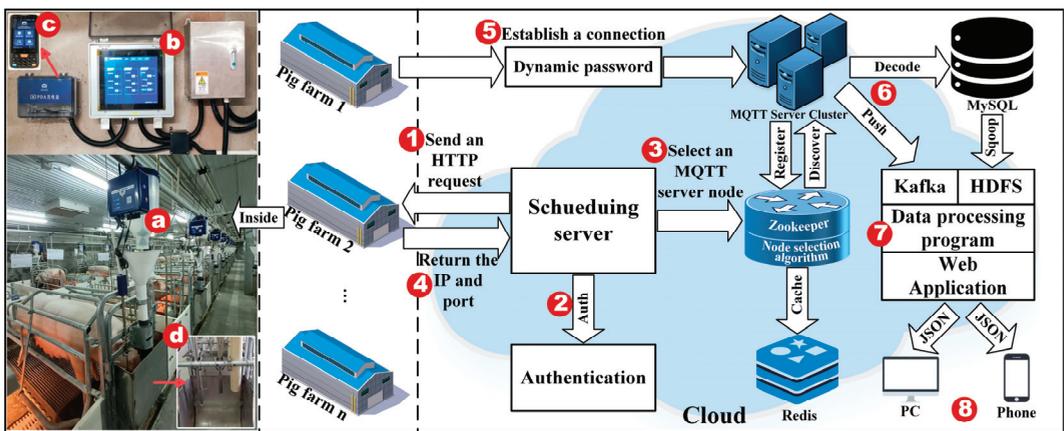


Figure 1. Architecture of the IoT-based intelligent feeding equipment and cloud components. (a) Electronic sow feeder, (b) central controller, (c) PDA, (d) feed discharge pipe and water discharge pipe. (1) The central controller sends request to scheduling server, (2) permission information is verified for the central controller, (3) an MQTT server node is selected, (4) the internet protocol (IP) and port of the selected server node are returned, (5) the central controller establishes a connection with the service node, (6) the service node transfers the received data to the data processing program, (7) composition of data processing program, and (8) PC and phone.

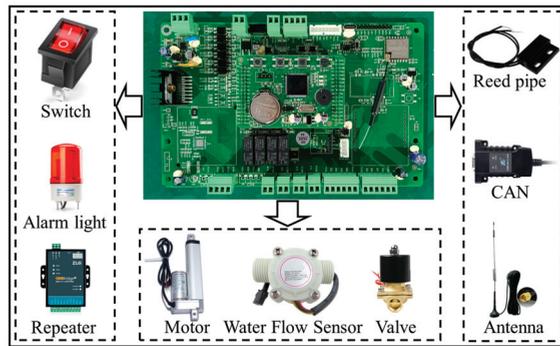
2.2. ESF and Central Controller

Figure 1 shows the designed system architecture, which includes the ESF and central controller. Detailed information of the components and instrumentation is provided in Table 1.

An entire circuit setup with its components is included inside the plastic container. The feeder control method adopts STM32 (ST Microelectronics 32-bit Microcontroller) embedded technology (Made by STMicroelectronics, Geneva, Switzerland). The core board chip uses STM32F103VET6 with large-capacity storage. The CPU is an ARM Cortex-M3, and the internal clock frequency of the main chip is increased to 72 MHz. The maximum power dissipation is 434 MW, and the operating temperature range is $-40\text{ }^{\circ}\text{C}$ to $85\text{ }^{\circ}\text{C}$. The main interfaces of the core circuit include CAN-bus communication, RS-232 serial communication, WiFi communication, motor control output, flow meter A/D acquisition, and alarm light display, as shown in Figure 2.

Table 1. Brief information of components used.

Component Name	Model	Operating Voltage	Output	Component Role	Manufacturers
Motor	10–50 mm	24 V	Analog	Maximum thrust: 250 N	Louis, Changzhou, China
Water flow sensor	YF-S201	5 V	Digital	Flow range: 1–25 L/min	Dijiang, Cangzhou, China
Electromagnetic valve	2W160-15	24 V	Analog	Temp: −5 °C–150 °C	Xiangjun, Hangzhou, China
CAN	USBCAN-E-mini	5 V	Digital	Baud rate: 5 kbps~1 Mbps	Ligong, Guangzhou, China

**Figure 2.** Main interfaces of the core circuit.

The central controller is a 12-inch touchscreen all-in-one machine (Huaxiang Zhichuang Technology Co., Ltd., Shenzhen, China). The system runs on Windows 10 (Intel(R), Celeron(R) CPU J1900, 1.99 GHz, Intel Corp, Santa Clara, CA, USA), with a memory (RAM) capacity of 4.00 GB. The central controller dynamically records and displays the actual feed intake and water consumption of sows, with the operating interface programmed using the C++ language.

2.3. MQTT Service Cluster

The purpose of the MQTT cluster service is to complete the reception of data and the issuance of operation instructions [25]. In this study, an MQTT server was designed based on the Netty [26] network communication framework to receive and parse data.

2.3.1. MQTT Service Node Design

Netty is a framework for quickly developing high-performance and reliable network servers that is easy to use [27]. The MQTT protocol is a very simple, lightweight information transmission protocol designed for use with restricted equipment, low bandwidth, high latency, or unreliable networks [28]. Many studies have shown that MQTT outperforms other protocols [29–31].

This section describes the design of a single-node server based on Netty and MQTT protocols. When the node starts, the ServerBootstrap component of Netty binds to a network port to listen to and process connection requests from the central controller. After a successful connection, the WorkerEventLoopGroup receives data sent by the central controller and allocates processing threads from the thread pool to the corresponding connection channels. The encoding and decoding of pig feeding data are achieved using MqttDecoder and MqttEncoder. Finally, data parsing is performed through a self-written MqttServerHandler. The feed and water intake data are sent to MySQL [32], while the electronic feeder log data are sent to Kafka [33]. The process of the server handling Connect, Subscribe, and Publish requests is depicted in Figure 3.

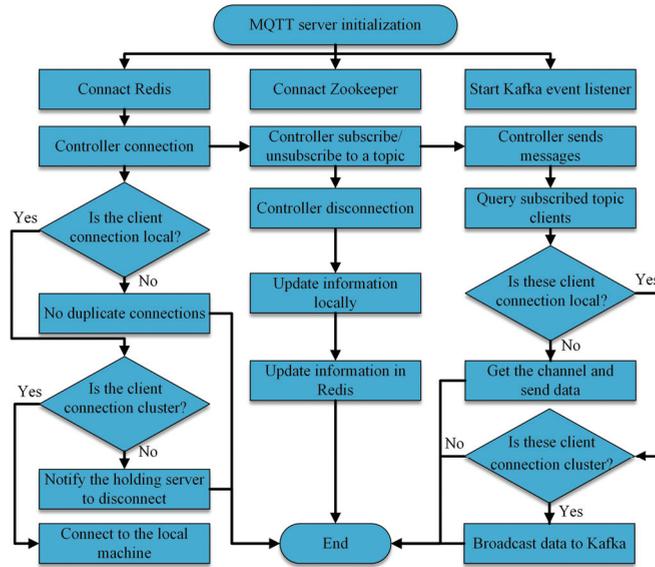


Figure 3. The process of the server handling Connect, Subscribe, and Publish requests.

2.3.2. MQTT Service Cluster Construction

As the number of central controllers in pig farms increases, single-node servers quickly reach performance bottlenecks, making it difficult to monitor a large number of sows in real time. The cluster architecture can upgrade system scalability, stability, and performance at a lower cost [34]. Ensuring the cluster can provide services to the central controller of the pig farm normally during the cluster construction process, and the registration, scheduling, and management of nodes, are very important. The cluster organizational structure is shown in Figure 4.

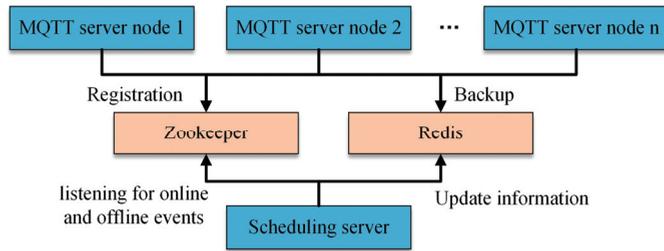


Figure 4. The components and overall processes used in the cluster.

A. Registration of cluster nodes

The registration of MQTT service nodes in the cluster is completed using ZooKeeper and Redis. After the successful initialization of MQTT service nodes, corresponding ephemeral nodes are created in Zookeeper. The ephemeral nodes store information such as the MQTT server name, IP (Internet Protocol), port, and the number of connected central controllers. The system utilizes Zookeeper as the registration center, with the scheduling server acting as the Zookeeper client. By employing the Zookeeper node listening mechanism, it detects the online and offline status of MQTT servers and synchronizes data to Redis. The node registration process is shown in Figure 5.

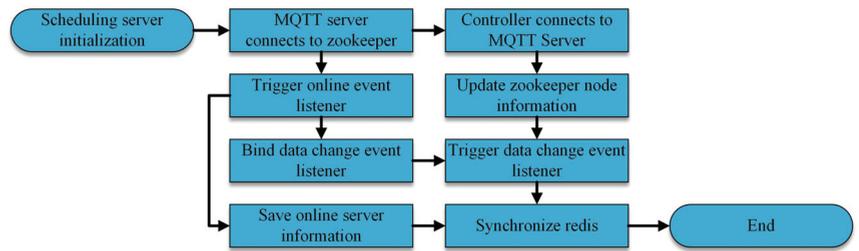


Figure 5. The registration process of the MQTT server in the cluster.

B. Connection and selection of cluster nodes

The central controller of the pig farm sends data based on the MQTT protocol, so the central controller needs to use socket information (IP and port) to establish a connection with the cluster. Each MQTT node in the cluster has its own socket information, making the central controller unable to establish a connection with the cluster through a unified entry point. Moreover, the service nodes of the cluster need to be scheduled reasonably.

To address the issues of node connection and node scheduling in the cluster, we propose deploying a scheduling server in the MQTT service cluster. This provides a unified connection interface for the central controller and realizes the reasonable scheduling of service nodes. The scheduling server is implemented using the Netty framework, essentially functioning as an HTTP (Hypertext Transfer Protocol) server. The central controller accesses the HTTP interface provided by the scheduling server (Figure 1(1)). After receiving the HTTP request, the system performs authorization validation (Figure 1(2)) and then selects appropriate MQTT service nodes using a load-balancing algorithm (Figure 1(3)). The scheduling server returns the IP and port of the selected node as interface response values to the central controller (Figure 1(4)). After receiving the socket information, the central controller establishes a connection with the target node (Figure 1(5)).

The node-scheduling server selects suitable service nodes for the central controller of the pig farm by implementing load-balancing algorithms. The algorithmic selection result is primarily determined by two factors: the load of service nodes, which refers to the number of connected pig farm controllers, and the weight of server nodes, which represents the performance configuration of the servers.

The load-balancing algorithm adopts the concept of a weighted connection count. It selects nodes with higher weights and lower loads for the central controller each time. Assuming there are n server nodes, the selection rules are as shown in Equation (1).

$$S_k = \min \left\{ \frac{C(S_i)/C_{sum}}{W(S_i)} \right\}, (i = 0, 1, 2, \dots, n - 1) \quad (1)$$

where i is the selected server node; $C(S_i)$ is the number of connections held by server node i ; $W(S_i)$ is the weight of server node i ; $W(S_i) > 0$; and $C_{sum} = \sum C(S_i)$ is the total number of connections held by server nodes in the cluster.

Because C_{sum} is a constant, and division requires more CPU cycles than multiplication, the optimization of the judgment condition can be represented as in Equation (2).

$$C(S_k) \times W(S_i) < C(S_i) \times W(S_k) \quad (2)$$

When there are multiple nodes in the cluster, this rule is utilized to sort and identify the node that currently has a lower load and higher weight.

2.3.3. Transmission Security Design

Devices may be subject to malicious connections from unrelated devices when service nodes provide MQTT connections externally [35]. This can lead to increased server load and affect normal operation. Because the MQTT CONNECT message permits the

inclusion of a username and password, the system is designed with a dynamic password-generation algorithm for authentication during the establishment of connections by the central controller.

The core concept of dynamic password generation is illustrated in Figure 6 and can be summarized as follows: (1) randomly generate a string S of length len ; (2) extract characters from indices m , $m - m/2$, and $m + m/2$, convert these three characters to decimals, and calculate Num ; (3) calculate result by combining Num with the system time ($Time$); (4) convert the result to a hexadecimal and insert it at position $S[m]$ in the original string S to generate a new string S' .

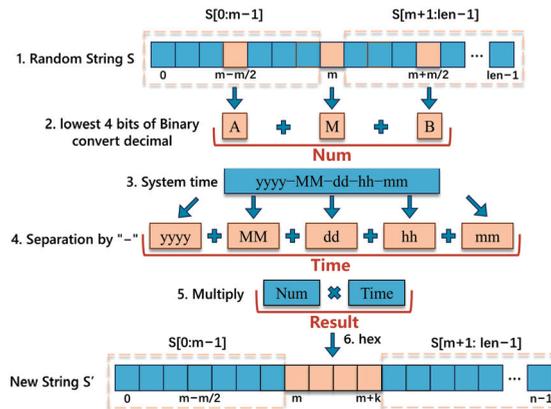


Figure 6. The core concept of dynamic password generation.

The algorithm incorporates the current system time into the calculation, enabling the addition of timeliness to the generated verification code. The summation of the split characters will ignore the seconds when the time format is set to “yyyy-MM-dd-hh-mm”. The system will yield the same result when verifying the verification code within one minute. If the time format changes to “yyyy-MM-dd-hh”, then the expiration time changes from one minute to one hour. Additionally, the efficiency of authentication is improved by sending the dynamic password only once for authentication. However, the password generation algorithm requires setting factory information for the central controller in the system, including registration string length (len), index (m), and time format. The algorithm is suitable for scenarios where device registration information can be prepared in advance.

2.3.4. Integration of ProtoStuff Encoder and Decoder

Netty can integrate third-party encoders to accelerate data transmission efficiency. We compared mainstream serializers such as Java Serializer, Marshalling, and Protostuff. The experimental results demonstrate that Protostuff exhibits the highest efficiency. Therefore, we ultimately finalized the integration with Protostuff. Due to the absence of corresponding encoders and decoders in Netty, users need to implement Protostuff encoders and decoders by themselves. The Protostuff encoder and decoder need to inherit from the `MessageToByteEncoder` class and the `ByteToMessageDecoder` class provided by Netty. It is also necessary to override the `encode()` and `decode()` methods within the inherited classes. Thus, the data flow sent and received by the MQTT server is directed to the Protostuff encoder and decoder for processing.

2.4. Data Processing Program

The system was built on a big data framework to establish a data processing platform for the statistical analysis of data. The platform was built upon HDFS (Hadoop Distributed File System) [36] and Spark [37] to establish an offline data warehouse, which is utilized for processing parsed sow production data. The data platform constructs hierarchical

dimension tables, including fields such as pig farm ID, house number, stall number, entry time, exit time, and farrowing date. The data platform utilizes dimensional tables to perform statistical analyses on litter size, number of live births, feed intake, water intake, and other data at different time intervals and pig farm granularities. The system utilizes Kafka and Flink for the real-time collection and processing of controller log data. The system conducts filtering and aggregation operations in Flink to calculate the number of devices online, the quantity of alarms, alarm information, and operational dynamics for intelligent devices in each pig farm. The architecture of the data processing platform is illustrated in Figure 7.

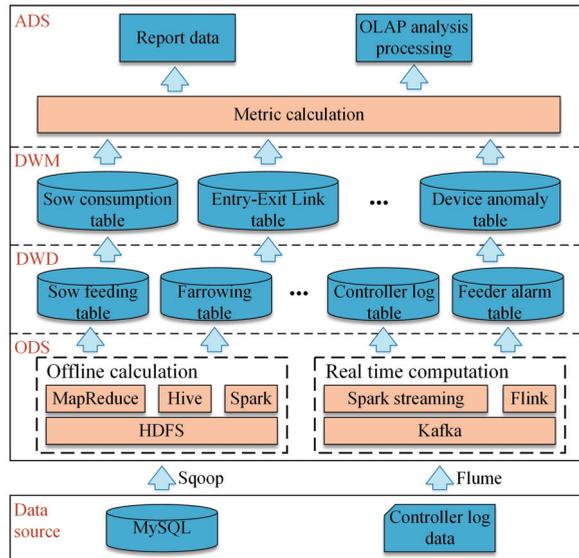


Figure 7. The architecture of the data processing platform. (1) ODS stands for operation data source, (2) DWD stands for data warehouse details, (3) DWM stands for data warehouse middle, (4) ADS stands for application data service, and (5) OLAP stands for Online Analysis Processing.

2.5. System Testing Methods

2.5.1. Test Environment

The system described in this paper was deployed on three Alibaba Cloud servers. The servers are of the shared computing type n4 (ecs.n4.small). The configuration information parameters are as follows: Intel(R) Xeon(R) CPU E5-2682 v4 @ 2.50 GHz, 1 vCPU, 2 GB RAM, and 100 GB HDD. The operating system was Alibaba Cloud Linux release 3, and JDK version 1.8 was used. Additionally, two computers with 8-core CPUs were used as load-testing machines. The detailed configuration information parameters are as follows: Intel(R) Core(TM) i5-1035G1 processor, 16 GB RAM, 500 GB solid-state drive, and Windows 10 operating system.

2.5.2. Test Indicators

Transactions per second (TPS): TPS is a performance metric for system processing capability, indicating the number of messages processed per second. A higher TPS indicates better system performance.

System response time (RT): RT is the time it takes for the system to respond to user input or requests. A shorter system response time indicates better system performance.

Error rate (ER): ER is the number of request errors as a percentage of the total number of requests.

2.5.3. Test Methods

A. Performance Testing of the Scheduling Server

The test involved performance testing of the scheduling server. The performance of the scheduling server was tested and compared with mainstream Tomcat web applications. The Tomcat web application was written in the Java programming language. We employed JMeter to conduct separate tests on both the Tomcat web application and the scheduling server [38]. The testing process was as follows: (1) Ten sets of HTTP requests were configured in JMeter. (2) The request threads were configured to start and complete within 3 s for each of the three cycles. (3) The comparison of the response time, throughput, and failure rate of the two systems was based on the JMeter aggregated report.

B. Load-balancing algorithm correctness testing

The test utilized three server nodes: node1, node2, and node3. The weight ratio among the three nodes was 6:3:1 with an initial connection count of 0. The testing machine sent HTTP requests to the scheduling server to trigger the load-balancing algorithm. After receiving responses, the testing machine recorded the node selection results to verify the correctness of the algorithm. The test evaluated the correctness of the algorithm under scenarios of node failure and reconnection after disconnection. The testing process was as follows: (1) Firstly, all three server nodes were simultaneously started. Then, during the test machine sending the request, we shut down the server node1. (2) Firstly, both server node2 and node3 were simultaneously started. Then, during the test machine sending the request, we opened server node1. (3) We recorded the node selection results.

C. Cluster Performance Testing

The average response time and the number of requests processed per unit time by the cluster in handling MQTT requests were tested using JMeter software (version: 5.6.2). The testing process was as follows: (1) We configured multiple sets of MQTT request counts and completed initialization within 1 s. (2) Each MQTT request sent a 507 byte JSON string to the server. (3) The cluster parsed the data and replied with a successful reception message.

D. Encoding and decoding efficiency testing

In order to test the impacts of different encoding and decoding methods on MQTT server data transmission, we compared the performances of three encoding and decoding methods: Java serialization, Marshalling, and Protostuff. The testing process was as follows: (1) The testing machine was configured with 100 thread groups and sent data three times. It sent 100, 500, and 1000 packets each time. The packet size was 1250 bytes and included the current timestamp information T_1 . (2) The tested program appended the current timestamp information T_2 to the data packet after completing data parsing and sent the data packet back to the load-testing machine. (3) The testing machine calculated the response time T_{rs} of the program based on the two timestamps (T_1 and T_2) in the packet.

$$T_{rs} = T_2 - T_1 \quad (3)$$

3. Results

3.1. Test Results

3.1.1. Scheduling Server Performance Test Results

The test results are shown in Figure 8a,b. The response time of the scheduling server was within 250 ms under 22,000 requests. The overall response time of the scheduling server was faster than that of the Tomcat web application. The TPS of the Tomcat Web application peaked at 6301 pcs/s with 16,000 requests, accompanied by a response time of 153 ms. The TPS of the scheduling server peaked at 7351 pcs/s with 18,000 requests, accompanied by a response time of 145 ms. The scheduling server exhibited an increase of 1050 pcs/s compared to the Tomcat application. The Tomcat web application and the scheduling server experienced request failures at 20,000 and 22,000 request counts, respectively. The

failure rates of both began to increase after the number of requests exceeded 20,000. Clearly, the increase in resource utilization led to a higher failure rate.

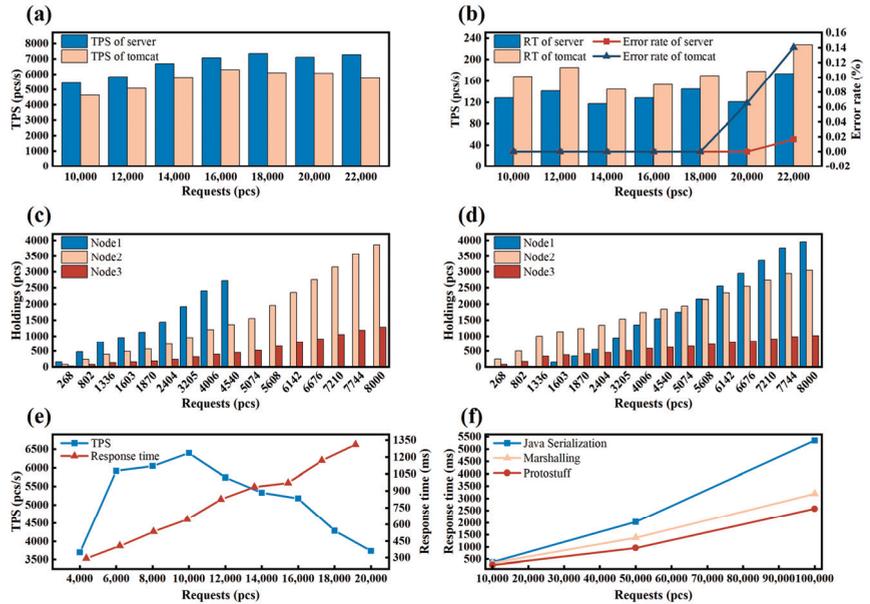


Figure 8. Test result graph. (a) TPS results of scheduling server and Tomcat application, (b) RT and ER results of scheduling server and Tomcat application, (c) the change in node connections when node1 crashes, (d) the change in node connection count during the reconnection of node1, (e) TPS and RT test results for cluster performance, and (f) TPS comparison results of three serialization methods.

3.1.2. Load-Balancing Algorithm Correctness Test Results

The test results are shown in Figure 8c,d. Figure 8c shows the change in node connections when node1 crashes. When node1 was shut down after the 5000th request, the connection counts of node2 and node3 gradually approached a ratio of 3:1. Figure 8d illustrates the change in node connection count during the reconnection of node1. Node1 was prioritized for connection requests after it was started at the 1500th request. Finally, the connection ratio of the three nodes tended towards 6:3:1.

The test results indicate that nodes with higher weights received more connections, but they did not hold all connections. As the number of requests increased, the proportion of connections among all nodes tended to approach the ratio of node weights. The results demonstrate that the load-balancing algorithm was influenced by both weights and connection counts, and it functioned properly even during node failures and reconnects.

3.1.3. Cluster Performance Test Results

The test results are shown in Figure 8e. When the number of requests reached 4000, 6000, and 8000, the TPS of the cluster gradually increased to 3699 pcs/s, 5917 pcs/s, and 6050 pcs/s respectively. When the number of requests reached 10,000, the TPS peaked at 6399 pcs/s, with a corresponding response time of 643 ms. After the number of requests exceeded 10,000, the TPS gradually decreased as the system reached its bottleneck. When there were 20,000 requests, the TPS was 3738 pcs/s, with a response time of 1308 ms. The test results indicate that the system performed well with 10,000 connections. Overall, it was capable of meeting the communication requirements of the central controller in pig farms.

3.1.4. Encoding and Decoding Efficiency Test Results

The test results are shown in Figure 8f. Under the test with 10,000 data packets, Protostuff showed a response time of 258 ms, while Marshalling and Java serialization showed 348 ms and 376 ms. The test results indicate that Java serialization had the longest response time, while Protostuff had the shortest response time. The advantages of Protostuff are more obvious with many data.

3.2. PC and App Application Interface

The system supports real-time access from both mobile phones and computers. The equipment status interface (Figure 9a) provides real-time displays on the deployment, online, and abnormal status of electronic sow feeding in the pig farm, along with detailed alarm information prompts. When the sow's feed intake is less than 40% of the planned feeding amount set by the electronic sow feeding, the system determines that the sow's feed intake is insufficient. Clicking on the number allows for querying the location of the alarmed electronic sow feeding in the stall. Figure 9b provides a summary of information for each farrowing house, including the number of sows, total feed intake for the day, and total feed intake for the batch. The feeding status interface displays detailed information for each sow in the stall (Figure 9c), including its ear tag number, farrowing information, and feed intake level. The system also supports displaying historical feed intake data for the sow since its entry into the stall in the form of a line graph. The feeding scheme interface (Figure 9d) displays the feeding plan for sows throughout the entire farrowing cycle, including key information such as daily feeding schedules and feed-to-water ratios.



Figure 9. The mobile interface. (a) The equipment status interface, (b) the farrowing house summary information interface, (c) the feeding status interface, and (d) the feeding scheme interface.

As shown in Figure 10, the PC interface displays the online status and detailed information of the MQTT server nodes in the cluster. This information includes TCP (Transmission Control Protocol) connection count, weight information, and node IP addresses, as well as the CPU, memory, and disk usage of the servers. Additionally, the system monitors the resource usage of key components such as MySQL, Zookeeper, Redis, and Kafka. When the system encounters abnormalities, it will notify administrators via email for timely resolution.

The system displays statistical data after the completion of data platform processing in the backend management interface (as shown in Figure 11). The interface presents data such as feed and water intake per farrowing house, daily farrowing statistics, sow feeding plans, and sow anomalies. Additionally, it allows for querying detailed meal information for each sow throughout the day.

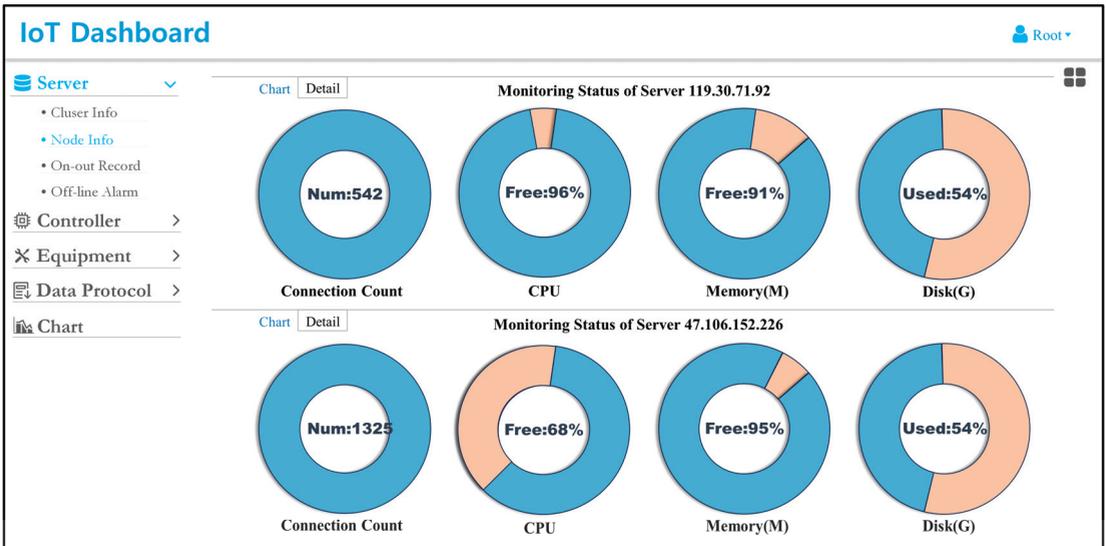


Figure 10. The online status and detailed information of the MQTT server nodes in the cluster. Blue means free, orange means used.



Figure 11. The backend management interface for displaying statistical data.

4. Discussion

4.1. Analysis of Feeding Behaviors in Pig Farms

This system is installed at the ZhongRenWang sow farm in Baise City, Guangxi Province, China (longitude: 105.465639, latitude: 24.595144). We installed equipment in five farrowing houses of the pig farm, with each farrowing house installing 1 central controller and 59 electronic sow feeders. In this section we analyze the feeding data of lactating sows collected from 1 June 2023 to 1 February 2024. The fields of the original data are shown in Table 2.

Table 2. Fields of original data.

Field Name	Data Type	Note
Pig_Farm_Id	Number	The only sign of the pig farm.
Center_Controller_Id	Number	The only sign of the central controller in houses.
ESF_Id	Number	The only sign of the ESF in farrowing houses.
House_Id	Number	The only sign of the farrowing house.
Stall_Id	Number	The only sign of the stall in farrowing houses.
Current_Date	String	Current date, format is “yyyy-MM-dd”.
Has_Pig	Boolean	Has a sow entered the stall?
Enter_Data	String	Sow entry date, format is “yyyy-MM-dd”.
Enter_Day	Number	Number of days sows have been in stall.
Is_Farrowing	Boolean	Has the sow finished farrowing?
Farrowing_Date	String	Date of farrowing, format is “yyyy-MM-dd”.
Feed_Intake	Float	Feed intake of the sow throughout the day.
Water_Intake	Float	Water intake of the sow throughout the day.

We eliminated data with negative values for feed and water intake and discontinuous entry dates. The time range of the production data of each lactating sow is from the time of entry to the stall to the time of exit from the stall. After cleaning the data, the production data of 1447 lactating sows were obtained. Moreover, descriptive analysis of the obtained data was performed using SPSS.

4.1.1. Production Interval Days Frequency Statistics

We calculated the frequency of different production interval days by analyzing the time intervals between sows in the same stall within each farrowing house (i.e., the time from the end of one sow’s lactation to the beginning of the next sow’s lactation). The time interval was calculated by subtracting the entry time of the next sow from the exit time of the previous sow. Finally, after the calculation of all stalls was completed, the number of occurrences of different time intervals was counted.

The frequency distribution of production interval days for each farrowing house is shown in Figure 12a. SPSS statistical indicators are shown in Table 3. The highest frequency days in each farrowing house are 8 d, 4 d, 5 d, 5 d, and 8 d. The average production interval days are 10.79 d, 8.06 d, 6.95 d, 12.29 d, and 12.29 d. These data show that the production interval days of the pig farm range from 6.95 d to 12.29 d, and the production rhythm of the No. 2 farrowing house is faster than that of other houses.

Table 3. Statistical indicator data of production interval days, weaning days, and prepartum days.

House	Production Interval Days			Weaning Days			Prepartum Days		
	Mode	Average	SEM *	Mode	Average	SEM	Mode	Average	SEM
House 1	8	10.79	0.465	21	19.51	0.176	13	13.16	0.235
House 2	4	8.06	0.282	23	22.85	0.212	3	5.03	0.193
House 3	5	6.95	0.184	19	19.35	0.200	12	9.80	0.175
House 4	5	12.29	0.462	18	17.68	0.288	6	6.42	0.119
House 5	8	12.29	0.317	13/21	17.91	0.244	11	8.44	0.163

* SEM: Standard error of mean.

4.1.2. Distribution of Weaning Days Frequency Statistics

Figure 12b shows the frequency distribution of weaning days for lactating sows. It represents the number of days post-farrowing when sows finish lactation. The number of weaning days was obtained by subtracting the farrowing date from the exit stall date of the lactating sow.

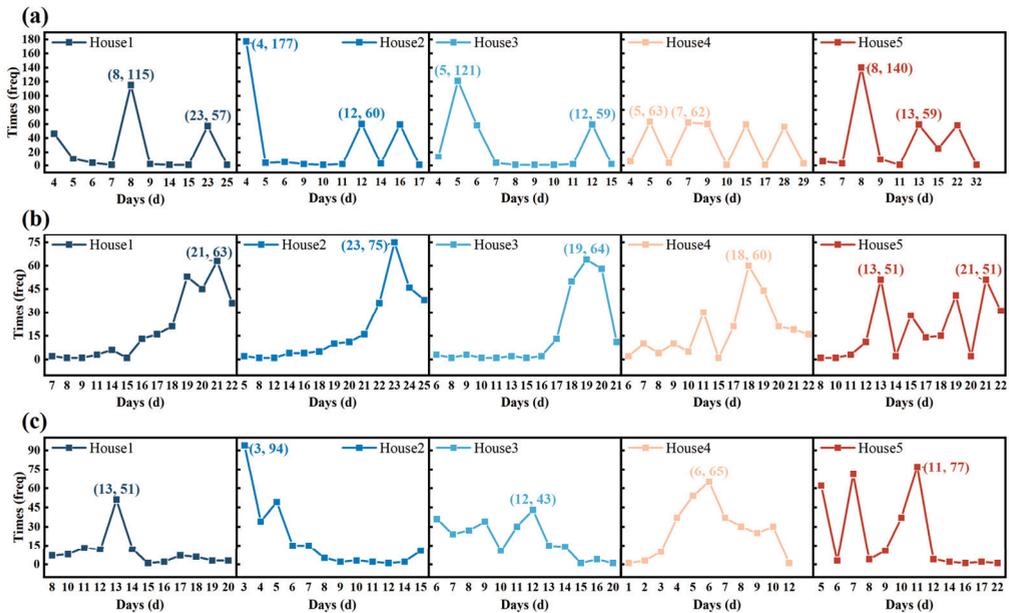


Figure 12. Frequency distribution figure of days. (a) The frequency distribution of production interval days for each farrowing house, (b) the frequency distribution of weaning days for lactating sows, and (c) the frequency distribution of prepartum days for lactating sows.

SPSS statistical indicators are shown in Table 3. The most frequent weaning days for each farrowing house are 21 d, 23 d, 19 d, 18 d, and 21 d. The data reflect the fact that the sow farms will wean the lactating sows around 20 days. Weaning in intensive pig production usually occurs between days 21 and 28 of age, which is beneficial for sow recovery and piglet survival [39]. Another frequently occurring weaning day in the No. 5 farrowing house is Day 13. This indicates that early weaning occurred in the farrowing house during a specific period.

4.1.3. Distribution of Prepartum Days Frequency Statistics

Figure 12c shows the frequency distribution of prepartum days for sows. It represents the number of days before parturition during which lactating sows entered the farrowing house. The number of prepartum days was obtained by subtracting the entry date from the farrowing date of the lactating sow.

SPSS statistical indicators are shown in Table 3. The most frequent prepartum days in each farrowing house are 13 d, 3 d, 12 d, 6 d, and 11 d. In intensive pig production, sows will enter the farrowing house about 7 days before parturition [40]. The average data indicate that the farm typically places lactating sows in the farrowing house around 5.03 d–13.16 d before farrowing, which is basically in line with the production rules of intensive pig farms. It is noteworthy that 94 sows in the No. 2 farrowing house were placed in the farrowing house only 3 days before farrowing. This may have increased the sows' stress response. The cause of this situation may be related to the fast production pace of this farrowing house.

4.1.4. Feed and Water Intake of Sows

Through the above analysis, it has been made evident that sows in this pig farm enter the farrowing house 10 days before farrowing and are weaned 20 days after farrowing. Therefore, the average feed intake and average water consumption levels of sows within this time range were calculated for analysis. The results are shown in Figure 13. Sows

showed a noticeable decrease in both feed intake and water consumption in the 3 days before farrowing. Sows consumed only a small amount of water and hardly ate on the day of farrowing. After farrowing, the feed intake and water consumption of sows started to increase gradually day by day. The growth rate was fastest during the first 7 days, slowed down between days 7 and 14, and stabilized after 14 days. In some farrowing houses, feed intake and water consumption decreased between 18 and 20 days. After consulting with the farrowing house manager, this was found to be related to the early weaning practice employed by the farm.

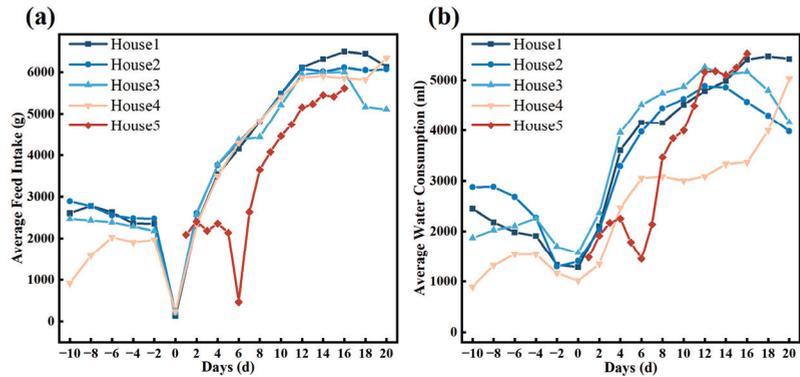


Figure 13. Feed and water intake of sows. (a) The average feed intake for each farrowing house and (b) the average water consumption for each farrowing house.

It is difficult to collect accurate feeding data for each production batch in traditional pig farms. Digital devices and systems can effectively address this issue. The production interval days can effectively reflect the production efficiency of the farrowing house [41]. The lactating sows at an average of 4.8 days after weaning have a lower anestrus rate [42]. By 1 week after weaning, 90–95% of the multiparous sows are expected to exhibit estrus [43]. Pig farm managers can utilize this system to track the duration between production cycles in the farrowing house. Additionally, the production rhythm of the farrowing house should be reasonably arranged based on the duration of the lactating sow's estrus cycle.

Related research indicates that extending the weaning age from 19 to 28 days has a positive effect on pig performance in the nursery, resulting in increased body weight sold per pig weaned. This suggests that 25 days is the optimal age for weaning [41,44]. Furthermore, increasing the weaning age from 18.5 to 24.5 days positively affected pig performance by increasing the average daily gain and feed intake [45]. Using digital technology to count the number of parturition and weaning days can help correct erroneous production strategies on pig farms. Furthermore, tracking the feed and water intake of lactating sows on different days will aid in establishing appropriate feeding plans for future pig farms.

4.2. IoT of Pig Farm

We focused on constructing an Internet of Things (IoT) system specifically designed for the feeding requirements of lactating sows. Table 4 shows the architectural layers of the present and earlier systems, as well as data displays, data transmission, and cluster technology, confirming that the proposed monitoring system is reasonable compared to other similar systems.

Table 4. Comparison of other similar systems with the present system.

Paper	Application	Architecture Layers	Cluster System	Data Display	Data Transmission
[46]	Livestock monitoring	2	No	PC	TCP/IP
[16]	Precision feeding	3	No	Mobile and PC	GPRS
[22]	Livestock monitoring	3	Yes	PC	HTTP
[15]	Intelligent feeding	2	No	Mobile and PC	GPRS
[12]	Electronic sow feeders	1	No	No	No
[47]	Environmental monitoring	2	No	No	WIFI
[48]	Environmental monitoring	3	No	PC	GPRS
This proposal	Sow monitoring	4	Yes	Mobile and PC	MQTT

As the level of technology in pig farms increases, an increasing number of intelligent devices are being employed. Various intelligent devices are used in farrowing houses, such as environmental monitoring equipment [47] and weighing devices [49]. Different intelligent devices can complement each other's functional deficiencies. However, the data barriers between different devices result in poor interoperability among them. This phenomenon is attributed to the adoption of different communication protocols and data formats by intelligent devices produced by different manufacturers. It is important to research universal data formats and communication protocols applicable to intelligent devices in pig farming [50]. Due to the lack of standardization in communication protocols for intelligent devices, the development of IoT platforms currently faces challenges, including high complexity and poor adaptability. The IoT platform struggles to meet the connectivity requirements of various types of intelligent devices.

With the further development of deep learning, intelligent cameras and recording devices have begun to be used in pig farms. They play an important role in disease monitoring [51], behavior monitoring [52], and video-based weight estimation [53]. The collection and storage of unstructured data such as video and audio also face challenges. The next focus of our research will be on meeting the connectivity requirements of diverse intelligent devices and achieving the remote collection of unstructured data (with a focus on audio and video) based on the developed system. In addition, pig farms have a high demand for the digital management of personnel, pigs, vehicles, and materials. It will be important to improve the management level of pig farms and prevent African swine fever by building a comprehensive digital platform for the entire pig farm using digital technology [54].

Establishing a cluster effectively addresses the issue of system concurrency in order to cope with the increase in intelligent equipment in pig farms. The volume of data in the Internet of Things (IoT) is also enormous. Effective data processing and analysis are required to extract useful information and insights. However, this requires powerful computing and data analysis capabilities [55]. The maturity of e-commerce platforms has a certain reference value for the pig farm IoT. E-commerce platforms need to deal with large amounts of product information, order data, and user information. Pig farm IoT systems also need to manage large amounts of pig data, sensor data, and environmental data. Efficient data storage and processing systems can be established drawing from the data management experience of e-commerce platforms. By leveraging the data management experience of e-commerce platforms, efficient data storage and processing systems can be developed to effectively manage and analyze pig data. In the process of digitization in pig

farming, drawing from the experiences of other industries can accelerate its development. However, it is necessary to customize the design according to the characteristics and needs of pig farming.

4.3. Limitations

4.3.1. Accurate Acquisition of Feed and Water Intake for Lactating Sows

The feed and water intake data described in this article refer to the amounts dispensed by the sow electronic feeder. We assume that the sow consumes all the dispensed feed and water. However, this assumption may not always hold true, highlighting a limitation that needs to be addressed in future work. References [12,56,57] suggest that using load cells to accurately measure pig feed intake is an effective solution.

4.3.2. Multi-Dimensional Data Collection and In-Depth Data Analysis

This article describes the remote monitoring of feed and water intake for lactating sows. However, actual production requires the collection of additional data, such as feeding times, body weight, litter size, weaning litter size, and weaning litter weight. Gathering more comprehensive data and scientifically analyzing them in conjunction with feed and water intake will help pig farm managers develop more effective feeding plans and advance the intelligent feeding of lactating sows.

5. Conclusions

We developed a smart IoT feeding system for lactating sows. The system implements an MQTT server based on the Netty framework to collect, transmit, parse, and store information regarding feed intake, water consumption, and the operational status of electronic sow feeders from the cloud. The system implements clustered job scheduling based on an HTTP server and a custom load-balancing algorithm for node selection. Additionally, we utilized the Spark and Flink computing frameworks to construct a big data analytics platform. This study lays the foundation for further in-depth analysis of sow feeding data. In actual operation on a pig farm, the system runs stably and can provide real-time feedback on sow feeding conditions. It holds practical value for application in the production and construction of pig farms.

Author Contributions: X.H.: writing—original draft, writing—review and editing, conceptualization, methodology, investigation, data curation, software, formal analysis, validation, resources. Z.Z.: writing—review and editing, supervision, conceptualization, resources, funding acquisition. Y.L. (Yanhua Liu): investigation, data curation, software, formal analysis, resources, conceptualization, funding acquisition. E.L.: data curation, investigation, methodology, software. J.X.: data curation, investigation, methodology, software. F.W.: data curation, resources, investigation, review, software. Y.L. (Yizhi Luo): investigation, data curation, formal analysis, resources. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Guangzhou Key Research and Development Project (2023B03J 1363), Special Fund for the Rural Revitalization Strategy of Guangdong (2023TS-3, 2024TS-3), Key Laboratory of Modern Agricultural Intelligent Equipment in South China, Ministry of Agriculture and Rural Affairs, China (HNZJ202209), Guangzhou Basic and Applied Research Project (2023A04J0752), Independent Research Project of Maoming Laboratory (2021ZZ003, 2021TDQD002), and Fundamental Research Funds for the State Key Laboratory of the Swine and Poultry Breeding Industry (ZQQZ-25), (2023QZ-NK16).

Institutional Review Board Statement: Not applicable.

Data Availability Statement: Because the raw data involve production privacy on commercial pig farms, the data presented in this study are available upon request from the corresponding author.

Acknowledgments: We acknowledge Guangzhou Jiaen Technology Co., Ltd. for the materials and facilities.

Conflicts of Interest: Author Feiren Wang was employed by the company Guangzhou Jiaen Technology Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Wang, B.; Tao, F.; Fang, X.; Liu, C.; Liu, Y.; Freiheit, T. Smart Manufacturing and Intelligent Manufacturing: A Comparative Review. *Engineering* **2021**, *7*, 738–757. [CrossRef]
2. Laufs, J.; Borrión, H.; Bradford, B. Security and the Smart City: A Systematic Review. *Sustain. Cities Soc.* **2020**, *55*, 102023. [CrossRef]
3. Deebak, B.D.; Al-Turjman, F. Smart Mutual Authentication Protocol for Cloud Based Medical Healthcare Systems Using Internet of Medical Things. *IEEE J. Sel. Areas Commun.* **2021**, *39*, 346–360. [CrossRef]
4. Roy, S.K.; De, D. Genetic Algorithm Based Internet of Precision Agricultural Things (IopaT) for Agriculture 4.0. *Internet Things* **2022**, *18*, 100201. [CrossRef]
5. Pereira, W.F.; Fonseca, L.D.S.; Putti, F.F.; Góes, B.C.; Naves, L.D.P. Environmental Monitoring in a Poultry Farm Using an Instrument Developed with the Internet of Things Concept. *Comput. Electron. Agric.* **2020**, *170*, 105257. [CrossRef]
6. Gaikwad, S.V.; Vibhute, A.D.; Kale, K.V.; Mehrotra, S.C. An Innovative IoT Based System for Precision Farming. *Comput. Electron. Agric.* **2021**, *187*, 106291. [CrossRef]
7. Misra, N.N.; Dixit, Y.; Al-Mallahi, A.; Bhullar, M.S.; Upadhyay, R.; Martynenko, A. IoT, Big Data, and Artificial Intelligence in Agriculture and Food Industry. *IEEE Internet Things J.* **2022**, *9*, 6305–6324. [CrossRef]
8. Gauthier, R.; Largouët, C.; Dourmad, J.-Y. Prediction of Litter Performance in Lactating Sows Using Machine Learning, for Precision Livestock Farming. *Comput. Electron. Agric.* **2022**, *196*, 106876. [CrossRef]
9. Gauthier, R.; Largouët, C.; Bussiès, D.; Martineau, J.-P.; Dourmad, J.-Y. Precision Feeding of Lactating Sows: Implementation and Evaluation of a Decision Support System in Farm Conditions. *J. Anim. Sci.* **2022**, *100*, skac222. [CrossRef]
10. Chen, W.-E.; Lin, Y.-B.; Chen, L.-X. PigTalk: An AI-Based IoT Platform for Piglet Crushing Mitigation. *IEEE Trans. Ind. Inform.* **2021**, *17*, 4345–4355. [CrossRef]
11. Torres, A.B.B.; Da Rocha, A.R.; Coelho Da Silva, T.L.; De Souza, J.N.; Gondim, R.S. Multilevel Data Fusion for the Internet of Things in Smart Agriculture. *Comput. Electron. Agric.* **2020**, *171*, 105309. [CrossRef]
12. Vargovic, L.; Hermes, S.; Athorn, R.Z.; Bunter, K.L. Feed Intake and Feeding Behavior Traits for Gestating Sows Recorded Using Electronic Sow Feeders. *J. Anim. Sci.* **2021**, *99*, skaa395. [CrossRef]
13. Halachmi, I.; Guarino, M.; Bewley, J.; Pastell, M. Smart Animal Agriculture: Application of Real-Time Sensors to Improve Animal Well-Being and Production. *Annu. Rev. Anim. Biosci.* **2019**, *7*, 403–425. [CrossRef] [PubMed]
14. Gaillard, C.; Dourmad, J.-Y. Application of a Precision Feeding Strategy for Gestating Sows. *Anim. Feed. Sci. Technol.* **2022**, *287*, 115280. [CrossRef]
15. Ma, W.; Fan, J.; Zhao, C.; Wu, H. The Realization of Pig Intelligent Feeding Equipment and Network Service Platform. In *Computer and Computing Technologies in Agriculture XI*; Li, D., Zhao, C., Eds.; IFIP Advances in Information and Communication Technology; Springer International Publishing: Cham, Switzerland, 2019; Volume 546, pp. 473–484, ISBN 978-3-030-06178-4.
16. Chen, C.; Liu, X.; Liu, C.; Pan, Q. Development of the Precision Feeding System for Sows via a Rule-Based Expert System. *Int. J. Agric. Biol. Eng.* **2023**, *16*, 187–198. [CrossRef]
17. Hussain, F.; Hussain, R.; Hassan, S.A.; Hossain, E. Machine Learning in IoT Security: Current Solutions and Future Challenges. *IEEE Commun. Surveys Tuts.* **2020**, *22*, 1686–1721. [CrossRef]
18. Friha, O.; Ferrag, M.A.; Shu, L.; Maglaras, L.; Wang, X. Internet of Things for the Future of Smart Agriculture: A Comprehensive Survey of Emerging Technologies. *IEEE/CAA J. Autom. Sin.* **2021**, *8*, 718–752. [CrossRef]
19. Chamara, N.; Islam, M.D.; Bai, G.F.; Shi, Y.; Ge, Y. Ag-IoT for Crop and Environment Monitoring: Past, Present, and Future. *Agric. Syst.* **2022**, *203*, 103497. [CrossRef]
20. Prakash, C.; Singh, L.P.; Gupta, A.; Lohan, S.K. Advancements in Smart Farming: A Comprehensive Review of IoT, Wireless Communication, Sensors, and Hardware for Agricultural Automation. *Sens. Actuators A Phys.* **2023**, *362*, 114605. [CrossRef]
21. Dineva, K.; Atanasova, T. Design of Scalable IoT Architecture Based on AWS for Smart Livestock. *Animals* **2021**, *11*, 2697. [CrossRef]
22. Mateo-Fornés, J.; Pagès-Bernaus, A.; Plà-Aragónés, L.M.; Castells-Gasia, J.P.; Babot-Gaspa, D. An Internet of Things Platform Based on Microservices and Cloud Paradigms for Livestock. *Sensors* **2021**, *21*, 5949. [CrossRef] [PubMed]
23. Dineva, K.; Atanasova, T. Architectural ML Framework for IoT Services Delivery Based on Microservices. In *Distributed Computer and Communication Networks*; Vishnevskiy, V.M., Samouylov, K.E., Kozyrev, D.V., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2020; Volume 12563, pp. 698–711, ISBN 978-3-030-66470-1.
24. Si, L.F.; Li, M.; He, L. Farmland Monitoring and Livestock Management Based on Internet of Things. *Internet Things* **2022**, *19*, 100581. [CrossRef]
25. Liu, X.; Zhang, T.; Hu, N.; Zhang, P.; Zhang, Y. The Method of Internet of Things Access and Network Communication Based on MQTT. *Comput. Commun.* **2020**, *153*, 169–176. [CrossRef]
26. Maurer, N.; Wolfthall, M.A. *Netty in Action*; Manning Publications: Shelter Island, NY, USA, 2015; ISBN 978-1-61729-147-0.

27. Amoretti, M.; Pecori, R.; Protskaya, Y.; Veltri, L.; Zanichelli, F. A Scalable and Secure Publish/Subscribe-Based Framework for Industrial IoT. *IEEE Trans. Ind. Inf.* **2021**, *17*, 3815–3825. [CrossRef]
28. Mishra, B.; Kertesz, A. The Use of MQTT in M2M and IoT Systems: A Survey. *IEEE Access* **2020**, *8*, 201071–201086. [CrossRef]
29. Glaroudis, D.; Iossifides, A.; Chatzimisios, P. Survey, Comparison and Research Challenges of IoT Application Protocols for Smart Farming. *Comm. Com. Inf. Sc.* **2020**, *168*, 107037. [CrossRef]
30. Naik, N. Choice of Effective Messaging Protocols for IoT Systems: MQTT, CoAP, AMQP and HTTP. In Proceedings of the 2017 IEEE International Systems Engineering Symposium (ISSE), Vienna, Austria, 11–13 October 2017; IEEE: Vienna, Austria, 2017; pp. 1–7.
31. Luzuriaga, J.E.; Perez, M.; Boronat, P.; Cano, J.C.; Calafate, C.; Manzoni, P. A Comparative Evaluation of AMQP and MQTT Protocols over Unstable and Mobile Networks. In Proceedings of the 2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC), Las Vegas, NV, USA, 9–12 January 2015; pp. 931–936.
32. Nichter, D. *Efficient MySQL Performance*; O'Reilly Media, Inc.: Sonoma County, CA, USA, 2021; ISBN 978-1-09-810506-8.
33. Calderon, G.; del Campo, G.; Saavedra, E.; Santamaría, A. Monitoring Framework for the Performance Evaluation of an IoT Platform with Elasticsearch and Apache Kafka. *Inf. Syst. Front.* **2023**, 1–17. [CrossRef]
34. Nguyen, N.; Kim, T. Toward Highly Scalable Load Balancing in Kubernetes Clusters. *IEEE Commun. Mag.* **2020**, *58*, 78–83. [CrossRef]
35. Yazdinejad, A.; Parizi, R.M.; Dehghantanha, A.; Zhang, Q.; Choo, K.-K.R. An Energy-Efficient SDN Controller Architecture for IoT Networks with Blockchain-Based Security. *IEEE Trans. Serv. Comput.* **2020**, *13*, 625–638. [CrossRef]
36. Mostafaeipour, A.; Jahangard Rafsanjani, A.; Ahmadi, M.; Arockia Dhanraj, J. Investigating the Performance of Hadoop and Spark Platforms on Machine Learning Algorithms. *J. Supercomput.* **2021**, *77*, 1273–1300. [CrossRef]
37. Zhang, X.; Li, X.; Du, H.; Ruiz, R. Task Scheduling for Spark Applications with Data Affinity on Heterogeneous Clusters. *IEEE Internet Things J.* **2022**, *9*, 21792–21801. [CrossRef]
38. Wang, J.; Wu, J. Research on Performance Automation Testing Technology Based on JMeter. In Proceedings of the 2019 International Conference on Robots & Intelligent System (ICRIS), Haikou, China, 15–16 June 2019; pp. 55–58.
39. Blavi, L.; Solà-Oriol, D.; Llonch, P.; López-Vergé, S.; Martín-Orúe, S.M.; Pérez, J.F. Management and Feeding Strategies in Early Life to Increase Piglet Performance and Welfare around Weaning: A Review. *Animals* **2021**, *11*, 302. [CrossRef]
40. Tokach, M.D.; Menegat, M.B.; Gourley, K.M.; Goodband, R.D. Review: Nutrient Requirements of the Modern High-Producing Lactating Sow, with an Emphasis on Amino Acid Requirements. *Animal* **2019**, *13*, 2967–2977. [CrossRef]
41. Bortolozzo, F.P.; Zanin, G.P.; Ulguim, R.D.R.; Mellagi, A.P.G. Managing Reproduction in Hyperprolific Sow Herds. *Animals* **2023**, *13*, 1842. [CrossRef]
42. Gianluppi, R.D.F.; Lucca, M.S.; Mellagi, A.P.G.; Bernardi, M.L.; Orlando, U.A.D.; Ulguim, R.R.; Bortolozzo, F.P. Effects of Different Amounts and Type of Diet during Weaning-to-Estrus Interval on Reproductive Performance of Primiparous and Multiparous Sows. *Animal* **2020**, *14*, 1906–1915. [CrossRef]
43. Poleze, E.; Bernardi, M.L.; Amaral Filha, W.S.; Wentz, I.; Bortolozzo, F.P. Consequences of Variation in Weaning-to-Estrus Interval on Reproductive Performance of Swine Females. *Livest. Sci.* **2006**, *103*, 124–130. [CrossRef]
44. Faccin, J.E.G.; Laskoski, F.; Hernig, L.F.; Kummer, R.; Lima, G.F.R.; Orlando, U.A.D.; Gonçalves, M.A.D.; Mellagi, A.P.G.; Ulguim, R.R.; Bortolozzo, F.P. Impact of Increasing Weaning Age on Pig Performance and Belly Nosing Prevalence in a Commercial Multisite Production System. *J. Anim. Sci.* **2020**, *98*, skaa031. [CrossRef] [PubMed]
45. Faccin, J.E.G.; Tokach, M.D.; Allerson, M.W.; Woodworth, J.C.; DeRouchey, J.M.; Dritz, S.S.; Bortolozzo, F.P.; Goodband, R.D. Relationship between Weaning Age and Antibiotic Usage on Pig Growth Performance and Mortality. *J. Anim. Sci.* **2020**, *98*, skaa363. [CrossRef]
46. Germani, L.; Mecarelli, V.; Baruffa, G.; Rugini, L.; Frescura, F. An IoT Architecture for Continuous Livestock Monitoring Using LoRa LPWAN. *Electronics* **2019**, *8*, 1435. [CrossRef]
47. Li, Y.; Fu, C.; Yang, H.; Li, H.; Zhang, R.; Zhang, Y.; Wang, Z. Design of a Closed Piggery Environmental Monitoring and Control System Based on a Track Inspection Robot. *Agriculture* **2023**, *13*, 1501. [CrossRef]
48. Zeng, Z.; Zeng, F.; Han, X.; Elkhouchlaa, H.; Yu, Q.; Lü, E. Real-Time Monitoring of Environmental Parameters in a Commercial Gestating Sow House Using a ZigBee-Based Wireless Sensor Network. *Appl. Sci.* **2021**, *11*, 972. [CrossRef]
49. Chen, H.; Liang, Y.; Huang, H.; Huang, Q.; Gu, W.; Liang, H. Live Pig-Weight Learning and Prediction Method Based on a Multilayer RBF Network. *Agriculture* **2023**, *13*, 253. [CrossRef]
50. Astill, J.; Dara, R.A.; Fraser, E.D.G.; Roberts, B.; Sharif, S. Smart Poultry Management: Smart Sensors, Big Data, and the Internet of Things. *Comput. Electron. Agric.* **2020**, *170*, 105291. [CrossRef]
51. Laguna, E.B.; Mun, H.-S.; Ampode, K.M.B.; Chem, V.; Kim, Y.-H.; Yang, C.-J. Artificial Intelligence for Automatic Monitoring of Respiratory Health Conditions in Smart Swine Farming. *Animals* **2023**, *13*, 1860. [CrossRef] [PubMed]
52. Li, B.; Xu, W.; Chen, T.; Cheng, J.; Shen, M. Recognition of Fine-Grained Sow Nursing Behavior Based on the SlowFast and Hidden Markov Models. *Comput. Electron. Agric.* **2023**, *210*, 107938. [CrossRef]
53. Paudel, S.; De Sousa, R.V.; Sharma, S.R.; Brown-Brandl, T. Deep Learning Models to Predict Finishing Pig Weight Using Point Clouds. *Animals* **2023**, *14*, 31. [CrossRef] [PubMed]
54. Collins, L.M.; Smith, L.M. Review: Smart Agri-Systems for the Pig Industry. *Animal* **2022**, *16*, 100518. [CrossRef] [PubMed]

55. Rudrakar, S.; Rughani, P. IoT Based Agriculture (Ag-IoT): A Detailed Study on Architecture, Security and Forensics. *Inf. Process. Agric.* **2023**, *in press*. [CrossRef]
56. He, Y.; Tiezzi, F.; Howard, J.; Maltecca, C. Predicting Body Weight in Growing Pigs from Feeding Behavior Data Using Machine Learning Algorithms. *Comput. Electron. Agric.* **2021**, *184*, 106085. [CrossRef]
57. Thomas, J.; Rousselière, Y.; Marcon, M.; Hémonic, A. Early Detection of Diarrhea in Weaned Piglets from Individual Feed, Water and Weighing Data. *Front. Anim. Sci.* **2021**, *2*, 688902. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.



Article

ApIsoT: An IoT Function Aggregation Mechanism for Detecting *Varroa* Infestation in *Apis mellifera* Species

Ana Isabel Caicedo Camayo, Martin Alexander Chaves Muñoz and Juan Carlos Corrales *

Telematics Engineering Group, Faculty of Electronics Engineering and Telecommunications, Universidad del Cauca, Popayán 190003, Colombia; anitacaicedo@unicauca.edu.co (A.I.C.C.); martinchaves@unicauca.edu.co (M.A.C.M.)

* Correspondence: jcorral@unicauca.edu.co

Abstract: In recent years, the global reduction in populations of the *Apis mellifera* species has generated a worrying deterioration in the production of essential foods for human consumption. This phenomenon threatens food security, as it reduces the pollination of vital crops, negatively affecting the health and stability of ecosystems. The three main factors generating the loss of the bee population are industrial agriculture, climate changes, and infectious diseases, mainly those of parasitic origin, such as the *Varroa destructor* mite. This article proposes an IoT system that uses accessible, efficient, low-cost devices for beekeepers in developing countries to monitor hives based on temperature, humidity, CO₂, and TVOC. The proposed solution incorporates nine-feature aggregation as a data preprocessing strategy to reduce redundancy and efficiently manage data storage on hardware with limited capabilities, which, combined with a machine learning model, improves mite detection. Finally, an evaluation of the energy consumption of the solution in each of its nodes, an analysis of the data traffic injected into the network, an assessment of the energy consumption of each implemented classification model, and, finally, a validation of the solution with experts is presented.

Keywords: machine learning; *Varroa destructor*; *Apis mellifera*; IoT; precision beekeeping; aggregation functions; energy consumption

Citation: Camayo, A.I.C.; Muñoz, M.A.C.; Corrales, J.C. ApIsoT: An IoT Function Aggregation Mechanism for Detecting *Varroa* Infestation in *Apis mellifera* Species. *Agriculture* **2024**, *14*, 846. <https://doi.org/10.3390/agriculture14060846>

Academic Editors: Gniewko Niedbala, Sebastian Kujawa, Tomasz Wojciechowski and Magdalena Piekutowska

Received: 8 April 2024
Revised: 2 May 2024
Accepted: 6 May 2024
Published: 28 May 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The *Apis mellifera* species is crucial in pollinating various plant species, contributing significantly to agricultural production and food security. According to data from the Food and Agriculture Organization of the United Nations (FAO), approximately 75% of the world's crops depend heavily on pollination [1]. This activity carried out by bees increases the production of fruits, vegetables, and seeds and improves the quality and diversity of crops. This function is essential for the sustainability of agricultural systems and biodiversity conservation. Furthermore, bees contribute significantly to climate change mitigation, as they promote the regeneration and maintenance of plant ecosystems, helping to absorb carbon dioxide and stabilize the global climate [2].

Western bee colony losses can vary significantly from year to year, but they are becoming worse overall. Annual studies of honey bee colony loss show considerable variation in temporal and spatial rates of colony loss, as well as contributing factors [3,4]. Some studies such as [5] report that the annual colony loss rate of *Apis mellifera* in Latin America varied between 17% and 48%, depending on the country and the year. The percentage of colony loss of meliponine bees varied between 33% and 46% annually. *Varroa destructor* mites and the viruses that spread these parasites are now considered the most significant contributors to these losses. However, queen problems, such as a lack of natural food sources, the expansion of urban areas, and excessive use of pesticides, are also factors [6]. This phenomenon threatens global food security by reducing pollination of vital crops. At the same time, declining bee populations negatively impact the health and stability of ecosystems, compromising their ability to provide critical ecosystem services.

The *Varroa destructor* mite, along with the viruses it spreads, is an important catalyst for this decline. This threat generates negative impacts at an individual and collective level, compromising bees' immunological and nutritional health. It causes physical injuries and promotes the spread of fungi, bacteria, and viruses within the colonies it infests since it acts as a vector of microorganisms [3]. The consequences of *Varroa* infestation significantly impact beekeeping and food security since it has contributed to the decline of bee populations worldwide.

For beekeepers, it is crucial to detect the presence of the mite early and accurately to prevent its spread and protect the health and well-being of bees. Over the years, artisanal techniques have been practiced, such as the *Varroa* test with isopropyl alcohol, considered the most precise and effective, where bees are immersed in a mixture of water and alcohol to count the mites present and establish a percentage of infestation [7]. However, this method can have disadvantages, such as non-representative samples and instant death of bees [8]. Another method is the sanitary floor, which involves replacing the traditional floor of the hives with one that collects fallen mites. However, it has disadvantages, such as damage to the apiaries, and the effectiveness of the test is variable [8]. Finally, the *Varroa* test with powdered sugar involves shaking a colony sample in a jar with sugar and then counting the mites released during the process, which causes stress and death of the bees evaluated [8].

To address this problem, IoT-based monitoring systems and artificial intelligence algorithms are integrated into beekeeping practices [9,10]. This is closely linked to the concept of precision beekeeping, which emphasizes the use of technologies such as advanced sensors, data analysis, and process automation for precise management of bee colonies [11].

This study proposes adopting an integrated solution based on the Internet of Things (IoT) and supervised learning as a critical element in detecting the presence of the *Varroa destructor* mite in hives. To this end, the aggregation of functions and their derived mechanisms are used as the primary strategy to develop a robust and efficient prototype capable of accurately identifying the presence of the mite. This proposal aims not only to guarantee and improve the quality of life of bees but also to support the work carried out by beekeepers and to significantly reduce the restrictions associated with battery, transmission, and data storage. The proposed hardware prototype uses low-cost and easy-to-achieve components, ensuring they are accessible to beekeepers.

The present work involved the following sections. Section 2 presents a systematic review of the available literature on emerging technologies used for hive monitoring and detecting the *Varroa destructor* mite. Section 3 identifies the occurrence of *Varroa* infestation based on discrete variables and captures functional and non-functional requirements; Section 4 shows the prototype's implementation and hardware and software development. In Section 5, we analyze and discuss the results. Finally, in Section 6, we expose the conclusions and future work.

2. Related Works

Following the adjusted methodology of Petersen [12] and Kitchenham [13], the existing research trends related to the use of IoT against detecting *Varroa* infestation in *Apis mellifera* species are presented.

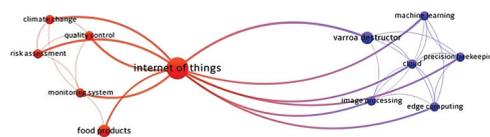
- Research question: based on the problem of mass mortality of bees [3] and seeking to know the technological proposals from the area of IoT and the aggregation of functions, we pose the following research question: What research exists within the IoT area that uses the aggregation of functions to monitor or detect *Varroa* infestation in *Apis mellifera* hives?
- Search strategies: information is collected through scientific databases, such as Scopus and Web of Science. Additionally, we use VOSviewer to analyze the resulting keywords.
- Selection of studies: the search string and the inclusion and exclusion criteria are defined to select the relevant articles (Table 1).

Table 1. Study selection criteria.

Search string	("bee" OR "honey bee" OR "apis mellifera") AND ("IoT" OR "internet of things") AND ("varroa" OR "varroa infestation")
Inclusion criteria	Data Aggregation, Multisensor Platform
Exclusion criteria	Artificial Bee Colony Algorithm, Food Products

Through this process, we found 11 related works subjected to keyword analysis to detect possible biases and areas of interest for research.

This article is formulated as a solution to the detection of *Varroa* infestation in species of *Apis mellifera*. For this reason, it is essential to analyze the works dedicated to this purpose. In Figure 1, two specific branches are distinguished. The first is dedicated to *Varroa* infestation detection solutions based on image processing, machine learning, edge, and cloud computing within the framework of precision beekeeping. The second identifies factors that intervene in the detection process, such as climate change, honey quality control, and hive monitoring. Finally, and as the central node, there is the IoT, an emerging technology that allows both gaps to be bridged into a comprehensive solution for the beekeeper.

**Figure 1.** Keyword mapping.

- **Gaps:** During the literature review, we found that there are currently no works related to feature aggregation in precision beekeeping. Furthermore, most of the works focus on emerging technologies such as deep learning [14], the cloud, and edge computing, where it is highlighted that the mechanisms proposed by the authors [15–17] are carried out in simulation environments. Although they can provide a general view of a system's performance, it is necessary to provide a broader view in a hardware environment.

In [18–21], the authors give priority to the implementation of the proposed solution in a natural environment, without regard for precision beekeeping systems, such as battery consumption and processing of large volumes of data. The works do not consider approaches such as analyzing the collected data and processing as a strategy for detecting complex events such as *Varroa* infestation.

Studies such as [17–21] demonstrate that it is possible to detect anomalies in the hive, including the *Varroa destructor* mite, using sensors that monitor physical variables. However, the proposed systems offer effectiveness levels greater than 75% and have proven to be minimally invasive techniques for bees. The authors highlight some aspects that have yet to be considered and are addressed in this research, such as incorporating commercial and low-cost sensors for temperature, humidity, CO_2 , and TVOC. In addition, a data processing strategy was implemented by aggregating functions and a classification model based on supervised learning to detect the mite.

Some of the sensors used by different authors specifically for detecting *Varroa* in bee hives are presented in Table 2.

Table 2. Sensors used for different physics variables.

Variable Physics	Sensors Used	Reference
CO ₂	GS4161	[20,22]
	TL6615	[21]
	SGP30, BME680	[19]
TVOC	SGP30, BME680	[19]
	TGS2600	[20]
Temperature	MCP9700A, DS18B20	[20,23]
	BME680	[19]
	DHT22	[21,24]
Humidity	DHT22	[21,24]
	DHT11	[23]
	BME680	[19]
	808H5V5	[20]

3. Materials and Methods

This section focuses on describing the four phases carried out for the research. The first focuses on characterizing the presence of the mite in terms of discrete variables; the second refers to the data preprocessing technique used, and the third presents the classification models based on supervised learning developed to provide a level of alertness—finally, the fourth focuses on integrating the solution into the selected hardware.

3.1. Characterization of the Appearance of *Varroa* Infestation

Several studies have shown that the behavior of some internal variables of the hive, such as temperature, humidity, and levels of gases such as CO₂ and TVOC, are crucial indicators of bee health and the risk of infections [21,23,25]. Both the *Varroa destructor* mite and the *Apis mellifera* species react differently to changes in internal temperature. In this context, the characterization considers the phoretic phase of *Varroa*, a stage in which the mite is outside the brood cells, and represents a critical moment for evaluating and controlling this threat [8].

Based on interviews with beekeepers in the region, as well as studies such as [25] focused on characterizing the conditions of different variables inside a hive and the presence of *Varroa* in Colombia, it was possible to identify that for variables such as temperature, the normal ranges inside a hive are between 32 °C and 36 °C, considering a maximum and minimum range between 20 °C and 37 °C, and with relative humidity between 50% and 75%. Meanwhile, for variables such as CO₂ and TVOC, the characterization is performed based on studies such as [19].

3.1.1. Temperature

Some research, such as [26], suggests that temperatures above 38 °C are critical for both bees and *Varroa*, who are vulnerable and cannot carry out their activities usually. If the temperature remains in that state for prolonged periods, it can cause death. *Varroa*'s reproduction rate and phoretic phase's activity phase decrease significantly in this case. Therefore, the risk of contracting *Varroa* infestation is estimated to be low under these conditions. Between 36 °C and 37.9 °C, bees enter a phase of thermal stress that causes excessive energy consumption and makes them especially vulnerable. *Varroa* can use this condition to reproduce even if it is not in the best conditions.

Between 33 °C and 35 °C, the ideal environment is established for the reproductive cycle of bees. This temperature also benefits *Varroa* since the females of this mite can enter the unprotected brood cells and reproduce. However, this temperature does not favor the phoretic phase of *Varroa* [15].

On the other hand, a temperature below 28 °C makes bees vulnerable due to their inactivity, which reduces their ability to defend themselves against possible threats. In this scenario, the phoretic phase of the *Varroa* develops to its maximum, allowing easy access to the brood cells. These cells, which maintain a higher internal temperature, facilitate the normal reproductive process of *Varroa* [27].

3.1.2. Humidity

Initially, research such as [20] suggests that a humidity level greater than 65% favors the reproduction and spread of the *Varroa destructor* mite, increasing the risk of infection. This extremely humid environment weakens the bees' immune systems and makes them more susceptible to infections, representing a high health risk. Secondly, a humidity level between 40% and 60% is considered moderate, as it balances bees, allowing them to use their natural defense mechanisms and reducing the risk of infection by *Varroa destructor*. This risk level is classified as medium [19].

Finally, a low humidity level, less than 40%, is considered a dry environment that can dehydrate bees and weaken their immune system, making them susceptible to certain infections. However, *Varroa* mites do not adapt quickly to this environment, so the risk of contracting *Varroa* infestation is low [20].

3.1.3. Concentration of Volatile Gases

Recent research [17,19,22] has found that the levels of different types of gases within a hive can be indicators of health, such as CO₂ and TVOC. These gases, produced by bees and other biological processes, can regulate the hive's internal environment and the bees' well-being. Research has established that TVOC and CO₂ levels vary considerably as the level of infestation increases or decreases. For this analysis, it is essential to highlight that gas levels depend on different factors, such as the number of bees in the hive, the quality of the outside air, and the hive's location. In this case, an average hive was used for beekeeping activities, that is, 40,000 to 60,000 bees. The hive that has not been parasitized with *Varroa* has a TVOC level ranging from 0 ppb to 300 ppb and CO₂ from 0 ppm to 600 ppm. A low *Varroa* infestation level has a TVOC level of 300 ppb to 600 ppb and CO₂ of 600 ppm to 1000 ppm. An average *Varroa* infestation level has a TVOC level of 600 ppb to 1500 ppb and CO₂ of 1000 ppm to 2000 ppm. A high *Varroa* infestation level has a TVOC level more fabulous than 1000 ppb and CO₂ greater than 2000 ppm. Finally, a hive receiving chemical treatment against *Varroa* has a TVOC level of 1900 ppb to 3500 ppb and CO₂ of 1000 ppm to 1600 ppm.

3.1.4. Identified Alerts

After analyzing the presence of the *Varroa* mite using various discrete variables, the information collected is synthesized to establish an alert level, whether very high, high, medium, or low, on the possible presence of the mite in the hive, as shown in the Table 3.

Subsequently, based on the previous description, the percentage of hive infestation is established according to each alert level [7,17,19,22,26].

Table 3 establishes the alert level of *Varroa* infestation in a hive in terms of percentages. This considers the result provide by [7] and the information delivered by expert beekeepers in interest. The data provided by the sources are mainly based on *Varroa* tests using different methods and hives of approximately 40,000 to 60,000 bees.

Table 3. Alert-level classification based on discrete variables.

Variable	Range	Alert level	% of Infestation
Temperature (°C) Humidity (%) TVOC (ppb) CO ₂ (ppm)	Less than 28 Greater than 75 Greater than 1500 Greater than 2000	Very high	Greater than 10%
Temperature (°C) Humidity (%) TVOC (ppb) CO ₂ (ppm)	28–30 65–75 600–1500 1000–2000	High	Between 5% and 10%
Temperature (°C) Humidity (%) TVOC (ppb) CO ₂ (ppm)	30–33 50–65 300–600 600–1000	Medium	Between 2% and 5%
Temperature (°C) Humidity (%) TVOC (ppb) CO ₂ (ppm)	34–37 40–50 0–300 400–600	Low	Less than 2%

3.1.5. Requirements Exploration and Capture

A workshop was conducted with 16 beekeepers from the Beekeeping and Agroindustrial Association of Piendamó and Tunía (ASAPIT) to identify the needs and establish the solution’s requirements. The interviews accomplished in this workshop established that the majority knew the *Varroa destructor* mite and recognized the importance of variables such as temperature and humidity in its development. Although some do not apply preventive treatments, most prioritize hygiene and periodic check-ups. Additionally, beekeepers show interest in a remote system to monitor their hives, preferring weekly notifications (Figure 2).

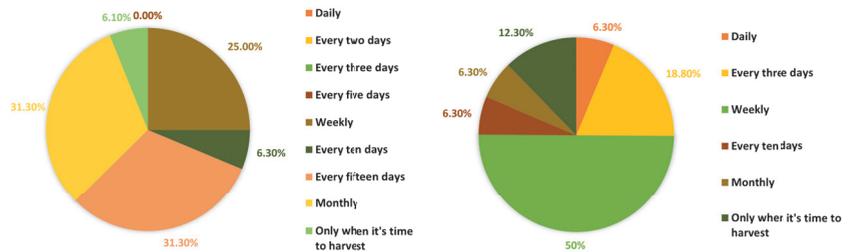


Figure 2. Apiary visit frequency and notification frequency preference among beekeepers.

Subsequently, considering the needs of beekeepers and the information selected, the functional and non-functional requirements of the system are established.

Functional requirements:

1. Capture environmental data such as temperature, humidity, CO₂, and TVOC.
2. Monitor the hive for an extended period.
3. Detect the presence of *Varroa* infestation from data collected over a week.
4. Determine and classify the *Varroa* infestation rate in a hive into four levels (very high, high, medium, or low).
5. Notify the beekeeper of said index and provide preventive information through a message.

Non-functional requirements:

1. Manage the battery to make the solution work continuously for a long time.
2. Ensure the connection between devices to transmit data.

3. Process data to avoid network saturation.
4. Manage data storage.
5. Adapt the system to different environments to function correctly in various environmental and climatic conditions.

3.2. Function Aggregation

Adding functions in an IoT system refers to applying, combining, and consolidating different functions or features in a system to improve efficiency, usefulness, and data management. This section discusses feature aggregation as a preprocessing strategy for captured data.

3.2.1. Centralized Feature Aggregation Mechanism

We base the proposed IoT solution on a centralized data aggregation mechanism, the main actors of which are the sensor, aggregator, and base station nodes. The focus of the solution is centralized data aggregation (CDA), and this focus marks the pause in its operation. The network diagram is presented below (Figure 3), with the hardware selected.

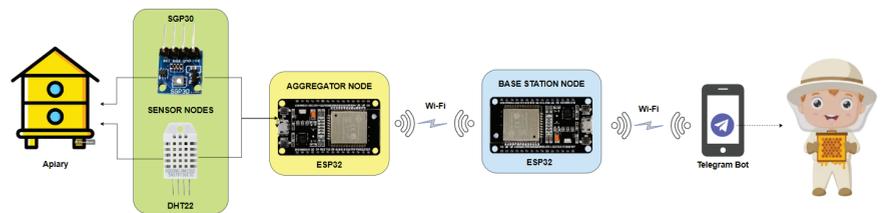


Figure 3. Network topology based on the centralized data aggregation model.

Figure 4 shows the system connection diagram for the sensor and aggregator nodes. In red, the connections related to the direct power supply of the source are observed; in orange, the connections that correspond to a voltage input regulated at 5 V; in black, the ground connection; in brown and violet, the data connection for the SGP30 carbon dioxide and volatile organic compounds gas sensor (Sensirion AG, Stäfa, Switzerland); in cyan, the data connection for the Temperature and humidity sensor DHT22 (Adafruit Industries, New York, NY, USA); in gold, the data connection for the Voltage sensor fz0430 (Analog Devices, Inc., Norwood, MA, USA). The aggregator node and the Base station node are made up of the ESP32 development board (Espressif Systems, Shanghai, China) [28].

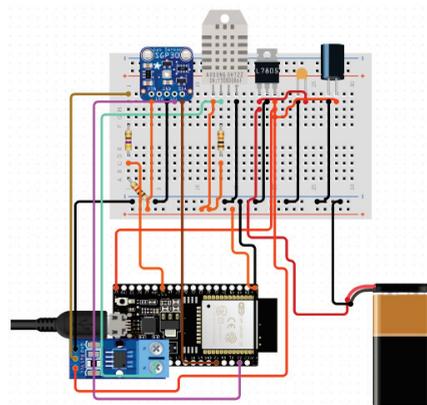


Figure 4. System connection diagram.

Two sensor nodes, DHT22 [24], are installed inside the apiary, specifically in the hive’s central frame. They provide temperature and humidity data and SGP30 [29] data at the CO₂ and TVOC levels. These nodes capture this information within the hive and transmit it to the aggregator node.

The aggregator node (AN) performs battery management, supported by the FZ0430 sensor [30]. It also guarantees the system’s storage. It takes the data captured by the sensor nodes, whose primary function is to apply the corresponding aggregation functions and transmit the aggregated data to the base station node (BSN).

For its part, the BSN receives the aggregated data and, through a weighted multi-criteria algorithm, normalizes the information that is subsequently sent to a classification model based on supervised learning, which determines the alert level in the hive; this alert is notified to the beekeeper via a text message in the Telegram application.

The need to use a BSN arises due to the nature of the centralized function aggregation mechanism, designed to generate a scalable solution in case it is necessary to monitor more than one hive, thus facilitating its management.

3.2.2. Battery Management

It is a crucial factor because all nodes in the network run on batteries and, therefore, have energy limitations. Battery life can directly impact system availability and performance. To cover this requirement, we propose the following statements:

- Limit system functionalities: the system takes a certain number of samples (Table 4) depending on the battery’s state before entering an energy-saving state.

Table 4. Battery management.

Battery	Description	Periodicity	# of Samples
Greater than 30%	Capture temperature samples	2 h	12
	Capture humidity samples	2 h	12
	Capture CO ₂ samples	12 h	60
	Capture TVOC samples	12 h	60
Less than 30%	Capture temperature samples	4 h	6
	Capture humidity samples	4 h	6
	Capture CO ₂ samples	12 h	30
	Capture TVOC samples	12 h	30

- Hibernation: Using predefined functions of the EP32 board, such as the *light_sleep* method, allows the sensor nodes to enter a hibernation or low-power state until it is time to capture a sample.
- Internet connection: The interconnection between the base station node and the aggregator node occurs only when data need to be sent; otherwise, the devices remain disconnected from the Internet. The solution focuses on minimizing the power consumption of the devices used, especially the internal Wi-Fi modules, reducing the connection only when transmitting or receiving information due to high consumption, according to the ESP32 board datasheet [28].

3.2.3. Transmission Management

It allows the network to reduce traffic overload and optimize battery and bandwidth resources, improving efficiency and minimizing errors and information loss.

- Information reduction: The amount of data captured is reduced, applying average (AVG) as an aggregation function. It consists of taking the data captured in one day and calculating their arithmetic mean, thus reducing the number of samples to a single significant sample per variable.
- Internet connection: The interconnection between the base station node and the aggregator node occurs only when data need to be sent; otherwise, the devices remain disconnected from the Internet.

- Connection with the beekeeper via Telegram: Telegram is chosen as a communication bridge with the beekeeper since it is a free and accessible application. Thanks to its functionality against bot management, it allows data transmission from the hive to the beekeeper.

3.2.4. Data Storage Management

It is an essential process for the system because it optimizes the ESP32 board flash memory and processing resources. It is necessary to guarantee the availability, integrity, and usefulness of the data. In addition to reducing the risk of loss of critical information, it accelerates the identification of relevant data and facilitates adaptation to changing environmental needs.

Figure 5 offers a visual representation of the system's process of eliminating data redundancy using aggregation functions. Below, we describe the operation in each of the nodes.

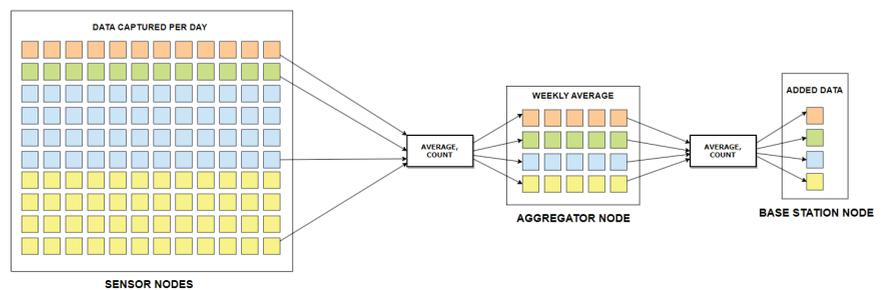


Figure 5. Data storage management.

- **Aggregator node:** This node is configured to capture all the data from the sensor nodes, where the data from the entire day are taken and processed at the end of the day through the AVG aggregation functions and COUNT. The processing afterward allows us to obtain the daily arithmetic mean of the four variables inside the hive for subsequent sending to the BSN. Applying these functions allows for freeing up the storage space of the node to prevent its saturation due to lack of memory and loss of information due to possible failures.
- **Base station node:** This node receives and stores the data corresponding to the arithmetic means of the four variables for each of the five monitoring days. It stores twenty points of data in its memory later converted into four significant variables using the COUNT and AVG functions. These data make up the input data vector for the classification model.

3.2.5. Weighted Multi-Criteria Aggregation Algorithm

It consists of assigning a weight to each variable according to its relevance and calculating a score [31]. The sum of the individual scores of each variable will be the basis for detecting the alert level of the hive [32]. This process is commonly used in various areas of decision-making since it allows for establishing priority classes [33].

For its implementation, initially, the priority level of each variable is defined, with one being the most relevant and four being the least relevant (Equation (1)); temperature is defined as the most pertinent factor (1), followed by humidity (2), CO₂ (3), and TVOC (4). Subsequently, the maximum and minimum values each variable can take are identified, and the range is calculated, which refers to the difference between the maximum and minimum values. The MAX and MIN aggregation functions are used (Equation (2)).

$$W_j = \frac{\frac{1}{r_j}}{\sum_{i=1}^n \frac{1}{r_i}} \quad (1)$$

$$\text{subtraction} = x_{MAX} - x_{MIN} \quad (2)$$

The priority of each variable was determined considering the results of the systematic review carried out in Section 2 and the capture of requirements applied to ASAPIT beekeepers.

Once the initial data have been calculated, the score or normalized score is identified, assigning a value between 0 and 1 to the value of the captured sample, considering the causal relationship between each variable and the infestation due to *Varroa* infestation [34]. According to the classification of alert levels, Table 3, the score is calculated considering that the temperature variable indicates a low *Varroa* infestation level since it takes higher values (Equation (3)). Likewise, the humidity variables, CO_2 and TVOC, indicate a low level of infestation, which lowers the value taken from the sample (Equation (4)).

$$x_{ij}^n = \frac{MAXx_{ij} - x_{ij}}{MAXx_{ij} - MINx_{ij}} \quad (3)$$

$$x_{ij}^n = \frac{x_{ij} - MINx_{ij}}{MAXx_{ij} - MINx_{ij}} \quad (4)$$

The product between the score and the weight assigned to each variable determines the normalized score for each sample. The basis for the alert level classification is determined by taking the weighted sum of the individual contributions of the four variables. Table 5 and Figure 6 illustrate the algorithm's application [35].

Using the weighted multi-criteria algorithm allows the collected data to be transformed into normalized data within the range of zero to one. It is less complex for the classification model to identify patterns and alert labels, significantly reducing the computational costs, execution time, and energy consumption [34].

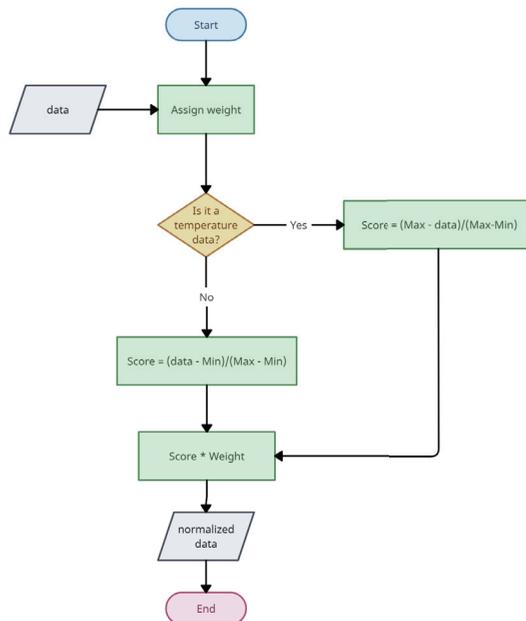


Figure 6. Flowchart of weighted multi-criteria algorithm.

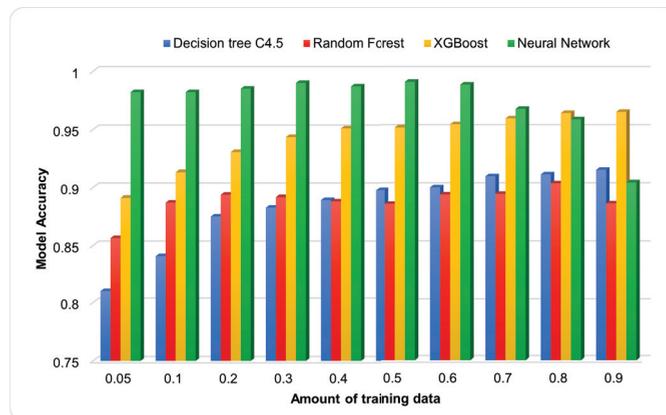
Table 5. Application of the weighted multi-criteria algorithm.

Variable	Weight	Sample	Score	Score × Weight
Temperature	0.48	22.5	0.8529411765	0.4094117647
Humidity	0.24	54.8	0.396	0.09504
CO ₂	0.16	600	0.0769230769	0.0123076923
TVOC	0.12	200	0.08	0.0096

3.3. Classification Models

Using classification models based on supervised learning to detect alerts regarding the health status of hives offers a systematic and automated approach to performing tasks. These models can learn complex patterns from a training dataset, allowing them to make accurate predictions on new data. The ability to process complex data makes machine learning (ML) algorithms a robust and adaptable option for classification in hive monitoring systems, overcoming the limitations of traditional classification methods, such as heuristic rules or expert-based systems.

We divided the models into “white-box” and “black-box” models, representing two contrasting ML approaches (Figure 7). White-box models, such as decision trees, Random Forest, and Gradient Boosting [36], are transparent about how they arrive at their decisions, making them easily interpretable; they provide a clear view of how input features relate to predictions, making the classification process easier to understand. On the other hand, black-box models, such as neural networks, are opaque about their internal workings and do not directly explain their decisions. However, they can capture complex pat-terns and relationships in data that white box models may miss [37].

**Figure 7.** Comparison of classification models.

To date, we have not found a database with actual data that capture the presence of *Varroa* in hives, considering the discrete variables mentioned above. Therefore, we built a synthetic dataset with 10,000 instances [38] characterizing the hives’ discrete variables following the CRISP-ML(Q) model. The dataset was generated using a Python script designed specifically for this purpose. The script was executed with a series of configurable parameters that allowed the ranges of each class to be adjusted (alert levels considered in Table 3). A descriptive statistical approach was used to examine the class distribution, considering specialized libraries such as Pandas and NumPy for data analysis [39]. As discussed later in this paper, the percentage of the dataset used to train the classification algorithms is an important characteristic, considering the processing and storage capabilities of the hardware used to model deployment. Then, from the dataset built, the C4.5 Decision

Tree models were implemented [40] and Random Forest [41], XGBoost (Extreme Gradient Boosting) (v2.0.3) [36], and neural network [42] algorithms were trained. Figure 5 shows the result of the analysis of the size variation of the training dataset. This analysis shows that white-box models are efficient for the research context because 40% of the data were used for training and 60% for validation. On the other hand, for the black-box model, 20% of the data were assigned for training and 80% for validation.

The original data presented in the study are openly available in *Varroa detection with discrete variables* on Kaggle at [<https://goo.su/39cnh>, accessed on 28 April 2024] or [38].

3.4. Hardware Configuration

This section focuses on developing and implementing the solution, considering the essential pillars of IoT. We emphasize the materials and hardware configuration for integrating the classification models.

With the established requirements, the IoT solution proposal is based on accessible, efficient, low-cost devices for beekeepers in developing countries. The functional requirements guided the construction of an IoT system to monitor bee hives, capturing variables such as temperature, humidity, CO_2 , and TVOC for approximately three weeks, with weekly notifications on the status of *Varroa* infestation. A detailed analysis of non-functional hardware and software requirements is needed to achieve these functionalities, including battery consumption and data storage and processing management. Specific sensors, such as the DHT22 for temperature and humidity, the SGP30 for gas concentration, and the FZ0430 for monitoring battery consumption, were selected. The ESP32 board [28] was used for data transmission. Given its compatibility and efficiency in energy consumption, we used the Arduino IDE environment (v2.2.0).

3.5. Implementation

Considering the five-layer IoT architecture model for constructing the Internet of Things solutions, the components are identified by layer and represented by a flow diagram (Figure 8). The IoT architecture model provides five defined layers; the perception layer is where the base hardware for the solution is located, such as the aggregator node and the sensor nodes. The network layer transmits information between nodes through Wi-Fi and system-beekeeper communication through the Telegram application. Transversally, there is the battery, which, although part of the perception layer, is in parallel since this component powers the layer to which it belongs and the network layer, mainly the base station node.

The middleware layer is the heart of the solution. It contains the aggregation functions responsible for manipulating and processing the data collected by the components of the perception layer. Within this third layer is the detection mechanism against *Varroa* infestation, which is the most critical aggregation function for the beekeeper. The application and business layers go hand in hand and are superficial to middleware in that they take the output from the third layer, which is generally an alert-level label, and are responsible for transmitting and documenting it for the end customer. The application layer connects through the network layer with the Telegram application, sending the beekeeper a message to his cell phone with the respective alert level of the hive and the relevant recommendations to act against it (business layer). In this way, the beekeeper can act or make decisions regarding his hive to ensure the health of his colony.

Once the topology, devices, and functional and non-functional requirements of the system have been defined, the scripts for configuring the aggregator and Base Station nodes are developed.

- Hardware configuration of the aggregator node: The aggregator node consists of the configuration of the sensors (SGP30, DHT22) and the ESP32 microcontroller board, as shown in Figure 3. In addition, the FZ0430 sensor has been integrated for sensor management. Following the principles of structured programming, the necessary libraries for the sensors, the Wi-Fi connection, and the transmission protocol (HTTP

client/server) are included. In the main loop, the functionalities of the aggregator node are configured, and the battery level is measured before each cycle. With these data, the daily measurement cycle begins, and at the end, the aggregation functions (COUNT and AVG) are applied to send the data to the base station node using HTTP POST. In addition, different exceptions are included to manage possible errors.

- Hardware configuration of the base station node: This node comprises only the ESP32 board, which communicates with the Telegram mobile application through the bot function, as shown in Figure 3. The node includes libraries for connecting to the Internet via Wi-Fi, running the classification model, communicating with the aggregator node, and sending data to the user on Telegram. Wi-Fi network credentials are initially set, and variables are created to store readings and monitoring status. The “sendTelegramMessage” function is configured to send messages via the Telegram API. This function uses the HttpClient class to request HTTP using the POST method. In the setup of the code, the configuration for the Wi-Fi connection is established, and the connection with the client (aggregator node) is expected; at the same time, the parameters of the classification model are established. In the main loop, the received data are processed, and it is verified that the variable readings are within safe ranges; then, Telegram will send alerts. After five days of monitoring, the averages of the accumulated readings are calculated, and the weighted multicriteria aggregation algorithm is applied, accompanied by the classification model, to determine the alert level. Finally, a detailed report with the alert level and recommendations for the beekeeper are sent to Telegram.

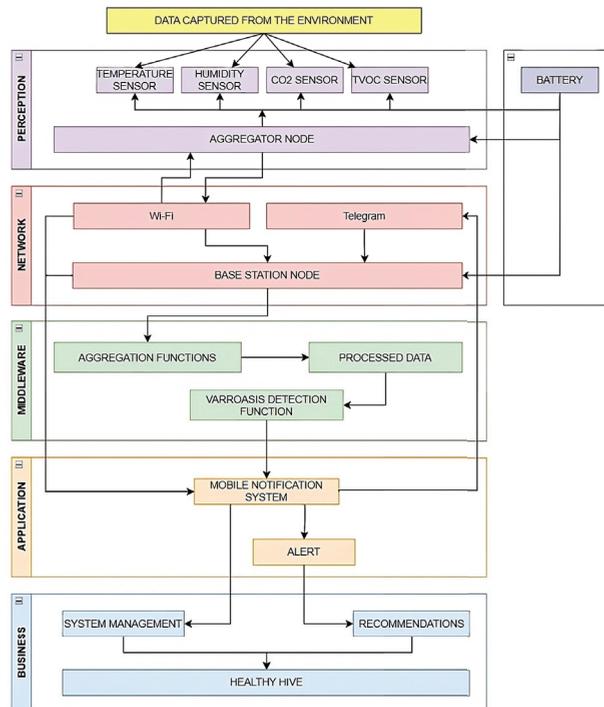


Figure 8. Data flow diagram based on IoT architecture model.

- Integration of classification models in hardware: First, the weighted multicriteria algorithm is applied to integrate the classification model in the hardware (Figure 9), and the generated dataset is processed using the Google Colaboratory development environment. Later, *TensorFlow Eloquent*, a Python library recognized for simplifying the creation of machine learning models in TensorFlow, was used, following the CRISP-ML(Q) process. In addition, to adapt to microcontrollers, the *micromlgen* library was used to translate the classification models from Python (v3.10.12) to C++. The models were implemented in the Arduino IDE environment [43]. An .h file was created for the white-box models and a hexadecimal file for the black-box model. In it, the trained and previously translated model was transcribed using a C++ script. The necessary namespaces were declared in the core sketch (.ino), and it was specified that the trained model should be traversed once the input data vector was received. This ensures continuous improvement, smooth implementation, and efficient execution of the classification model in the hardware environment.

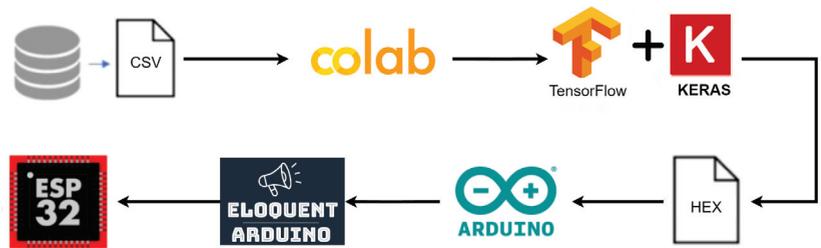


Figure 9. Process of integrating classification models with hardware.

3.6. Description and Approach to Testing

We proposed creating two test scenarios (Table 6) to develop and execute tests.

Table 6. Test scenarios for the aggregator node.

S1	Cycle with Function Aggregation	24 h cycle with the implementation of aggregation functions and battery optimization.
S2	Cycle without Function Aggregation	24 h cycle excluding the use of aggregation and battery optimization functions.

Scenario 1 includes all the functionalities implemented in Section 3.2, which refers to the different aggregation and optimization functions for the transmission processes, storage management, and battery management in the aggregator and base station nodes. Finally, it determines an alert level. Scenario 2 excludes the aggregation and optimization functions. Its operation is based on sending and receiving data from the aggregator node to the base station node. The base station node finally runs the classification model and sets the alert level.

Table 7 specifies the metrics used to evaluate the performance of each scenario in each of the tests.

Table 7. Description of evaluation metrics.

Test	Evaluation Metrics	Description
Evaluation of the energy consumption of the solution in each of its nodes	Battery consumption in (mAh)	Battery consumption in each scenario for the BSN and AN based on data provided by datasheets and hardware measurements.
Analysis of data traffic injected into the network	Total number of packages sent	Simulation of packet traffic in a period of 5 days between the AN and BSN for each scenario.
Analysis of hardware memory occupancy	Memory usage percentage for classification model	Total percentage of flash memory used by the ESP32 board using the Arduino Millis function.
Evaluation of the energy consumption of each classification model	Run time in seconds and Battery consumption in (mAh)	Run the selected classification model and determine total execution time to estimate the total board consumption when performing this task.

4. Results

This section initially presents the scenarios proposed for the validation tests and also includes results from the implementation of the developed IoT solution, including an exhaustive validation focused on system optimization and the critical advantages of function aggregation. We established five testing perspectives to evaluate the solution.

4.1. Evaluation of the Energy Consumption of the Solution in Each of Its Nodes

Battery performance tests are performed on each node, considering their specific roles directly affecting battery consumption and duration. For this purpose, information provided by manufacturers through datasheets and direct measurements performed on hardware are used.

Aggregator Node: This test evaluates the performance of the node's battery with an external power supply or battery. The objective is to analyze the battery capacity concerning operating time, measured in milliampere-hours (mAh). The power consumption of each node component at specific times, both in active and idle mode, is examined. Two scenarios are considered for comparison to understand the energy impact of function aggregation.

Next, total consumption is calculated, considering each component's duration in seconds (s) in each operating mode for 24 h.

From Table 8, for the active mode in both scenarios, the consumption of the sensor is low since samples are taken for short periods. In Scenario 1, the Wi-Fi module wakes up once a day for 10 seconds to send data, while in Scenario 2, all components, including the Wi-Fi module, are constantly active, quickly depleting the power battery. To the idle mode: although the active mode consumption of the components is higher, the aggregator node remains in active mode for only 0.2% of the total time. It makes the total amount of milliamp-hours (mAh) consumed in an idle mode much higher since this mode operates 9.8% of the time.

Table 8. Total consumption in mAh and total time in seconds of the aggregator node.

Scenario	Total Active Time (s)	Total Idle Time (s)	Consumption Active (mA)	Consumption Idle	Consumption (mAh)
S1	143	86257	3.3455555	224.312553	9.485754618
S2	798	85602	59.566388	857.6682	38.218108

In Scenario 1, the consumption in idle mode for all sensors, with the low-power *light_sleep* functionality, is practically zero, on the order of μA . The board's internal volt-

age regulator primarily results in battery consumption. In contrast, Scenario Two does not implement low-power features during idle periods, resulting in significant battery consumption and preventing long battery life. Now, an analysis of the performance and duration of commercial batteries that can power the aggregator node is carried out, adapting to the necessary specifications according to the needs of beekeepers. Three battery options are presented. A commercial 9V lithium battery with a capacity of 1200 mAh, with an L7805 voltage regulator to constantly provide 5 volts to the system. As a second option, four AA 1.5 V lithium batteries, with a combined capacity of 3600 mAh, also with an L7805 regulator. Finally, a 5 V power bank with a capacity of 4000 mAh [44].

Figure 10 shows the total operating duration in hours for each battery in each scenario. Battery 1 has the shortest operating duration, powering the node for a single cycle because its mAh capacity is low. In Scenario 1, it lasts approximately 5.25 days, while in Scenario 2, it only reaches 31 h of operation. Battery 2 lasts 15 days in the first scenario and 3.91 days in the second, which limits its ability to complete a monitoring cycle. Battery 3 showed better results in terms of duration. In Scenario 1, the total operation duration was 17.5 days, covering three monitoring cycles, while in Scenario 2, it spanned 4.33 days.

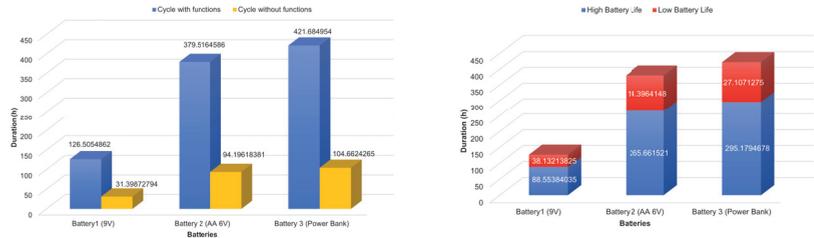


Figure 10. Total battery life in the aggregator node in the two scenarios and total battery life.

Base Station Node: The node comprises the CPU and the Wi-Fi module of the board, which stores, processes, and sends data. Batteries or the mains can power it, regardless of the hive’s location. Power consumption is calculated and compared in active and idle modes according to the scenarios.

From Table 9, in active mode, it is observed that in Scenario 1, the board is activated once a day to receive data and send a cycle initialization message via Telegram. In Scenario 2, it is activated several times while the Wi-Fi module is in receive mode to receive messages from the base station node. In both cases, data reception is the leading power consumer. Integrating the optimization function “*Light_Sleep*” in idle mode significantly reduces the board’s power consumption with minimal drain on the internal voltage regulator. In contrast, Scenario 2 lacks power-saving features during idle, resulting in constant power consumption that affects battery life.

Table 9. Total consumption in mAh and total time in seconds of the base station node.

Scenario	Total Active Time (s)	Total Idle Time (s)	Consumption Active	Consumption Active	Consumption (mAh)
S1	25	86,375	1.35	235.1354	9.7973
S2	1330	8570	44.125	852.114	35.504

4.2. Analysis of Data Traffic Injected into the Network

Data transmission performance tests are carried out to test the premise of reducing traffic injected into the network through function aggregation. We used Wireshark as a tool for network traffic analysis. It allows the transmitted data to be captured and examined in real time, offering a detailed view of the information flow and the protocols used [45].

Considering Table 6, two test scenarios are again analyzed. In this case, the simulation of a five-day cycle is carried out.

Figure 11 presents the total packet traffic between the nodes during each scenario’s data transmission and reception processes. In this test, the tool captured 150 packets over a 5-day cycle in Scenario 1, setting an average of approximately 30 packets daily. In contrast, in Scenario 2, 2887 packets were captured, equivalent to 578 per day.

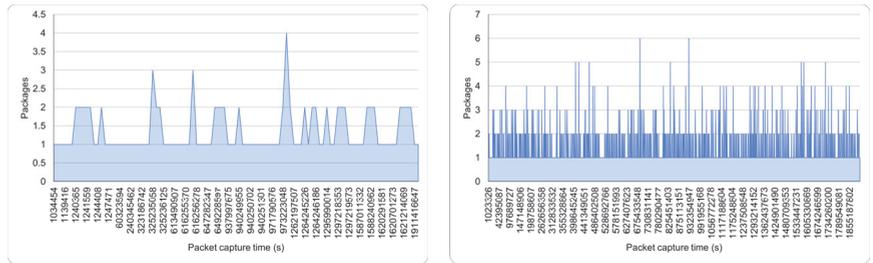


Figure 11. Network traffic to test scenario 1 and network traffic to test scenario 2.

4.3. Analysis of Hardware Memory Occupancy Concerning the Classification Model Used

This test comprehensively examines how implementing a high-accuracy classification model intended for *Varroa destructor* mite detection impacts the flash memory occupancy of the network BSN. It allows informed decisions regarding model selection and dataset size for respective training.

Figure 12 shows that tree-based models increase their size and memory footprint as the training dataset grows. In contrast, the neural network-based model maintains a stable memory occupancy of approximately 75%, regardless of the training dataset size. To guarantee the correct operation of the BSN CPU, up to 85% of the flash memory can be used, leaving the remaining 15% for executing additional tasks [28].

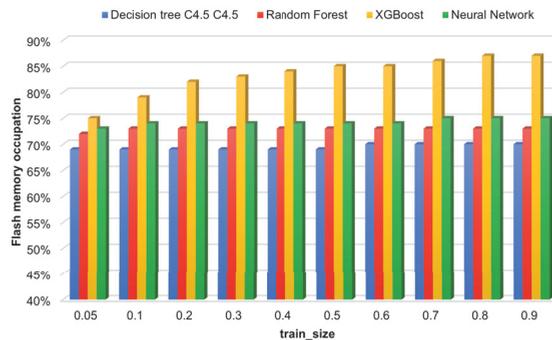


Figure 12. Flash memory consumption in relation to the training dataset size for each model.

Figure 13 shows the relationship between flash memory occupancy and accuracy for each implemented model. It is highlighted that the neural network is the model that exhibits the best performance regarding the relationship between memory resource constraints and accuracy. In contrast, tree-based models, such as XGBoost, show a high level of accuracy but consume a considerable amount of flash memory resources, reaching their maximum limit. This situation affects the performance of the BSN during the execution of daily tasks. On the other hand, the C4.5 Tree and Random Forest models are discarded due to their low accuracy in the classification task, even though they require a low amount of flash memory for their implementation.

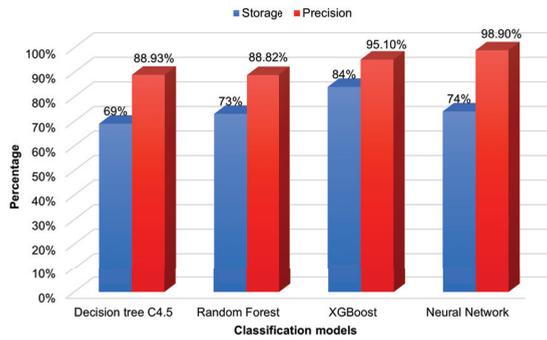


Figure 13. Flash memory consumption about the accuracy of the implemented classification models.

4.4. Evaluation of the Energy Consumption of Each Implemented Classification Model

This section compares the energy consumption of the four classification models to understand how their execution impacts energy consumption and determine the best solution. For this purpose, a script is proposed using Arduino IDE, which allows each model to execute a certain number of classifications continuously. It offers a vision regarding the time necessary for the card to complete the classification process and thus estimate your energy consumption.

From Table 10 and Figure 14, among the three white-box classification models, Tree C4.5 shows superior performance in terms of battery consumption and lower execution time; this model, being a single tree, can perform classifications faster compared to Random Forest and XGBoost models and employ multiple trees for the classification process. The difference between these two models is that in Random Forest, the prediction is performed by averaging or voting the predictions of each tree individually, which can be more efficient in terms of runtime [46]. In contrast, in XGBoost, the prediction process involves summing the predictions of multiple sequential trees and applying additional regularization processes [47].

Table 10. Classification time of the model’s dataset.

Model	Average Runtime (ms)	Total Time (h)
Tree C4.5	123.246	0.34235
Random Forest	129.825	0.360625
XGBoost	143.791	0.39941944
Neural Network	124.014	0.34449166

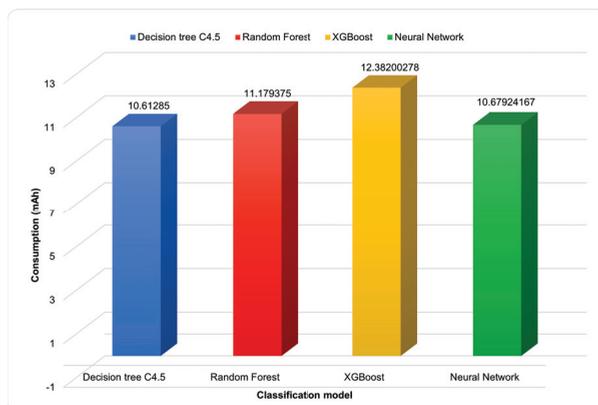


Figure 14. Total consumption in milliamp-hours for the classification of the dataset of each model.

5. Discussion

By implementing function aggregation on the aggregator node, such as AVG and COUNT for Scenario 1, the captured data are synthesized at the end of the day, temporarily storing the data in flash memory and sending the data only once. In comparison, in Scenario 2, where the data are sent immediately after their capture via Wi-Fi, higher energy consumption is observed, the main differentiating factor in the consumption in active mode in the aggregator node. Furthermore, implementing optimization features such as *light_sleep* on the card reduces battery consumption significantly during idle periods for the two nodes, in contrast to Scenario 2, where their absence makes a notable difference.

Based on the established battery management processes, the total battery lifetime is determined by considering operating conditions when the battery charge is greater than or less than 30%. Battery 1 operates for four days in high and one day in low battery mode, completing only one cycle. Battery 2 lasts eleven days in high and four days in low, equivalent to three complete cycles. On the other hand, battery 3 operates for twelve days in high battery mode and five days in low battery mode, covering three full cycles. During the remaining time, notifications about abnormal ranges or sensor problems and recommendations are sent to beekeepers.

The traffic analysis results show that function aggregation effectively controls the amount of data transmitted, reducing approximately 95% of the total data load. However, considerable packet loss is observed in Scenario 2, which highlights the importance of implementing feature aggregation to improve the reliability and efficiency of network communication. Regarding the size of the training datasets of the models and their implementation, it was decided to select a size of 0.4 for tree-based models, equivalent to 40% of the data for training and 60% for validation. The choice is since the XGBoost model exhibits the highest classification accuracy (94%) and reaches the possible memory limit since it occupies 84% of the flash memory, compared to the other two models (Random Forest and Tree C4.5), whose memory occupancies range from 69% to 73%. For the black-box model, a training data distribution of 20% was chosen, and the remaining 80% was reserved for validation. It resulted in a flash memory occupancy of 74%, which resulted in an accuracy of 99%.

Maintaining an appropriate balance between model accuracy and memory occupancy must be considered. The results emphasize the need to find an optimal point where the model accuracy is high while minimizing the consumption of memory resources. This balance ensures the system's efficient operation in practical environments when facing resource limitations such as those in the BSN.

When comparing the white-box models with the black box, it is observed that the neural network offers fast and efficient battery consumption performance and is on par with the Decision Tree. The results obtained from this comparison reveal the importance of considering both predictive performance and resource consumption when selecting a classification model for practical applications. While the C4.5 Decision Tree emerges as a promising option in terms of efficiency in battery consumption and execution time, the neural network also offers competitive and efficient performance.

Implementing a classification model on the ESP32 [28] microcontroller board represented a significant challenge in practice, mainly because some of the mechanisms used as a basis from other research available in repositories or developer forums needed to meet this research's requirements satisfactorily. While some approaches differed in nature or scope, others needed more flexibility to adapt to the project's specific demand. Despite this, the advantages of white-box and black-box models in terms of accuracy, scalability, and interpretability are considered, such as heuristic models or models based on expert knowledge. The research focused on exploring various alternatives designed to address the limitations above and optimize model performance on the board.

After exhaustive research in multiple sources, it was determined that incorporating libraries such as *EloquentTinyML* or *tflm*, developed by [48], surpasses other alternatives, as they are designed for devices with limited resources. These libraries optimize the use of memory and processing. Therefore, this study considers them and is valid as a complete

solution that addresses all the identified limitations. Additionally, they offer a wide range of functionality, integrating multiple machine learning models and enabling sophisticated and versatile solutions in embedded systems.

6. Conclusions and Future Works

This research characterized the climatic conditions affecting the presence of *Varroa* mites in bee hives, using discrete variables as critical indicators. Significant findings were obtained through a detailed analysis of climatic data, mite infestation records, and expert input. Potential limitations were noted, such as the lack of available information and studies, and caution was made against natural climate variability that may affect data interpretation. Although climatic patterns related to the *Varroa* mite were identified, it was highlighted that correlation does not imply direct causation. In addition, the possible influence of other factors not considered was mentioned, such as human activity and the use of chemical treatments in hives.

Throughout this research, we encountered limitations when selecting and implementing devices to monitor *Varroa* mites. To address these limitations, we added multiple additional functions. This allowed for more efficient management of monitoring cycles in the hives, adjusting them according to beekeepers' needs. Additionally, we conducted a detailed analysis of power consumption to design a solution that minimized battery usage without compromising functionality and accuracy. The integration of these functions facilitated effective management of data capture and transmission times, resulting in maximum battery savings from the card and, simultaneously, a significant reduction in network traffic. Finally, thanks to these efforts, we managed to reduce the battery consumption of the devices by up to 75% compared to implementation without these functions and reduce network traffic by 95%.

Integrating functions as part of our data preprocessing strategy significantly reduced redundancy, reducing the amount of data from 720 to a vector of 4 highly representative aggregated data points. This approach not only optimized storage on limited hardware but also allowed for more efficient management thereof. By combining this hardware with a supervised machine learning model, specifically a neural network, we developed a comprehensive solution for *Varroa* mite detection. Of all the models tested, the neural network was the most suitable choice, as it allowed us to strike a balance between high accuracy (99%) and low hardware memory consumption (74%). This lays the groundwork for future research in precision beekeeping and automated monitoring of bee health, seeking to overcome the limitations of traditional detection methods, such as *Varroa* testing, which are laborious and require high human intervention.

Considering the interest of beekeepers in implementing more sensors and devices in the hives and developing additional functionalities based on the collected data, a future approach to designing and creating a more complete and adaptable monitoring and control system is suggested. This system could integrate additional sensors to capture relevant data such as daily bee flow and hive weight and develop advanced algorithms to identify complex patterns related to colony health.

As possible future research work, developing a more robust predictive model for forecasting *Varroa* mite occurrence in beehives is suggested. It implies considering other climatic constraints, parameters, and variabilities, such as local topography, seasonal variability, and human activity related to beekeepers' hive management.

In addition, long-term studies are proposed to validate the model's effectiveness in different regions and climatic conditions, improving the understanding of the relationship between climate and mite infestation.

Author Contributions: A.I.C.C., M.A.C.M. and J.C.C. proposed the concept of this research. A.I.C.C., M.A.C.M. and J.C.C. contributed to the state of art and final paper draft revisions. A.I.C.C., M.A.C.M. and J.C.C. wrote the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research is funded by Universidad del Cauca, Popayán, Colombia.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Acknowledgments: We would like to thank the Universidad del Cauca and the Beekeeping and Agroindustrial Association (ASAPIT) for participating in the data collection and for providing the interviews and feedback on this work.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CO ₂	Carbon Dioxide
TVOC	Total Volatile Organic Compounds
ASAPIT	Beekeeping and Agroindustrial Association of Piendamó and Tunía
CRISP-ML(Q)	Cross Industry Standard Process for Machine Learning
IoT	Internet of Things
ML	Machine Learning
AN	Aggregator Node
BSN	Station Base Node
CDA	Centralized Data Aggregation
AVG	Average
CPU	Central Processing Unit
XGBoost	Extreme Gradient Boosting

References

1. Food and Agriculture Organization of the United Nations (FAO). World Bee Day-Food and Agriculture Organization of the United Nations (FAO). Available online: <https://www.fao.org/world-bee-day/es> (accessed on 16 March 2024).
2. Rodríguez, D. Protection of bees for food sustainability in Colombia. *J. Apic. Res.* **2022**. Available online: <https://www.portafolio.co/tendencias/proteccion-de-abejas-para-las-sostenibilidad-alimentaria-de-colombia-569382> (accessed on 9 March 2023).
3. Stahlmann-Brown, P.; Hall, R.J.; Pragert, H.; Robertson, T. Varroa Appears to Drive Persistent Increases in New Zealand Colony Losses. *Insects* **2022**, *13*, 589. [CrossRef] [PubMed]
4. Brown, M.J.F.; Dicks, L.V.; Paxton, R.J.; Baldock, K.C.R.; Barron, A.B.; Chauzat, M.; Freitas, B.M.; Goulson, D.; Jepsen, S.; Kremen, C.; et al. A horizon scan of future threats and opportunities for pollinators and pollination. *PeerJ* **2016**, *4*, e2249. [CrossRef] [PubMed]
5. Requier, F.; Antúnez, K.; Porrini, M.P. Pérdidas de Colonias de Abejas Melíferas y Abejas Meliponinas en América Latina en el Periodo 2016–2018. In *Libro Resúmenes 2020: XIV Congreso Latinoamericano de Apicultura*; Federación Internacional Latinoamericana de Apicultura: Santiago, Chile, 2020. Available online: <https://ri.conicet.gov.ar/handle/11336/186909#:~:text=Los%20resultados%20informan%20que%20la,33%25%20y%2046%25%20anual> (accessed on 23 January 2024).
6. De Jong, D.; Lester, P.J. The global challenge of improving bee protection and health. *Front. Bee Sci.* **2023**, *1*, 1118292. [CrossRef]
7. AGROSAVIA-Colombian Agricultural Research Corporation. Monitoring of Varroa Destructor in Beekeeping Production Systems. Available online: <https://editorial.agrosavia.co/index.php/publicaciones/catalog/download/155/137/1091-1?inline=1> (accessed on 7 April 2024).
8. Dietemann, V.; Nazzi, F.; Martin, S.J.; Anderson, D.L.; Locke, B.; Delaplane, K.S.; Wauquiez, Q.; Tannahill, C.; Frey, E.; Ziegelmann, B.; et al. Standard methods for varroa research. *J. Apic. Res.* **2013**, *52*, 1–54. [CrossRef]
9. Zabasta, A.; Zhiravetska, A.; Kunicina, N.; Kondratjevs, K. Technical implementation of IoT concept for bee colony monitoring. In Proceedings of the 2019 8th Mediterranean Conference on Embedded Computing (MECO), Budva, Montenegro, 10–14 June 2019.
10. Edwards-Murphy, F.; Magno, M.; Whelan, P.M.; O'Halloran, J.; Popovici, E.M. b+wsn: Smart beehive with preliminary decision tree analysis for agriculture and honey bee health monitoring. *Comput. Electron. Agric.* **2016**, *124*, 211–219. [CrossRef]
11. Zacepins, A.; Brusbardis, V.; Meitalovs, J.; Stalidzans, E. Challenges in the development of Precision Beekeeping. *Biosyst. Eng.* **2015**, *130*, 60–71. [CrossRef]
12. Kai, P.; Feldt, R.; Mujtaba, S.; Mattsson, M. *Systematic Mapping Studies in Software Engineering*; BCS Learning Development: Karlskrona, Sweden, 2008. Available online: <https://www.scienceopen.com/hosted-document?doi=10.14236/ewic/EASE2008.8> (accessed on 25 October 2022). [CrossRef]
13. Barbara, K.; Charters, S. Guidelines for performing Systematic Literature Reviews in Software Engineering. In *Keele University and Durham University Joint Report*; EBSE 2007-001; Keele University: Durham, UK, 2007. Available online: https://www.elsevier.com/_data/promis_misc/525444systematicreviewsguide.pdf (accessed on 18 January 2024).
14. George, V.; Moraiti, A.; Kontogiannis, S. Deep Learning Beehive Monitoring System for Early Detection of the Varroa Mite. *Signals* **2022**, *3*, 506–523. [CrossRef]

15. Anna, W.; Pytlik, J.; Małyśiak-Mrozek, B.; Tokarz, K.; Mrozek, D. Edge Computing in IoT-Enabled Honeybee Monitoring for the Detection of Varroa Destructor. *Int. J. Appl. Math. Comput. Sci.* **2022**, *32*, 355–369. [CrossRef]
16. Dariusz, M.; Gorny, R.; Wachowicz, A.; Małyśiak-Mrozek, B. Edge-Based Detection of Varroosis in Beehives with IoT Devices with Embedded and TPU-Accelerated Machine Learning. *Appl. Sci.* **2021**, *11*, 11078. [CrossRef]
17. Francesco, B.; Turvani, G.; Garlando, U.; Riente, F. An Integrated Multi-Sensor System for Remote Bee Health Monitoring. In Proceedings of the 2022 IEEE Workshop on Metrology for Agriculture and Forestry (MetroAgriFor), Perugia, Italy, 3–5 November 2022; pp. 334–338. [CrossRef]
18. Bellos, C.V.; Fyrraridis, A.; Stergios, G.S.; Stefanou, K.A.; Kontogiannis, S. A Quality and disease control system for beekeeping. In Proceedings of the 2021 6th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM), Preveza, Greece, 24–26 September 2021; pp. 1–4. [CrossRef]
19. König, A. An in-hive soft sensor based on phase space features for *Varroa* infestation level estimation and treatment need detection. *J. Sens. Syst.* **2022**, *11*, 29–40. [CrossRef]
20. Murphy, F.E.; Magno, M.; Whelan, P.; Vici, E.P. b+WSN: Smart beehive for agriculture, environmental, and honey bee health monitoring—Preliminary results and analysis. In Proceedings of the 2015 IEEE Sensors Applications Symposium (SAS), Zadar, Croatia, 13–15 April 2015; pp. 1–6. [CrossRef]
21. Cecchi, S.; Terenzi, A.; Orcioni, S.; Spinsante, S.; Primiani, V.M.; Moglie, F.; Ruschioni, S.; Mattei, C.; Riolo, P.; Isidoro, N. Multi-sensor platform for real time measurements of honey bee hive parameters. *IOP Conf. Ser. Earth Environ. Sci.* **2019**, *275*, 012016. [CrossRef]
22. Szczurek, A.; Maciejewska, M.; Zajczek, Z.; Bąk, B.; Wilk, J.; Wilde, J.; Siuda, M. The Effectiveness of Varroa destructor Infestation Classification Using an E-Nose Depending on the Time of Day. *Sensors* **2020**, *20*, 2532. [CrossRef] [PubMed]
23. Martínez, R. Sistema Para el Monitoreo de Variables Físicas Dentro de una Colmena. *Pistas Educativas*. Available online: <https://pistaseducativas.celaya.tecnm.mx/index.php/pistas/article/view/1072> (accessed on 29 April 2024).
24. Whisnumurti, A.; Daru, A.; Hirzan, A. Temperature and Humidity Monitoring Using DHT22 Sensor and Cayenne API. *J. Transform.* **2020**, *17*, 209–214. ISSN: 2460-6731. [CrossRef]
25. Salamanca, G.; Osorio, M.; Rodríguez, N. Presencia e incidencia forética de Varroa destructor A. (Mesostigma: Varroidae) en colonias de abejas Apis mellifera (Hymenoptera: Apidae), en Colombia. *Zootec. Trop.* **2012**, *30*, 183–195.
26. The Queen Bee. Varroa in Bees: Symptoms and Treatment. Available online: <https://www.laabejareina.com/varroa/> (accessed on 27 January 2024).
27. Hou, C.S.; Li, B.B.; Deng, S.; Diao, Q.Y. Effects of Varroa destructor on temperature and humidity conditions and expression of energy metabolism genes in infested honeybee colonies. *Genet. Mol. Res.* **2016**, *15*, 15038997. [CrossRef] [PubMed]
28. Espressif Systems. ESP32—Espressif Systems. Available online: <https://www.espressif.com/en/products/socs/esp32> (accessed on 3 June 2023).
29. Sensirion. SGP30—Sensirion. Available online: <https://sensirion.com/products/catalog/SGP30> (accessed on 20 December 2023).
30. Ardobot. FZ0430 0V-25V Voltage Sensor Module. Available online: <https://www.ardobot.co/modulo-sensor-de-voltaje-fz0430-0v-25v.html> (accessed on 10 January 2024).
31. Dova, J.M.; Pelaez, J.I.; Holgado, A.; Hidalgo, O. Analysis of the genetic algorithm Gamc for multicriteria classification of inventories. In Proceedings of the KES'2000. Fourth International Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies. Proceedings (Cat. No.00TH8516), Brighton, UK, 30 August–1 September 2000; Volume 2, pp. 606–611. [CrossRef]
32. Sari, F.; Ceylan, D.A.; Özcan, M.M.; Özcan, M.M. A comparison of multicriteria decision analysis techniques for determining beekeeping suitability. *Apidologie* **2020**, *51*, 481–498. . [CrossRef]
33. Sergio, B.R.; Sanabria, D.F.N. Multi-Criteria Model Applied to Decision Making That Can Be Represented in Ishikawa Diagrams. Undergraduate Thesis, Universidad Distrital Francisco José de Caldas, Bogotá, DC, USA, 2018. Available online: <https://repository.udistrital.edu.co/bitstream/handle/11349/13894/BernalRomeroSergio2018.pdf?sequence=1&isAllowed=y> (accessed on 28 November 2023).
34. Hadhami, K.; Jabeur, K.; Ladhari, T. Genetic Algorithm to infer criteria weights for Multicriteria Inventory Classification. In Proceedings of the 2014 International Conference on Control, Decision and Information Technologies (CoDIT), Metz, France, 3–5 November 2014; pp. 276–281. [CrossRef]
35. Esteban, P.d.l.; David, F. Multicriteria decision-making method: Normalized weighted. *YouTube Video*. 2019. Available online: <https://www.youtube.com/watch?v=xIwhFc5lsg0> (accessed on 28 October 2023).
36. Bentéjac, C.; Csörgő, A.; Martínez-Muñoz, G. A comparative analysis of gradient boosting algorithms. *Artif. Intell. Rev.* **2021**, *54*, 1937–1967. [CrossRef]
37. Umit, C.; Kuzlu, M.; Pipattanasomporn, M.; Kempf, J.; Bai, L. Foundations of Big Data, Machine Learning, and Artificial Intelligence and Explainable Artificial Intelligence. In *Digitalization of Power Markets and Systems Using Energy Informatics*; Springer International Publishing: Cham, Switzerland, 2021; pp. 115–137. [CrossRef]
38. Caicedo Camayo, A.I. *Varroa Detection with Discrete Variables: Synthetic Dataset for Varroa Detection from Discrete and Normalized Variables*; Kaggle: Popayán, Colombia, 2024. Available online: <https://www.kaggle.com/datasets/16945694fbf6c04c6e6d51b52a06548ba8fc586647c632f9d65c274106112271> (accessed on 4 April 2024).

39. McKinney, W. *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython*; O'Reilly Media: New York, NY, USA, 2017.
40. Sonia, S.; Giri, M. Comparative Study Id3, Cart And C4.5 Decision Tree Algorithm: A Survey. *Int. J. Adv. Inf. Sci. Technol. IJAIST* **2014**, *3*, 97–103. [CrossRef]
41. Ghimire, B.; Rogan, J.; Galiano, V.; Panday, P.; Neeti, N. An evaluation of bagging, boosting, and random forests for land-cover classification in Cape Cod, Massachusetts, USA. *Gisci. Remote. Sens.* **2012**, *49*, 623–643. [CrossRef]
42. Stergiou, C.; Siganos, D. Neural Networks. Surveys and Presentations in Information Systems Engineering. *SURPRISE 96 Journal* 2006. Available online: <https://srii.sou.edu.ge/neural-networks.pdf> (accessed on 7 December 2023).
43. Arduino. Eloquenttinyml Documentation. Available online: <https://github.com/eloquentarduino/EloquentTinyML> (accessed on 8 January 2024).
44. Avido. Avido 4000mAh Power Bank-Instructional Manual. Available online: <https://avidopower.com/wp-content/uploads/2017/03/Avido-4000mAh-Power-Bank-Manual.pdf> (accessed on 29 February 2024).
45. Wireshark. Wireshark User's Guide. Available online: https://www.wireshark.org/docs/wsug_html_chunked/ChapterIntroduction.html (accessed on 8 January 2024).
46. Shebl, A.; Abriha, D.; Fahil, A.S.; El-Dokouny, H.A.; Elrasheed, A.A.; Csámer, Á. PRISMA hyperspectral data for lithological mapping in the Egyptian Eastern Desert: Evaluating the support vector machine, random forest, and XG boost machine learning algorithms. *J. Afr. Earth Sci.* **2021**, *178*, 104251. [CrossRef]
47. Tianqi, C.; Guestrin, C. XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794. Available online: <https://www.kdd.org/kdd2016/papers/files/rfp0697-chenAemb.pdf> (accessed on 8 December 2023).
48. Salerno, S. EloquentTinyML. An Eloquent Interface to Tensorflow Lite for Microcontrollers. Available online: <https://www.arduino.cc/reference/en/libraries/eloquenttinyml/> (accessed on 25 October 2023).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.



Article

Unmanned Aerial Vehicle-Based Techniques for Monitoring and Prevention of Invasive Apple Snails (*Pomacea canaliculata*) in Rice Paddy Fields

Senlin Guan ^{1,*}, Kimiyasu Takahashi ¹, Shunichiro Watanabe ² and Katsunori Tanaka ²

¹ Division of Crop Rotation Research for Lowland Farming, Kyushu-Okinawa Agricultural Research Center, National Agriculture and Food Research Organization, 496 Izumi, Fukuoka 833-0041, Chikugo, Japan

² Nileworks Co., Ltd. 2F, Kanda Square Front, 1-4-3 Kanda Nishikicho, Chiyoda, Tokyo 101-0054, Japan

* Correspondence: guan.s@affrc.go.jp; Tel.: +81-942-52-0692; Fax: +81-942-53-7776

Abstract: The destructive impact of invasive apple snail (*Pomacea canaliculata*) on young rice seedlings has garnered global attention, particularly in warm regions where rice production occurs. The preventative application of insecticide, particularly in areas with young rice seedlings and water depths exceeding 4 cm, has proven effective in mitigating this damage. In line with this recommendation, our study investigates the efficacy of site-specific drone-based insecticide applications to mitigate snail damage in rice paddies. These site-specific drone applications were strategically executed as directed by a highly accurate prescription map indicating the required insecticide quantity at specific locations. The prescription map was automatically generated through an advanced data processing program that used the aerial images acquired by a Real-Time Kinematic (RTK)-Unmanned Aerial Vehicle (UAV) as the input. Criteria were established to select the treatment locations; a value of below 4 cm from the top 95% percentile in the histogram of ground elevation data was used as a threshold to identify areas with a high-density of snail damage. The results demonstrated reductions in both the rates of rice damage and chemical usage following site-specific drone applications compared with the control fields. The findings in this study contribute to the advancement of effective site-specific pest control in precision agriculture.

Keywords: unmanned aerial vehicle; remote sensing; apple snail; *Pomacea canaliculata*; site-specific application; drone-based application

Citation: Guan, S.; Takahashi, K.; Watanabe, S.; Tanaka, K. Unmanned Aerial Vehicle-Based Techniques for Monitoring and Prevention of Invasive Apple Snails (*Pomacea canaliculata*) in Rice Paddy Fields. *Agriculture* **2024**, *14*, 299. <https://doi.org/10.3390/agriculture14020299>

Academic Editors: Gniewko Niedbala, Sebastian Kujawa, Magdalena Piekutowska and Tomasz Wojciechowski

Received: 15 January 2024

Revised: 8 February 2024

Accepted: 10 February 2024

Published: 13 February 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The invasive apple snail (*Pomacea canaliculata*), recognized as one of the world's 100 most invasive alien species in the International Union for Conservation of Nature's list [1]. It has extensively invaded agricultural and natural ecosystems in numerous moderate-temperature regions across the globe [2]. *P. canaliculata* exhibits a high degree of adaptability, thriving in both wetland and specific dryland habitats. Primarily, the snail feeds on a variety of bright green and succulent aquatic or semiaquatic plants, including duckweed, young rice (*Oryza sativa* L.) seedlings, lotus root, and other small plants that float on the surface of water. When *P. canaliculata* snails infest rice paddy fields with young seedlings, they feed voraciously on the rice plants, which results in significant crop losses if suitable alternative food sources are unavailable. Their extensive invasion into rice paddy fields, through irrigation canals and rivers, may not only devastate the crops but also out-compete native local snails [3,4], potentially altering the normal function of natural ecosystems [5–7].

Flooded rice paddy fields offer a habitat for *P. canaliculata* snails, allowing them to thrive and reproduce rapidly. The water depth in the flooded rice paddy field has a crucial influence on the movement and feeding activities of *P. canaliculata* snails. Research has shown that during the early growth stage of rice (specifically for young seedlings between 21 and 40 days after emergence), if they are submerged in water deeper than 5 cm,

the damage caused by *P. canaliculata* ranges from 46% to 100% [8]. Among the suggested strategies for preventing damage to young seedlings of both direct-seeded and transplanted rice, drainage or maintaining shallow water before the V5 growth stage [9,10], mechanical control through tillage and puddling [11,12], chemical control, crop rotation, and biological control [13] are considered effective.

Nonetheless, maintaining a shallow water level, for example, a level not exceeding 4 cm [14], is not straightforward because most paddy fields are not perfectly level. Water depth varies across the paddy field due to differences in ground elevation; some areas are higher and others are lower, and the elevation throughout the paddy field is generally uneven. Given that areas near water inlets and outlets tend to be deeper, local chemical treatments for these areas are often employed by farmers. However, this method often focuses on addressing limited areas, leaving untreated areas further from the field's access points. Comprehensive chemical treatments for the entire paddy field are also feasible, but are associated with high costs. Therefore, treating only the areas with deep water in the paddy field is a sensible approach.

Water depth data in a paddy field can be obtained by installing sensors at fixed points to measure the water level [15,16]. However, this method can only capture localized data points; it is incapable of providing comprehensive water depth data for the entire field. Obtaining accurate water depth distribution data becomes extremely challenging when the paddy field is flooded with seedlings or there is debris on the flat-water surface. As an alternative, measuring the ground height of the rice paddy when it is not flooded and estimating the water depth during irrigation is a more practical approach. There are currently many highly useful technologies that can be applied to measure bare ground elevation (also known as assessing ground or soil surface elevation). The assessment of ground surface elevation typically follows the criteria recommended for rice paddy fields, indicating an elevation difference within ± 5 cm or ± 3.5 cm from the mean horizontal plane [17] or an elevation standard deviation below 2 cm [18].

It is challenging to achieve ground surface elevation data with centimeter level precision; traditional methods involving measurements at multiple points, such as Total Stations, theodolites, or Global Navigation Satellite System (GNSS) receivers, may result in missing or incomplete data for unmeasured locations within the designated paddy field. The recent advanced measurement methods that can be adapted to assess ground surface elevation in paddy fields include Terrestrial Laser Scanning (TLS), Unmanned Aerial Vehicle (UAV)-based Light Detection and Ranging (LiDAR), and UAV-based photogrammetry [19,20]. Although TLS technology has proven effective in various applications, such as topographic mapping, architectural documentation, and cultural heritage preservation [21], it exhibits limitations when applied to relatively broad and flat rice paddies, including uneven data density arising from the lack of distinct topographical features, significant time and cost requirements, and the challenge of establishing suitable measurement points during the scanning process. UAV-based LiDAR is well-suited to assessing forest structure [22] or crop height [23–25]. However, when mapping relatively flat terrain, UAV-based photogrammetry is a more advanced choice owing to its cost-effectiveness, outstanding flexibility, and high-resolution. By utilizing high-resolution aerial imagery with accurate Real-Time Kinematic (RTK)-GNSS geo-positioning, UAV-based photogrammetry can achieve measurement accuracy within several centimeters by referencing Ground Control Points (GCPs) [26,27].

Owing to its ability to measure ground elevation differences to within a few centimeters, UAV-based photogrammetry can be applied to our approaches for controlling *P. canaliculata* snails in paddy fields. After the water depth has been estimated from the measured ground elevation data, site-specific application or variable rate (VR) pesticide application [28] can be implemented to save time and reduce costs. In precision agriculture, site-specific application, VR pesticide application, and VR fertilizer application are common practices that tailor input and usage to achieve desired outcomes based on timely diagnosis information on crop growth and the surrounding environment [29]. VR applications are typically performed with reference to the diagnostic information acquired from an on-board

sensor or a prescription map calculated from vegetation indices (VIs) such as the Normalized Difference Vegetation Index (NDVI) [30,31]. For example, based on UAV-derived Green NDVI (GNDVI), Yi et al. [32] used variable rate spraying for cotton defoliation applications. Another valuable indicator for representing ground elevation or crop height, the Digital Surface Model (DSM), has been applied in site-specific applications for orchards or terrains with significant height variations [33]. However, owing to limitations in the accuracy of elevations determined using UAV-based methods, no research has explored the use of DSM indicators to identify and select areas with subtle differences, such as of a few centimeters, for targeted treatments in rice paddy fields.

Building upon insights gained from advanced UAV-based techniques, our study takes a further step by proposing a comprehensive approach that leverages these techniques to enhance the monitoring and prevention of *P. canaliculata* in rice paddy fields. We focus on three key aspects: (1) developing a UAV-based method to assess the field levelness and estimate water depth, enabling the accurate identification of areas prone to snail infestation; (2) implementing site-specific pesticide application based on a prescription map, allowing precise targeting of snail populations while minimizing pesticide usage; and (3) employing UAV-derived imagery analysis to assess and quantify the extent of damage caused by *P. canaliculata* in rice paddy fields. These contributions aim to enhance the efficiency and effectiveness of pest management strategies and ultimately reduce the economic and ecological impacts of *P. canaliculata* in rice production.

2. Materials and Methods

2.1. Study Site

The study site is situated in the northwest of Kyushu, southern Japan, specifically in the Saga Plain. Over the past decade, the average annual temperature in this area is 17.4 degrees Celsius, as derived from historical weather data collected by an automated meteorological data acquisition system near the study site. Owing to the suitable climate and fertile soil, agricultural practices in this area often involve a double-cropping system, with wheat or barley cultivated in the winter and rice or soybeans in the summer. *P. canaliculata* thrives in the summer paddy fields; in the winter, the climate is conducive to a proportion of snails successfully overwintering in the paddy fields [34].

P. canaliculata (Figure 1a–d) becomes active following the transplantation of rice, typically in mid to late June in this double-cropping system period, causing damage to rice seedlings for approximately two weeks. During the machinery-based transplanting process, an insecticide known as Sukuminon, which contains 10.0% metaldehyde, is commonly applied to prevent the damage caused by *P. canaliculata*. This insecticide provides effective control for a period of approximately one week. However, following this period, the efficacy of the insecticide weakens and gradually diminishes; then, combined with historically abundant rainfall during this season, the snails swiftly invade rice paddies through agricultural waterways, channels, and rivers. This rapid invasion leads to destructive damage in rice fields where the protective effect of the insecticides is diminished, which particularly impacts young seedlings (Figure 1e).

Our experimental fields were selected within the high-damage zones caused by *P. canaliculata* snails at the study site, divided into three groups (G_1 , G_2 , and G_3), with each group containing two paired rice paddies, labeled as follows: F_1 and F_2 , F_3 and F_4 , F_5 , and F_6 (Figure 2). The two paddies in the same group were situated in close geographical proximity; the environmental conditions for rice growth and cultivation practices were identical. Traditional *P. canaliculata* management practices, which involve plowing, puddling, and transplanting with the simultaneous application of the insecticide Scuminon, were carried out in all experimental paddies. The paired paddies in each group were designated as controls: one underwent timely treatment by drone-based application; the other followed traditional management practices in which the initial insecticide was only applied during the transplanting process. Detailed information about the experimental fields is provided in Table 1.

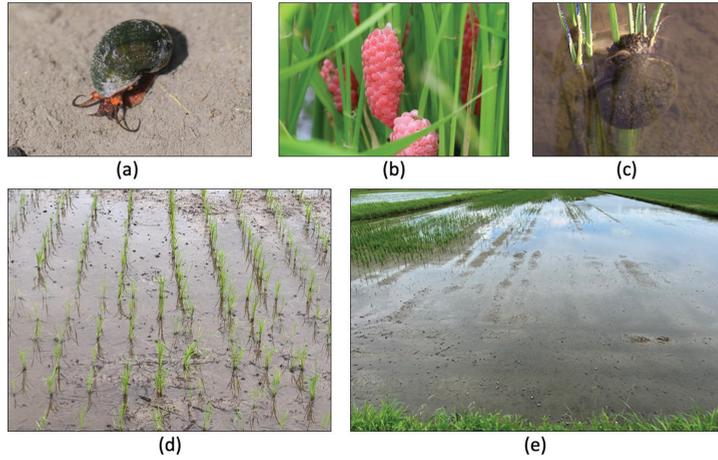


Figure 1. *P. canaliculata* snails. (a) Snails inhabiting the paddy field (shell height mostly ranging from 0.3 to 3 cm). (b) Egg masses—strong reproductive ability, with annual egg production ranging from 2000 to 8000. (c) Snail damage to rice plants. (d) Shallow water management at approximately 1 cm depth for paddy fields infested by the snails. (e) Significant seedling losses in a paddy field infested with the snails.

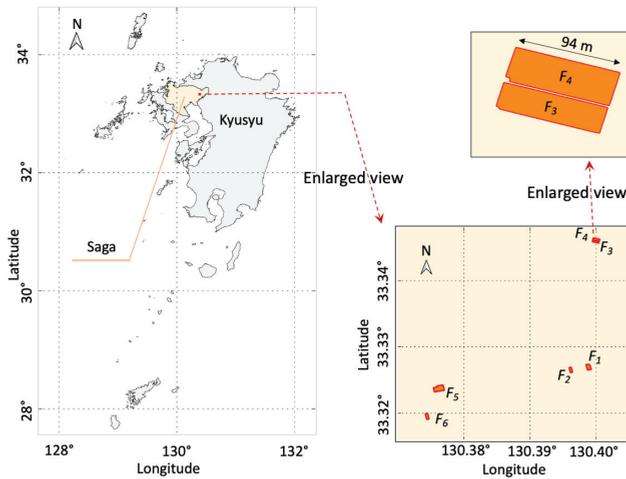


Figure 2. Map of the experimental fields.

Table 1. Information on the experimental fields.

Group Field	F_1	G_1	F_2	F_3	G_2	F_4	F_5	G_3	F_6
Area (ha)	0.49		0.26	0.26		0.33	1.18		0.33
Latitude,	33.326974 N	33.326546 N	33.345911 N	33.346190 N	33.323753 N	33.319522 N			33.319522 N
longitude	130.398870 E	130.396180 E	130.399962 E	130.400063 E	130.376322 E	130.374446 E			130.374446 E
Transplanting date	26 June 2023	26 June 2023	24 June 2023	24 June 2023	22 June 2023	23 June 2023			
Timely treatment	drone application			drone application			drone application		

2.2. UAVs and Agricultural Drones

To estimate the water depth in the experimental fields, we utilized an RTK-UAV, specifically the Phantom 4 RTK (DJI, China). This UAV is equipped with an on-board RGB camera capable of capturing high-resolution images at 4864×3648 pixels, ensuring a spatial resolution of 2.7 cm Ground Sampling Distance (GSD) when photographs are taken at an altitude of 100 m Above Ground Level (AGL). The UAV exhibits high positioning accuracy, with a vertical precision of 1.5 cm and horizontal accuracy of 1 cm specified by the manufacturer.

The UAV flight experiment, when the RTK system was in a fixed state, was conducted on 7 June 2023, subsequent to the plowing process that occurred after the harvest of wheat or barley and prior to the puddling. Flight parameters were configured using the DJI GS Pro application, with a flight altitude of 100 m AGL, 75% overlap in both forward and side directions, and a capture mode of 3-s intervals. The weather conditions were clear, with a wind speed below 1 m/s.

The selection of the flight altitude of 100 m AGL is primarily due to its capability to ensure accuracy within a few centimeters in both horizontal and vertical directions, particularly in the flat terrain of lowland rice production areas. According to a review [35], flying the UAV at a higher altitude can significantly improve vertical accuracy over flat terrain, a finding that is similarly corroborated by our previous field experiments [36]. With appropriate methods, it is even feasible to generate an orthomosaic without the need for GCPs [36]. Furthermore, flying at this altitude facilitates efficient surveying, making it suitable for practical large-scale paddy field measurements. Considering the typical row spacing of 30 cm and seedling spacing of 20–25 cm in conventional rice cultivation, the resulting GSD of 2.7 cm at this altitude is adequate for distinguishing individual rice plants, especially for discerning rice canopy from the original images and RGB orthomosaic.

Another flight experiment, executed with the same UAV and settings, was planned and performed on 23 August 2023, approximately 1 week before the rice heading stage. During the flight, the weather conditions were mostly cloudy, with an approximate wind speed of 1.5 m/s.

In our experimental fields dedicated to controlling *P. canaliculata* snails, we opted for Sukumin Bait 3, a granular eradicator containing 3.0% iron (II) phosphate hydrate, with no specific restrictions on application periods or frequency. The granule materials were dispersed from the air using two models of agricultural drones: Nileworks Nile-JZ (Tokyo, Japan) and DJI Agras T10 (Shenzhen, China). The Nile-JZ drone has a maximum takeoff weight of 27 kg, featuring six propellers, an internal load capacity of 8 kg for granule spreading, and achieves a spreading width of over 5 m. The Agras T10 drone, with a maximum takeoff weight of 24.8 kg and four propellers, is equipped with an internal load capacity of 10 kg for granule spreading and achieves a spreading range of 5 to 7 m.

The experimental conditions for aerial spreading are detailed in Table 2. During fixed RTK states, both the Nile-JZ and Agras T10 drones executed automatic spreading operations following predesigned flight routes, including automatic takeoff, spreading along the predetermined paths, and automatic landing. The flight altitudes were set at 1.5 m for Nile-JZ and 2 m for Agras T10, which corresponded to the height of the paddy field ridges and the height from the takeoff point, respectively. It should be noted that the implementation date for the drone-based insecticide application on F_5 preceded F_1 and F_3 by a week due to the earlier transplanting date in field F_5 .

Table 2. The experimental conditions for air spreading.

Drone Model		
	Nile-JZ	Agras T10
Target fields	F_1 and F_3	F_5
Application date	6 July 2023	29 June 2023
Weather	Sunny	Shower
Wind speed (m/s)	4.5	5
Wind direction	SSE	SSW
Temperature (°)	31.7	28.6

2.3. The Research Methodology

Figure 3 illustrates a block diagram of our research methodology; the right-hand column indicates the corresponding instruments and tools used. The research methodology comprises:

- Data acquisition, specifically for aerial images using the RTK-UAV.
- Automated Data Processing (ADP), developed with the Python language and PIX4D engine SDK 1.4 [37]. This includes procedures such as inputting aerial images, generating orthomosaics and the DSM, clipping the DSM of the field, calculating the areas for spreading, and creating the prescription map. The DJI Terra software, not in the procedures for ADP, was used for creating the prescription map for Agras T10.
- Drone flight plan, automatically generated by the Nile app and DJI Terra for their corresponding drone models.
- Drone application, carried out by Nile-JZ or Agras T10.
- Assessment of the damage rate caused by *P. canaliculata* snails, determined from the aerial images acquired by the RTK-UAV.

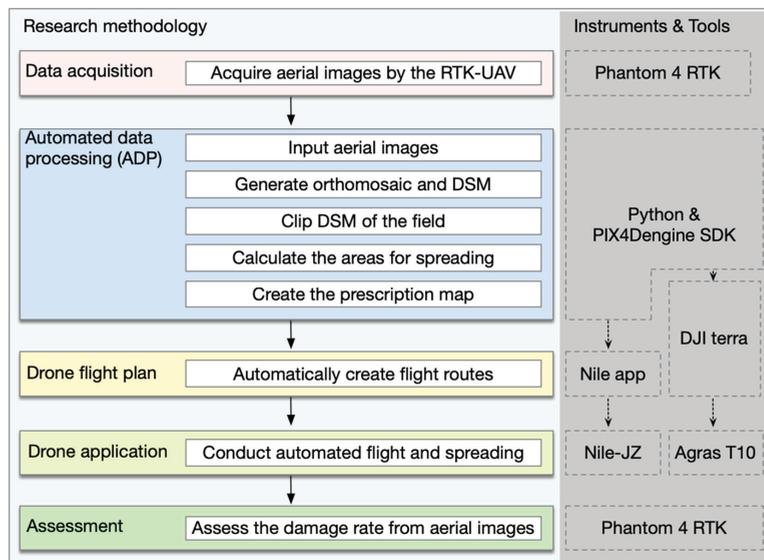


Figure 3. The research methodology, instruments, and tools.

ADP was developed using Python 3.8 on a Windows 10 Pro, 64-bit operating system. The development environment was a high-performance PC equipped with an Intel Xeon(R) Silver 4208 (2.1 GHz, 8 cores), 48 Gb of RAM, and a graphics accelerator with 3328 CUDA cores (NVIDIA RTX A2000, 12 Gb). In addition to PIX4Dengine SDK, the implementation utilized open-source tools such as GDAL/OGR 3.3.2 [38]. PIX4Dengine SDK offered a flexible interface for customizable configurations and processing controls, largely encompassing the capabilities found in PIX4D Mapper Pro, a Structure from Motion (SfM) [39] software developed by the same provider.

After completing aerial imagery acquisition using RTK-UAV at the study site, we obtained a set of images with precise geographical coordinates to serve as the input data for ADP. To better differentiate ground elevation within the same rice paddy, we established predefined geographical boundaries for each of our experimental fields, excluding boundary ridge areas with significant variations in ground elevation that differed from the base ground elevation of the rice paddy. Subsequently, we imported this dataset into ADP for automated processing.

The prescription map, provided in vector data formats such as GeoJSON or ESRI Shapefile by ADP, can be easily imported into Nile app to create optimized flight routes for Nile-JZ. However, DJI Terra, lacking the ability to generate a prescription map based on indices other than NDVI, requires manual referencing to the spreading area.

The assessment of the damage rate caused by *P. canaliculata* snails is based on the selection of geographical pixels according to the $\frac{R}{G}$ and $\frac{B}{G}$ ratios [40], along with the excess G index [41,42], where R , G , and B denote red, green, and blue digital values ranging from 0 to 255, respectively, in the RGB orthomosaic generated from the RTK-UAV-derived aerial imagery. The assessment was conducted using the following formulas:

$$C = \begin{cases} 1 & \text{if } \left(\frac{B}{G} < \theta_1\right) \text{ and } \left(\frac{R}{G} < \theta_2\right) \text{ and } (2 \times G - R - B > \theta_3) \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

$$r = \frac{\text{sum}(C)}{\text{count}(C)} \quad (2)$$

where r represents the canopy cover rate of rice areas, and the threshold values θ_1 , θ_2 , and θ_3 were determined using the following procedure:

-
- 01: **begin**
 - 02: Set the sample size n
 - 03: Extract a set of points P_n from within rice canopy regions
 - 04: Extract a set of points Q_n from outside rice canopy regions
 - 05: Calculate the values of $\frac{R}{G}$, $\frac{B}{G}$ and $2 \times G - R - B$ for each point in P_n and Q_n using a GIS program or tool
 - 06: Assign the 97.5th percentile value of $\frac{R}{G}$ within P_n to θ_1
 - 07: Assign the 97.5th percentile value of $\frac{B}{G}$ within P_n to θ_2
 - 08: Assign the 2.5th percentile value of $2 \times G - R - B$ within P_n to θ_{3a}
 - 09: Filter a subset Q' of points from Q_n with the filter $\frac{R}{G} < \theta_1$ and $\frac{B}{G} < \theta_2$
 - 10: Calculate the 2.5th percentile value θ_{3b} for $2 \times G - R - B$ within Q'
 - 11: Choose the value for θ_3 as the maximum of θ_{3a} and θ_{3b}
 - 12: **end**
-

In our study, with a sample size of 34, the values of θ_1 , θ_2 , and θ_3 were determined as 0.82, 0.82, and 60, respectively. These values are subject to variation based on changes in the sample size and sampling points, particularly at critical points between the rice canopy and water surface. To mitigate noise associated with these critical points, values for θ_3 were selected at the 97.5th and 2.5th percentiles. The geographical raster file of the rice canopy was calculated using Equation (1) and validated through a comparison with the RGB orthomosaic.

The values of θ_1 , θ_2 , and θ_3 significantly deviate from the suggested default values of 0.95, 0.95, and 20 in [43]. This variance may be attributed to specific conditions observed in flooded rice paddies and the presence of floating residues. Damage to rice seedlings by *P. canaliculata* snails results in the loss or partial damage of rice seedlings. In the current mechanized rice cultivation system, replanting will not be conducted for the missing parts, leaving these areas blank until harvest. These areas are easily distinguishable from the aerial images at a flight altitude of 100 m AGL. Partially weak or damaged rice seedlings gradually recover and can reach the maximum canopy area around the rice heading stage. During this stage, aerial images are highly effective in distinguishing rice canopy, where non-rice canopy areas consist of either water surfaces or distinctively colored floating residues. Since we can clearly discern rice canopy from aerial images, we opted to substitute the traditional field sampling method for calculating damage rates. Traditional field sampling also faces challenges, such as determining sampling protocols and accurately aligning predetermined sampling points with positions on the aerial RGB orthomosaic. Moreover, due to discrepancies in evaluation criteria, actual field sampling surveys may not necessarily yield accurate actual damage rates.

3. Results

3.1. Implementation of Site-Specific Drone Application

Figure 4a illustrates the visualized ground elevation map generated after completing the procedures of generating orthomosaics and the DSM, along with clipping DSM of the field F_1 . This map, which features color grading and 5 cm contours, provides an intuitive representation of the distribution of height differences and offers valuable insights for decision making in insecticide application.

ADP automatically analyzed the histogram of the elevation distribution data. Considering the possibility of increased water levels due to heavy rain, the 95% percentile of these data was considered to be the water level at 0, and values below the threshold (4 cm below the water level [14]) were identified as the points for insecticide application (Figure 4b). Based on this threshold value, areas requiring insecticide application were marked and transformed into a 1 m grid cell map, which is referred to as the prescription map (Figure 4c). The prescription map with a 1 m grid cell is advantageous for creating a flight route with high covering accuracy, particularly concerning the designated area, for agricultural drones. The flight routes for agricultural drones, with maximum effective working width, often require adjustments in flight direction based on on-site factors such as traffic, obstacles, and wind direction. However, the flight direction of the on-site generated grid cell, for example, 5 m \times 5 m, is likely not aligned with the 1 m grid cell direction, leading to errors in the designated spray quantity due to the intersection of grid cells of two different sizes. This error tends to decrease as the grid cell size becomes smaller.

Moreover, the prescription map with a 1 m grid cell broadens the applicability of the data. It can be utilized not only with the Nile-JZ used in this experiment but also with other agricultural drones. Additionally, it is compatible with various agricultural machinery, including boom sprayers, fertilizer applicators, and agricultural helicopters, that adhere to the ISO 11783-10 data interface. This compatibility arises from the fact that the 1 m grid cell has vertical and horizontal directions designated as north and east, aligning with the same orientation as the grid cell defined in ISO 11783-10.

Utilizing the prescription map, Nile-JZ autonomously performed the insecticide application, along with optimized flight paths parallel to the long side of the field, aiming to minimize untreated areas, aligning with the designated areas in the 1 m grid cell on the prescription map. The actual spreading paths (Figure 4d) were plotted from the aircraft log data. It is evident that the aircraft's spreading covered the vast majority of the designated spreading areas. Because the aircraft has a spreading width limitation of 5 m, some of the scattered and small areas indicated for spreading were not treated.

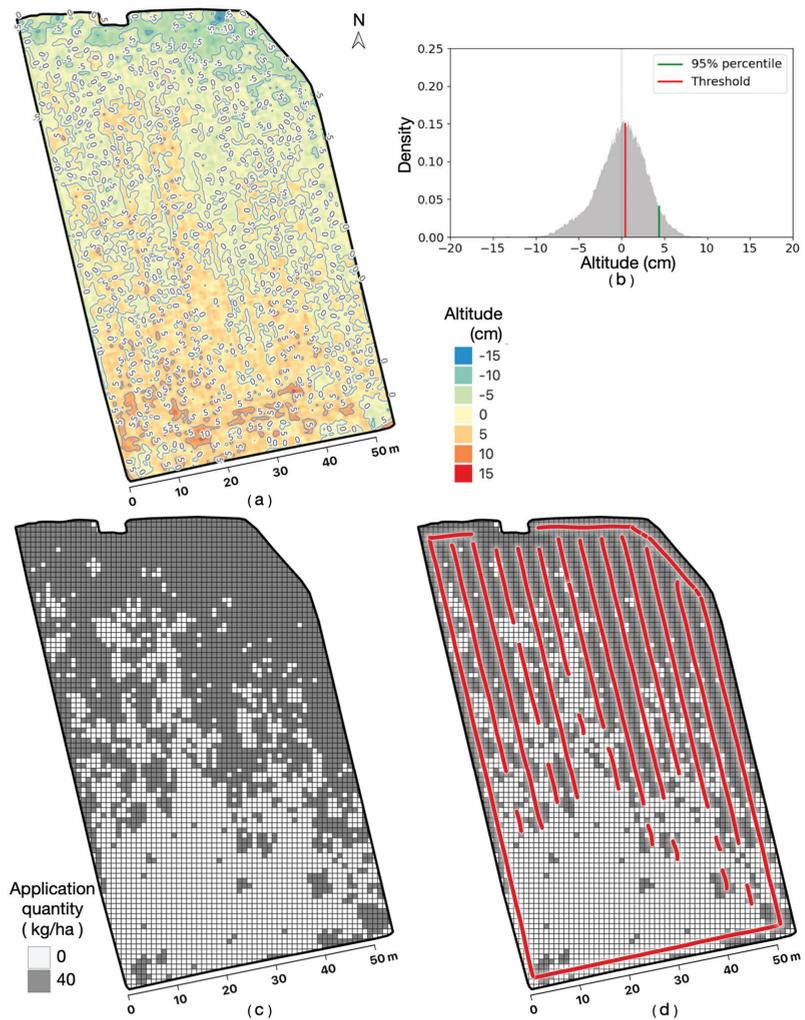


Figure 4. Site-specific drone application based on the prescription map for F_1 . (a) Visualized ground elevation map with color grading and 5 cm contours. (b) Histogram of ground elevation data with a red-line threshold and a green-line 95% percentile. (c) Prescription map with 1 m grid cell for application quantity. (d) Flight paths indicated by red lines for drone application using Nile-JZ.

The same procedures were applied to F_3 . In Figure 5a, the region with higher ground elevation (presented in deep red in the upper-left part) remained unsubmerged during the on-site inspection on the fourth day after transplantation (28 June 2023), whereas the water depth exceeded 15 cm in the lower-right corner. The spreading paths of Nile-JZ (Figure 5c) confirmed that no spreading operations were performed in the areas remaining unsubmerged areas, aligning with expectations.

Owing to various factors, including weather conditions, the optimal spreading implementation time window, and the scheduling of the experimental agenda, a different spreading method was applied for F_5 . Agricultural land identification and spreading area settings were manually specified on DJI Terra. Then, the Agras T10 executed continuous and uniform spreading in this area with 40 kg/ha settings (Figure 6b). The red lines of the spreading paths in Figure 6b were manually drawn based on a screenshot from the DJI Agras App, as flight log data were not accessible.

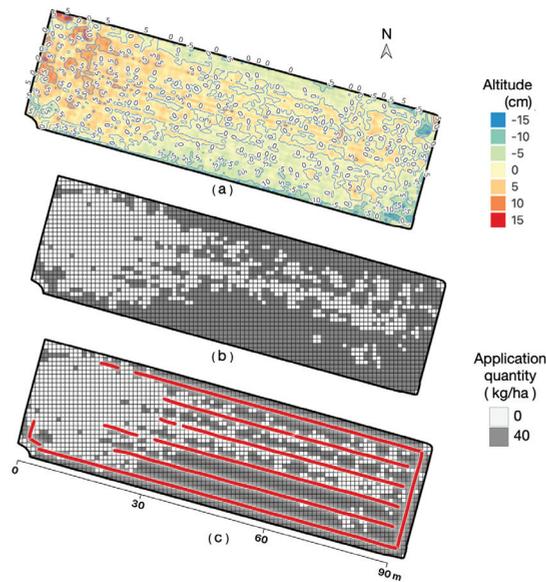


Figure 5. Site-specific drone application based on the prescription map for F_3 . (a) Visualized ground elevation map with color grading and 5 cm contours. (b) Prescription map with 1 m grid c for application quantity. (c) Flight paths indicated by red lines for drone application using Nile-JZ.

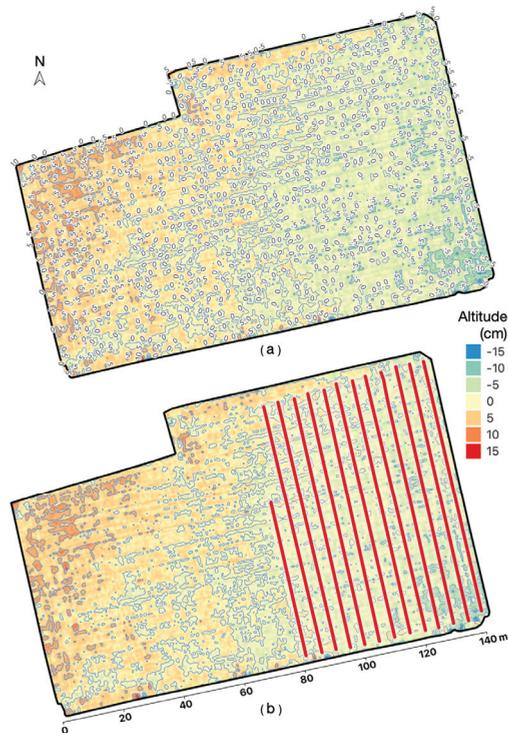


Figure 6. Site-specific drone application based on the prescription map for F_5 . (a) Visualized ground elevation map with color grading and 5 cm contours. (b) Flight paths indicated by red lines for drone application using Agras T10.

3.2. Effectiveness Analysis of Site-Specific Drone Application

The effectiveness of site-specific drone-based insecticide application was assessed approximately 1 week before the rice heading stage. This timeframe was strategically chosen as it provided the optimal evaluation window. At this point, rice plants that had suffered destructive damage were unlikely to recover, and the well-developed branches and leaves would effectively cover normal row and plant spacing. The evaluation process first involved aerial photography and the generation of an RGB orthomosaic. Subsequently, the canopy cover rate was calculated using Equation (1). These results are visualized in Figure 7. In this figure, green represents the canopy cover of rice and white represents areas other than rice, such as water surfaces. It is evident that the canopy cover area on the left side, corresponding to F_1 with drone application, is significantly more than the canopy cover area on the right side, corresponding to F_2 without drone application. This observation implies that the damage rate in F_1 , which was subject to drone application, is noticeably lower than in the untreated F_2 .

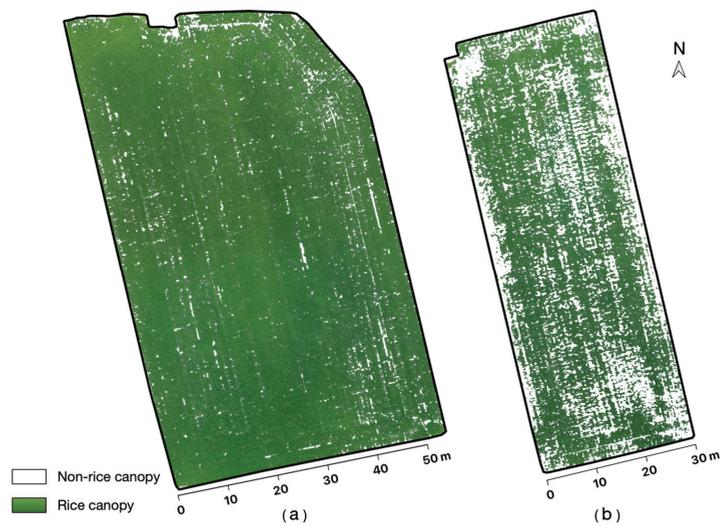


Figure 7. Rice green canopy for estimating damage rates. (a) Field F_1 with drone application; (b) Field F_2 without drone application.

A comparison of the damage rates in the three control experimental fields is presented in Figure 8a. When comparing damage rates in fields with prescription map-derived drone application (F_1, F_3) to those without drone application (F_2, F_4), the respective rates were 4.0%, 10.8% versus 33.2%, 23.6%. The field area-weighted arithmetic means for the two fields with prescription map-derived drone application and those without were 6.4% and 27.8%, respectively, indicating a difference of 21.4%. For field F_5 with non-prescription map-derived drone application and F_6 without drone application, the respective damage rates were 4.6% and 12.7%, with a difference of 8.1%.

By simultaneously reducing the damage rates, the quantity of insecticide applied through drone application is significantly lower compared with the total field application quantity by traditional uniform application (Figure 8b). For example, in F_1, F_3 , and F_5 , if traditional uniform application was employed at a rate of 40 kg/ha, the required quantity of chemicals would be 19.6 kg, 10.5 kg, and 47.3 kg, respectively. However, the quantities required for prescription map-derived drone application were 14.0 kg and 7.7 kg for F_1 and F_3 , equivalent to reductions of 28.4%, 26.5%, respectively. Similarly, the quantities required for non-prescription map-derived drone application were 19.2 kg, equivalent to reductions of 59.4%.

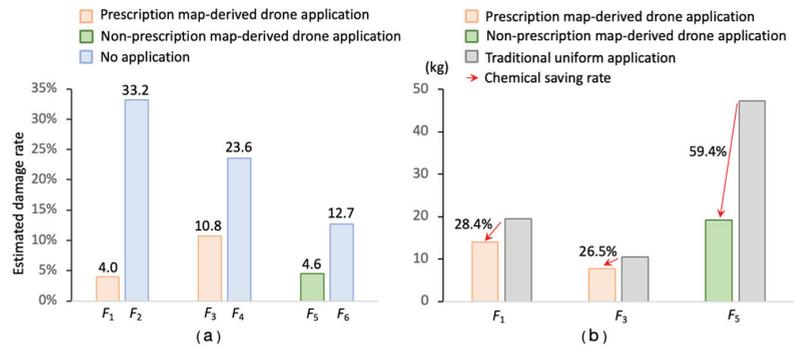


Figure 8. Rates of damage reduction and chemical savings. (a) Comparison of estimated damage rates at each field group between drone application and no application; (b) Chemical saving rates between total field application quantity by traditional uniform application and drone application.

If we neglect the method of implementing a prescription map in drone application, the field area-weighted arithmetic means for the three fields with drone application and those without application were 5.3% and 22.5%, respectively, which is a difference of 17.2%. This reduction in damage rate is expected to translate into an increase in final yield. Specifically, the yield in rice fields with drone application is anticipated to be 17.2% higher than in fields without application.

4. Discussion

This study utilized UAV-based sensing technologies to identify areas that were highly vulnerable to damage by *P. canaliculata* snails. Subsequently, autonomous agricultural drones were deployed to perform precise and efficient site-specific applications in these identified regions. In comparison with the traditional control approaches, this method not only mitigated damage rates but also reduced chemical usage. The accuracy in identifying areas at high risk of damage is pivotal for the success of the prescription map for drone-based applications. The precision of UAV-based DSM, portraying the ground elevation surface of rice paddies, is thus indispensable for the generation of prescription maps. In this study, we obtained highly precise DSM using the methods outlined in [29], which encompassed (1) conducting UAV flights at an altitude of 100 m AGL or higher during stable weather conditions and (2) achieving high-quality camera calibration during the orthomosaic generation process. If the quality in the orthomosaic generation report was suboptimal, calibration with GCPs is recommended.

It is essential to note that the time window available for drone sensing in double-cropping systems is not large. For transplanting rice, it is performed following the plowing process after the harvest of winter crops (such as wheat or barley) and before puddling. In the case of direct-seeded rice, it is advisable to conduct the operation before or around the germination period, preferably before the V3 growth stage to mitigate noise caused by water surfaces or young rice plants. Equally important is the correct weather conditions during UAV sensing; non-rainy days with ground wind speeds below 5 m/s are favored.

The generation of the prescription map is automated by ADP due to the utilization of PIX4Dengine SDK during orthomosaic and DSM creation. In addition to its capabilities for automatic processing, PIX4Dengine SDK can be extended to accommodate applications, including *P. canaliculata* control. Alternatively, standalone SfM software can also be used, but after obtaining the orthomosaic and DSM, manual tasks are required, such as clipping the DSM of the field, calculating thresholds for spreading areas, and conversion into a 1 m grid cell map.

As mentioned in [44], there is still a paucity of research on agriculture drone-based site-specific or variable rate applications, which is primarily attributable to the limited availability of drone models capable of such precise application. Furthermore, prescription

maps, while frequently grounded on NDVI indices or yield maps, might instead encompass more useful indices, such as other VIs or DSM. The selection of agricultural drones that support prescription maps created with DSM, such as Nile-JZ, or prescription maps in a universal format, is a prerequisite for the successful implementation of the technology in our study.

In the study, we did not conduct an actual yield survey because, in areas where rice is completely missing with no rice plants, the yield is naturally zero. In areas where rice is not missing, the number of rice plants does not necessarily equate to the damage rate, as yield can be influenced by individual differences in rice plants. Instead, we calculated potential yield losses based on the damage rate. Furthermore, we did not assess the final profits derived from our research methodology. Profit calculations would need to consider various other influencing factors, such as material and equipment costs, software expenses, development costs, and labor costs, which were beyond the scope of our study. Opting only for simple profit data based on increased yield and cost savings in pesticides may be feasible but lacks representativeness and persuasiveness, especially in the absence of exact yield data.

Currently, we are in the process of developing another study that comprehensively evaluates the practical operational benefits of using our research methodology, taking into account a wide range of factors. Additionally, our ongoing future works include: (1) attempting to expand the time window available for UAV-based sensing in double-cropping systems to alleviate the compression of intensive operations during rice transplantation; and (3) developing a public data interface for prescription maps in agricultural drones and applying it to other pest control or variable rate application scenarios.

5. Conclusions

Our study has established that site-specific drone-based insecticide application is a viable and efficient strategy for mitigating damage to rice paddies by *P. canaliculata*. The high precision of the prescription map, derived from RTK-UAV-based aerial imagery, provides accurate site-specific application guidance to facilitate efficient treatment precisely where needed. The automated data processing involved in generating the prescription map streamlines the postprocessing of raw aerial images, improving processing efficiency and reducing technological barriers for general farmers. The implementation of site-specific applications by two autonomous agricultural drones was able to achieve noticeable reductions in damage rates and chemical usage, showing the potential to increase yield and improve the sustainability of agricultural practices. These findings, which contribute to the advancement of precision agriculture, underscore the importance of considering topographical variations, including subtle changes at the centimeter level, in pest control strategies.

Author Contributions: S.G., conceptualization, methodology, investigation, software, validation, and writing—original draft preparation, review, and editing; K.T. (Kimiyasu Takahashi), conceptualization, methodology, project administration, investigation, supervision, funding acquisition, and writing—review. S.W., software, investigation, resources, and writing—review; K.T. (Katsunori Tanaka), project administration, investigation, resources, funding acquisition, and writing—review. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partly supported by “Research project for technologies to strengthen the international competitiveness of Japan’s agriculture and food industry” Shin1do1 (21452858) and the Research Program on Development of Innovative Technology grants (JPJ007097) from the Bio-oriented Technology Research Advancement Institution (BRAIN).

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: Sincere thanks are extended to Akitoshi Honbu, Makoto Nakajima, and Atsuya Yokota for their contributions to data acquisition in both the field survey experiment and UAV photogrammetry. Gratitude is also extended to Agribase Niiyama Ltd. for generously providing access to experimental fields and necessary resources. The authors would like to express special thanks to Masumi Kabashima for her diligent efforts in programming and assisting with data processing for this research.

Conflicts of Interest: Author Shunichiro Watanabe and Katsunori Tanaka were employed by the company Nileworks Co., Ltd. 2F. The authors declare that this study received fundings from “Research project for technologies to strengthen the international competitiveness of Japan’s agriculture and food industry” Shin1d01 (21452858) and the Research Program on Development of Innovative Technology grants (JPJ007097) from the Bio-oriented Technology Research Advancement Institution (BRAIN). The funders were not involved in the study design, collection, analysis, interpretation of data, the writing of this article or the decision to submit it for publication.

Abbreviations

The following abbreviations are used in this manuscript:

ADP	Automated Data Processing
AGL	Above Ground Level
DSM	Digital Surface Model
GCP	Ground Control Point
GNSS	Global Navigation Satellite System
GSD	Ground Sample Distance
GNDVI	Green Normalized Difference Vegetation Index
LiDAR	Light Detection and Ranging
NDVI	Normalized Difference Vegetation Index
RTK	Real-Time Kinematic
TLS	Terrestrial Laser Scanning
UAV	Unmanned Aerial Vehicle
VR	Variable Rate

References

1. Lowe, S.; Browne, M.; Boudjelas, S.; De Poorter, M. *100 of the World’s Worst Invasive Alien Species: A Selection from the Global Invasive Species Database*; Invasive Species Specialist Group: Auckland, New Zealand, 2000; Volume 12.
2. CABI Digital Library. *Pomacea canaliculata* (Invasive Apple Snail). 2013. Available online: <https://www.cabidigitallibrary.org/doi/10.1079/cabicompndium.68490> (accessed on 21 June 2023).
3. Halwart, M. The golden apple snail *Pomacea canaliculata* in Asian rice farming systems: Present impact and future threat. *Int. J. Pest Manag.* **1994**, *40*, 199–206. [CrossRef]
4. Chaichana, R.; Sumpan, T. The potential ecological impact of the exotic snail *Pomacea canaliculata* on the Thai native snail *Pila scutata*. *Sci. Asia* **2014**, *40*, 11–15. [CrossRef]
5. Carlsson, N.; Kestrup, Å.; Mårtensson, M.; Nyström, P. Lethal and non-lethal effects of multiple indigenous predators on the invasive golden apple snail (*Pomacea canaliculata*). *Freshw. Biol.* **2004**, *49*, 1269–1279. [CrossRef]
6. Carlsson, N.O.; Brönmark, C.; Hansson, L.A. Invading herbivory: The golden apple snail alters ecosystem functioning in Asian wetlands. *Ecology* **2004**, *85*, 1575–1580. [CrossRef]
7. Joshi, R. Problems with the management of the golden apple snail *Pomacea canaliculata*: An important exotic pest of rice in Asia. In *Area-Wide Control of Insect Pests: From Research to Field Implementation*; Springer: Dordrecht, The Netherlands, 2007; pp. 257–264.
8. Sin, T.S. Damage potential of the golden apple snail *Pomacea canaliculata* (Lamarck) in irrigated rice and its control by cultural approaches. *Int. J. Pest Manag.* **2003**, *49*, 49–55. [CrossRef]
9. Wada, T.; Ichinose, K.; Higuchi, H. Effect of drainage on damage to direct-sown rice by the apple snail *Pomacea canaliculata* (Lamarck)(Gastropoda: Ampullariidae). *Appl. Entomol. Zool.* **1999**, *34*, 365–370. [CrossRef]
10. Counce, P.A.; Keisling, T.C.; Mitchell, A.J. A uniform, objective, and adaptive system for expressing rice development. *Crop Sci.* **2000**, *40*, 436–443. [CrossRef]
11. Takahashi, K.; Seki, M.; Nishida, H. Prevention of the Harm from Apple Snail with Rotary Cultivator. *J. Jpn. Soc. Agric. Mach.* **2002**, *64*, 101–107. [CrossRef]
12. Takahashi, K.; Seki, M.; Higashi, S.; Miike, T. Development of Technology by Rotary Tiller to Control the Density of Apple Snail. *J. Jpn. Soc. Agric. Mach.* **2005**, *67*, 68–74. [CrossRef]
13. Wada, T. Strategies for Controlling the Apple Snail *Pomacea canaliculata* (Lamarck) (Gastropoda: Ampullariidae) in Japanese Direct-Sown Paddy Fields. *Jpn. Agric. Res. Q. JARQ* **2004**, *38*, 75–80. [CrossRef]

14. Makiyama, M.; Itoh, T. The Effects of Shallow Ponding Depth for Control of the Damage by *Pomacea canaliculata*. *J. Agric. Eng. Soc. Jpn.* **2005**, *73*, 793–796. [CrossRef]
15. Kawakami, Y.; Furuta, T.; Nakagawa, H.; Kitamura, T.; Kurosawa, K.; Kogami, K.; Tajino, N.; Tanaka, M.S. Rice cultivation support system equipped with water-level sensor system. *IFAC-PapersOnLine* **2016**, *49*, 143–148. [CrossRef]
16. Cruz, K.M.S.D.; Ella, V.B.; Suministrado, D.C.; Pereira, G.S.; Agulto, E.S. A Low-Cost Wireless Sensor for Real-Time Monitoring of Water Level in Lowland Rice Field under Alternate Wetting and Drying Irrigation. *Water* **2022**, *14*, 4128. [CrossRef]
17. Osari, H. A method for assessing land leveling to produce high-quality consolidated paddy fields. *Paddy Water Environ.* **2003**, *1*, 35–41. [CrossRef]
18. Yamaji, E. Standard and evaluation of paddy field consolidation. In Proceedings of the Soil and Water Engineering for Paddy Field Management, International Workshop, Bangkok, Thailand, 28–30 January 1992.
19. Mora, O.E.; Suleiman, A.; Chen, J.; Pluta, D.; Okubo, M.H.; Josenhans, R. Comparing sUAS photogrammetrically-derived point clouds with GNSS measurements and terrestrial laser scanning for topographic mapping. *Drones* **2019**, *3*, 64. [CrossRef]
20. Rivera, G.; Porras, R.; Florencia, R.; Sánchez-Solís, J.P. LiDAR applications in precision agriculture for cultivating crops: A review of recent advances. *Comput. Electron. Agric.* **2023**, *207*, 107737. [CrossRef]
21. Ulvi, A. Documentation, Three-Dimensional (3D) Modelling and visualization of cultural heritage by using Unmanned Aerial Vehicle (UAV) photogrammetry and terrestrial laser scanners. *Int. J. Remote Sens.* **2021**, *42*, 1994–2021. [CrossRef]
22. Wallace, L.; Lucier, A.; Malenovský, Z.; Turner, D.; Vopěnka, P. Assessment of forest structure using two UAV techniques: A comparison of airborne laser scanning and structure from motion (SfM) point clouds. *Forests* **2016**, *7*, 62. [CrossRef]
23. ten Harkel, J.; Bartholomeus, H.; Kooistra, L. Biomass and crop height estimation of different crops using UAV-based LiDAR. *Remote Sens.* **2019**, *12*, 17. [CrossRef]
24. Sofonia, J.; Shendryk, Y.; Phinn, S.; Roelfsema, C.; Kendoul, F.; Skocaj, D. Monitoring sugarcane growth response to varying nitrogen application rates: A comparison of UAV SLAM LiDAR and photogrammetry. *Int. J. Appl. Earth Obs.* **2019**, *82*, 101878. [CrossRef]
25. Maimaitijiang, M.; Sagan, V.; Erkbol, H.; Adrian, J.; Newcomb, M.; LeBauer, D.; Pauli, D.; Shakoor, N.; Mockler, T. UAV-based sorghum growth monitoring: A comparative analysis of lidar and photogrammetry. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2020**, *3*, 489–496. [CrossRef]
26. Urban, R.; Reindl, T.; Brouček, J. Testing of drone DJI Phantom 4 RTK accuracy. In *Advances and Trends in Geodesy, Cartography and Geoinformatics II*; CRC Press: Boca Raton, FL, USA, 2020; pp. 99–105.
27. Ekaso, D.; Nex, F.; Kerle, N. Accuracy assessment of real-time kinematics (RTK) measurements on unmanned aerial vehicles (UAV) for direct geo-referencing. *Geo Spat. Inf. Sci.* **2020**, *23*, 165–181. [CrossRef]
28. Swinton, S.; Lowenberg-DeBoer, J. Evaluating the profitability of site-specific farming. *J. Prod. Agric.* **1998**, *11*, 439–446. [CrossRef]
29. Bongiovanni, R.; Lowenberg-DeBoer, J. Precision agriculture and sustainability. *Precis. Agric.* **2004**, *5*, 359–387. [CrossRef]
30. Zhang, J.; Wang, W.; Krienke, B.; Cao, Q.; Zhu, Y.; Cao, W.; Liu, X. In-season variable rate nitrogen recommendation for wheat precision production supported by fixed-wing UAV imagery. *Precis. Agric.* **2022**, *23*, 830–853. [CrossRef]
31. Pawase, P.P.; Nalawade, S.M.; Bhanage, G.B.; Walunj, A.A.; Kadam, P.B.; Durgude, A.G.; Patil, M.G. Variable rate fertilizer application technology for nutrient management: A review. *Int. J. Agric. Biol. Eng.* **2023**, *16*, 11–19. [CrossRef]
32. Yi, L.; Lan, Y.; Kong, H.; Kong, F.; Huang, H.; Han, X. Exploring the potential of UAV imagery for variable rate spraying in cotton defoliation application. *Int. J. Precis. Agric. Aviat.* **2019**, *2*, 42–45. [CrossRef]
33. Wang, C.; Liu, Y.; Zhang, Z.; Han, L.; Li, Y.; Zhang, H.; Wongsuk, S.; Li, Y.; Wu, X.; He, X. Spray performance evaluation of a six-rotor unmanned aerial vehicle sprayer for pesticide application using an orchard operation mode in apple orchards. *Pest Manag. Sci.* **2022**, *78*, 2449–2466. [CrossRef]
34. Yoshida, K.; Wada, T.; Matsukura, K.; Shiba, T. Potential overwintering areas of the alien apple snail, *Pomacea canaliculata*. *Aquat. Invasions* **2022**, *17*, 402–414. [CrossRef]
35. Jiménez-Jiménez, S.I.; Ojeda-Bustamante, W.; Marcial-Pablo, M.D.J.; Enciso, J. Digital terrain models generated with low-cost UAV photogrammetry: Methodology and accuracy. *ISPRS Int. J. Geo-Inf.* **2021**, *10*, 285. [CrossRef]
36. Guan, S.; Takahashi, K.; Nakano, K.; Fukami, K.; Cho, W. Real-Time Kinematic Imagery-Based Automated Levelness Assessment System for Land Leveling. *Agriculture* **2023**, *13*, 657. [CrossRef]
37. PIX4D. *Pix4Dengine SDK Quick-Start*; PIX4D S.A.: Prilly, Switzerland, 2019.
38. GDAL/OGR Contributors. *GDAL/OGR Geospatial Data Abstraction Software Library*; Open Source Geospatial Foundation: Beaverton, OR, USA, 2021.
39. Westoby, M.J.; Brasington, J.; Glasser, N.F.; Hambrey, M.J.; Reynolds, J.M. ‘Structure-from-Motion’ photogrammetry: A low-cost, effective tool for geoscience applications. *Geomorphology* **2012**, *179*, 300–314. [CrossRef]
40. Paruelo, J.M.; Lauenroth, W.K.; Roset, P.A. Estimating aboveground plant biomass using a photographic technique. *Rangel. Ecol. Manag./J. Range Manag. Arch.* **2000**, *53*, 190–193. [CrossRef]
41. Meyer, G.E.; Neto, J.C. Verification of color vegetation indices for automated crop imaging applications. *Comput. Electron. Agric.* **2008**, *63*, 282–293. [CrossRef]
42. Woebbecke, D.M.; Meyer, G.E.; Von Bargen, K.; Mortensen, D.A. Color indices for weed identification under various soil, residue, and lighting conditions. *Trans. Asae* **1995**, *38*, 259–269. [CrossRef]

43. Patrignani, A.; Ochsner, T.E. Canopeo: A powerful new tool for measuring fractional green canopy cover. *Agron. J.* **2015**, *107*, 2312–2320. [CrossRef]
44. Song, C.; Zhou, Z.; Zang, Y.; Zhao, L.; Yang, W.; Luo, X.; Jiang, R.; Ming, R.; Zang, Y.; Zi, L.; et al. Variable-rate control system for UAV-based granular fertilizer spreader. *Comput. Electron. Agric.* **2021**, *180*, 105832. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.



Article

Wheat Teacher: A One-Stage Anchor-Based Semi-Supervised Wheat Head Detector Utilizing Pseudo-Labeling and Consistency Regularization Methods

Rui Zhang, Mingwei Yao, Zijie Qiu, Lizhuo Zhang, Wei Li and Yue Shen *

College of Information and Intelligence, Hunan Agricultural University, Changsha 410125, China;

* Correspondence: shenyue@hunau.edu.cn

Abstract: Wheat breeding heavily relies on the observation of various traits during the wheat growth process. Among all traits, wheat head density stands out as a particularly crucial characteristic. Despite the realization of high-throughput phenotypic data collection for wheat, the development of efficient and robust models for extracting traits from raw data remains a significant challenge. Numerous fully supervised target detection algorithms have been employed to address the wheat head detection problem. However, constrained by the exorbitant cost of dataset creation, especially the manual annotation cost, fully supervised target detection algorithms struggle to unleash their full potential. Semi-supervised training methods can leverage unlabeled data to enhance model performance, addressing the issue of insufficient labeled data. This paper introduces a one-stage anchor-based semi-supervised wheat head detector, named “Wheat Teacher”, which combines two semi-supervised methods, pseudo-labeling, and consistency regularization. Furthermore, two novel dynamic threshold components, Pseudo-label Dynamic Allocator and Loss Dynamic Threshold, are designed specifically for wheat head detection scenarios to allocate pseudo-labels and filter losses. We conducted detailed experiments on the largest wheat head public dataset, GWHD2021. Compared with various types of detectors, Wheat Teacher achieved a mAP0.5 of 92.8% with only 20% labeled data. This result surpassed the test outcomes of two fully supervised object detection models trained with 100% labeled data, and the difference with the other two fully supervised models trained with 100% labeled data was within 1%. Moreover, Wheat Teacher exhibits improvements of 2.1%, 3.6%, 5.1%, 37.7%, and 25.8% in mAP0.5 under different labeled data usage ratios of 20%, 10%, 5%, 2%, and 1%, respectively, validating the effectiveness of our semi-supervised approach. These experiments demonstrate the significant potential of Wheat Teacher in wheat head detection.

Keywords: digital agriculture; deep learning; semi-supervised object detection; wheat head detection

Citation: Zhang, R.; Yao, M.; Qiu, Z.; Zhang, L.; Li, W.; Shen, Y. Wheat Teacher: A One-Stage Anchor-Based Semi-Supervised Wheat Head Detector Utilizing Pseudo-Labeling and Consistency Regularization Methods. *Agriculture* **2024**, *14*, 327. <https://doi.org/10.3390/agriculture14020327>

Academic Editors: Gniewko Niedbala, Sebastian Kujawa, Tomasz Wojciechowski and Magdalena Piekutowska

Received: 11 January 2024

Revised: 13 February 2024

Accepted: 17 February 2024

Published: 19 February 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Wheat has consistently been one of the primary crops for human sustenance. Scientists in wheat breeding have persistently strived to develop new wheat varieties with higher yields. Traditional breeding methods still heavily rely on manual observation, and innovative gains in genetic improvement may arise from genomic selection, novel high-throughput phenotyping technologies, or a combination of both [1–4]. These techniques are crucial for selecting important wheat traits related to yield potential, disease resistance, or adaptation to abiotic stress. Despite the realization of high-throughput phenotypic data collection, the development of efficient and robust models for extracting traits from raw data remains a significant challenge. Among all traits, wheat head density (the number of wheat heads per unit ground area) is a major yield component and is still manually evaluated in breeding trials. This manual evaluation is labor-intensive and introduces measurement errors of approximately 10% [5–7]. Therefore, there is a need to develop image-based methods to enhance the throughput and accuracy of wheat head counting in field conditions. This advancement will assist breeders in balancing the components

of yield (plant number, head density, grains per head, grain weight) during breeding selections [8].

In recent years, with the advancement of computer vision and deep learning theories, methods based on deep neural network technology have made significant progress in addressing general object detection problems. Consequently, numerous researchers have endeavored to apply general object detection algorithms to the field of agricultural object detection. Xu et al. [9] employed the K-means clustering algorithm to automatically segment wheat head images and extract wheat head contour features, thereby significantly enhancing the efficiency and accuracy of wheat head counting. Wang et al. [10] utilized a multilevel neural network (SSRNET) for wheat head image segmentation, achieving rapid estimation of wheat head quantities under field conditions. Khaki et al. [11] drew inspiration from crowd counting research and constructed a fast and lightweight anchor-free wheat head counting network. Amirhossein et al. [12] applied automatic object enhancement technology to deep learning models and employed a hybrid AutoOLA-DL model to improve wheat head counting performance. Wang et al. [13] developed a lightweight apple detection model based on YOLOv5s, adopting a pruning and fine-tuning approach to maintain accuracy while ensuring overall model lightweightness for deployment on portable IoT devices. Sozzi et al. [14] extensively studied the performance differences of various YOLO networks in identifying white grape applications. He et al. [15] researched and simplified the YOLOv4 network structure, employing the K-means algorithm to recluster anchors, effectively addressing wheat head detection issues in natural scenes based on unmanned aerial vehicle platforms. Gong et al. [16] optimized the Spatial Pyramid Pooling (SPP) structure within the YOLOv4 network architecture to enhance feature learning capabilities, enlarging the convolutional network's receptive field, thereby improving recognition accuracy and speed. Zhao et al. [17] improved wheat head feature extraction by adding micro-scale detection layers and setting prior anchor boxes, optimizing the YOLOv5 network structure, significantly enhancing the detection accuracy of wheat head images captured by drones. Meng et al. [18] constructed an improved YOLOv7 model by incorporating attention modules, greatly enhancing wheat head detection effectiveness, and exploring the influence of attention module quantity and placement on the model. The primary research direction of the aforementioned studies can be summarized as augmenting various specially designed modules onto existing general object detectors according to the characteristics of the target objects to be detected, thereby achieving better detection performance.

However, existing wheat head detectors exhibit poor generalization in practical applications. This is attributed to several distinctive features commonly observed in wheat head detection scenarios: Different wheat varieties exhibit significant variations in wheat head color, size, and traits at various growth stages. Wheat fields are typically densely planted, leading to substantial mutual occlusion between wheat heads in images. Factors such as equipment, lighting, environmental conditions, and wind during image acquisition contribute to significant differences between wheat heads. Wheat is widely cultivated globally, and diverse wheat field environments present considerable variations based on climate, soil, altitude, etc. These characteristics result in substantial differences between wheat field scenes. Consequently, wheat head detectors trained on datasets specifically constructed for particular studies exhibit suboptimal performance when confronted with entirely new wheat field scenes beyond the scope of their training datasets [7].

To address the generalization issues of models, a larger dataset proves effective. In response to this challenge, the GWHD dataset was introduced by [8], and subsequently, the expanded GWHD2021 dataset was released [19]. The dataset encompasses 6515 wheat head images collected from 11 different countries and regions worldwide, making it the largest and most diverse publicly available wheat head dataset to date. However, compared to the commonly used MS-COCO dataset [20] in general object detection research, which comprises over 120,000 annotated images, the annotated images in GWHD2021 constitute only 5% of MS-COCO's annotated images. This discrepancy arises from the challenging

nature of creating wheat datasets due to the characteristics mentioned earlier in wheat head detection scenarios. Even with extensive training, annotators find the annotation process challenging and time-consuming [21]. This creates a paradox where despite the abundance of wheat head image data, the high cost in terms of both money and time required for annotation hinders their utilization. This underscores a major drawback of fully supervised object detection—overreliance on annotated datasets. In contrast, SSOD (semi-supervised object detection) proves effective in addressing the issue of insufficient annotated data.

SSL (Semi-Supervised Learning) is a learning paradigm that involves constructing models using both labeled and unlabeled data. These methods enhance learning performance by incorporating additional unlabeled instances, in contrast to supervised learning algorithms that rely solely on labeled data. SSL algorithms provide a means to extract latent patterns from unlabeled examples, reducing the dependency on a large number of labels [22]. DSSL (Deep Semi-Supervised Learning) is a specific research domain within SSL, exploring how to effectively leverage both labeled and unlabeled data through deep neural networks. Numerous DSSL methods have been proposed, categorizable into five main types based on distinctive features in semi-supervised loss functions and model designs. Generative methods: employing generative models to simulate the distribution of data and subsequently leveraging the generated data for learning purposes; a paradigmatic approach in this context is the utilization of Generative Adversarial Network (GAN). Consistency regularization methods: these methods encourage models to produce similar outputs for different views or perturbations of input data, thereby improving robustness to unlabeled samples. Graph-based methods: using graph structures to represent similarity relationships between samples, these methods leverage techniques like label propagation or semi-supervised graph convolutional networks for semi-supervised learning. Pseudo-labeling methods: these methods generate pseudo-labels for unlabeled samples by using the model's predictions, treating them as real labels for training. Hybrid methods: combining elements of the above four methods in a hybrid manner [23].

SSOD represents a research domain within DSSL, aiming to explore the application of semi-supervised methods to address object detection challenges. Traditionally, in the field of computer vision, semi-supervised methods have predominantly been employed for image classification tasks rather than object detection. The investigation into semi-supervised methods for object detection heavily relies on the foundational works of Sohn et al. [24] and Unbiased Teacher [25]. Sohn et al. [24] were pioneers in introducing pseudo-labeling methods to SSOD, establishing a paradigm where teacher and student models are concurrently trained. The teacher model utilizes pseudo-labels generated from unlabeled data to guide the training of the student model. Building upon this, Unbiased Teacher [25] introduced consistency regularization methods to SSOD. The unlabeled data fed into the teacher model undergoes mild data augmentation, while the unlabeled data for the student model undergoes strong data augmentation. Notably, the parameters of the teacher model are no longer frozen but are copied from the student model through the Exponential Moving Average (EMA) approach, a technique initially proposed by Tarvainen and Valpola [26] for solving classification problems. Subsequently, most SSOD methods adhere to the paradigms established by the aforementioned works, employing hybrid methods that combine pseudo-labeling and consistency regularization. Simultaneously training both teacher and student models, these methods use the teacher model's predictions on unlabeled data with confidence exceeding a threshold as pseudo-labels to guide the training of the student model. Our research aligns with these established paradigms.

Subsequent to these developments, the primary research focus within SSOD has been on optimizing the utilization of pseudo-labels and effectively distinguishing between reliable and unreliable pseudo-labels. Influential works in this direction include the following. Soft Teacher [27], which contends that pseudo-labels with confidence below a threshold contain valuable information and should not be discarded outright. Instead, they advocate for incorporating these pseudo-labels into the loss calculation after weighting. Humble Teacher [28], employing an abundance of region proposals and soft pseudo-labels as train-

ing targets for the student model. Dense Teacher [29], a pioneering work implementing SSOD on a one-stage anchor-free detector, departing from the traditional two-stage anchor-based detectors used in prior SSOD research. Unbiased TeacherV2 [30], proposing a perspective shift in treating pseudo-label boxes not as a whole but as individual edges, placing trust only in edges with confidence surpassing a threshold. LabelMatch [31], advocating for varied confidence thresholds for pseudo-labels of different object categories rather than a uniform threshold. Efficient Teacher [32], introducing the first one-stage anchor-based semi-supervised object detector. Consistent Teacher [33], introducing three modules—Adaptive Anchor Assignment (ASA), 3D Feature Alignment Module (FAM-3D), and Gaussian Mixture Model (GMM)—to address the issue of pseudo-label oscillation during student model training. Our research draws inspiration from these aforementioned studies.

Several researchers have endeavored to employ SSOD methods to address challenges in agriculture, particularly in the realm of weed identification, a prevalent issue across various agricultural scenarios. The definition of “weed” varies depending on the cultivated crops, adding significant complexity to the annotation process. Kerdegari et al. [34] introduced a weed identification model using generative methods; opting not to rely on pseudo-labeling or consistency regularization methods, they trained a Generative Adversarial Network (GAN) on labeled data and utilized the generated weed images as part of their dataset. Jiang et al. [35] proposed a weed identification model based on graph-based methods, employing a Graph Convolutional Network (GCN) built on CNN features to construct a graph using extracted weed CNN features and their Euclidean distances. Shorewala et al. [36] presented a generalized semi-supervised weed identification model effective in both carrot and sugar beet datasets, requiring no fine-tuning. Menezes et al. [37] introduced a soybean field weed crop recognition system based on superpixels, detecting weeds in soybean fields as well as other crops beyond soybeans. Liu et al. [38] proposed a semi-supervised wheat field weed detector incorporating attention mechanisms to aid precise herbicide application. Benchallal et al. [39] introduced a weed identification system using only consistency regularization methods without employing any pseudo-labeling methods. Beyond weed identification, researchers have applied SSOD to crop recognition problems. Khaki et al. proposed DeepCorn [40], estimating corn kernel density in images of corn ears and predicting the number of kernels based on the estimated density map. Casado et al. [41] extended object detection to semi-supervised semantic segmentation, aiming to segment mature grapes in vineyards. Xu et al. [42], utilizing YOLOv5X, constructed a semi-supervised semantic segmentation model for Maize Seedling Leaf counting. Johanson et al. [43] introduced S3AD, a semi-supervised detection system based on contextual attention and selective tiling, addressing small apple detection. Across these studies, semi-supervised methods have demonstrated immense potential, effectively leveraging unlabeled data to enhance detector performance and successfully addressing the issue of insufficient annotations.

Fourati et al. [44] and Chen et al. (https://github.com/ksnxr/GWC_solution accessed on 14 October 2023) attempted to address the wheat head detection problem using semi-supervised methods. However, both approaches adopted ensemble learning, which combines predictions from multiple networks to harness the potential of multiple models. Nevertheless, this comes at the cost of significantly increased training time. In contrast to fully supervised methods, semi-supervised training itself incurs substantial time overhead due to handling unlabeled data and simultaneous training of both teacher and student models. If ensemble learning methods are additionally employed, the training time cost becomes intolerable. Therefore, we propose a novel wheat head detector, Wheat Teacher, which achieves convergence of metrics in less than a day when trained on the GWHD2021 dataset using a single RTX4090. Wheat Teacher employs the one-stage anchor-based detector, YOLOv5, as its backbone. Most semi-supervised object detection methods are implemented using either one-stage anchor-free detectors like FCOS [45] or two-stage anchor-based detectors like Faster R-CNN. However, one-stage anchor-based detectors offer advantages such as high recall, numerical stability, and fast training speed, making

them particularly suitable for applications with densely populated targets like wheat head detection. In the context of wheat head detection, Wheat Teacher incorporates our proposed innovative Pseudo-label Dynamic Allocator and Loss Dynamic Threshold. These components are specifically designed to enhance wheat head detection performance. Specifically, the main contributions of this study are as follows:

1. Proposal of Wheat Teacher, a novel one-stage anchor-based semi-supervised wheat head detector employing YOLOv5 as its backbone. Achieving metrics comparable to various fully supervised object detectors utilizing 100% labeled data, Wheat Teacher demonstrates remarkable performance with only 20% labeled data.
2. Proposal of an innovative Pseudo-label Dynamic Allocator for dynamically allocating pseudo-labels.
3. Proposal of an innovative Loss Dynamic Threshold for adaptively filtering out irrelevant losses.

The remaining sections of this paper are organized as follows: Section 2 provides an introduction to the model and network architecture, offering detailed explanations of our semi-supervised training methodology. In Section 3, experimental details and results will be presented, comparing them with the results obtained through fully supervised methods. Subsequently, in Section 4, we discuss the methods employed in this study and propose future research directions. Finally, in Section 5, we conclude this paper.

2. Methods

2.1. Wheat Teacher

Wheat Teacher is a one-stage anchor-based semi-supervised wheat head detector that employs hybrid methods in the semi-supervised approach. The network architecture adopts a teacher-student paradigm, utilizing YOLOv5 as the backbone. Additionally, novel components, namely the Pseudo-label Dynamic Allocator and Loss Dynamic Threshold, are introduced specifically for wheat head detection scenarios to allocate pseudo-labels and filter losses.

The hybrid methods employed by Wheat Teacher combine pseudo-labeling methods and consistency regularization methods. These methods are based on two assumptions. The first assumption posits that using different data augmentations on the same image should yield consistent predictions from the same model. The second assumption suggests that if the predictions on the same image differ after weak and strong data augmentations, the predictions on the weakly augmented image should be considered more accurate, as it is easier to predict.

Building upon these assumptions, Wheat Teacher employs a teacher–student paradigm, consisting of two models: the teacher model and the student model. The basic training process involves weakly augmenting unlabeled data images and inputting them into the teacher model for predictions. The same images are then strongly augmented and input into the student model for predictions. The teacher model’s predictions with confidence exceeding a threshold are treated as the ground truth for the unlabeled data images. These predictions are compared with the student model’s predictions, guiding the calculation of the student model’s loss. The predictions from the teacher model serve as pseudo-labels.

The training process of Wheat Teacher comprises several steps. Firstly, in the burn-in stage, the backbone model is pre-trained using labeled data. After pre-training, the model is duplicated into the teacher and student models. In the semi-supervised stage, unlabeled data undergo weak data augmentation (e.g., horizontal and vertical flips, cropping and resizing, rotations, mosaic) and are input into the teacher model to generate predictions. Pseudo-labels from the teacher model, exceeding a confidence threshold, are considered the ground truth for unlabeled data. After generating and filtering pseudo-labels, the unlabeled data previously input into the teacher model undergo strong data augmentation (e.g., mixup, random erasing). These augmented unlabeled data, along with pseudo-labels (treated as annotations for unlabeled data), are concatenated with weakly augmented labeled data and input into the student model. The student model simultaneously calculates

losses for both unlabeled and labeled data, updating its parameters through backpropagation. After each iteration, the parameters of the teacher model are updated using an Exponential Moving Average (EMA) of the student model's parameters. This process repeats in subsequent iterations. Figure 1 illustrates the specific pipeline of Wheat Teacher.

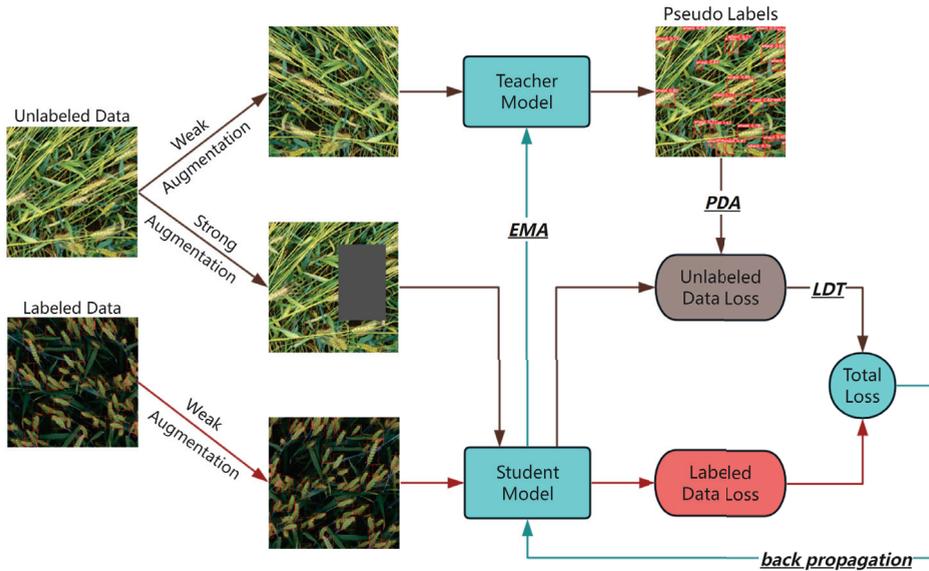


Figure 1. The pipeline of Wheat Teacher.

2.2. Pseudo-Label Dynamic Allocator (PDA)

The quality of predictions generated by the teacher model for unlabeled data is inconsistent, with most predictions being unsuitable as pseudo-labels for guiding the student model. Hence, there is a need for pseudo-label filtering. A common practice involves setting a pseudo-label threshold, categorizing pseudo-labels into reliable and unreliable binary distinctions. Early semi-supervised methods would outright discard unreliable pseudo-labels. However, contemporary semi-supervised approaches posit that even unreliable pseudo-labels may contain valuable information and should not be discarded outright; instead, they can be processed and utilized. For instance, allowing the loss computed from unreliable pseudo-labels to undergo weighted processing before backpropagation [27]. While setting a pseudo-label threshold allows for a quick and simple filtering of pseudo-labels, the overall scores of pseudo-labels tend to increase gradually throughout the entire semi-supervised training process. This phenomenon makes it challenging for the model to converge. We contend that the model's outputs consistently comprise both relatively correct and relatively incorrect predictions. Therefore, the threshold should dynamically adjust based on the overall distribution of pseudo-labels in each iteration. Additionally, we posit that a binary categorization of pseudo-labels is inappropriate. While unreliable pseudo-labels may contain useful information, there exists substantial variability among unreliable pseudo-labels. Some unreliable pseudo-labels contain minimal useful information, and attempting to leverage them may inevitably introduce noise, resulting in an unfavorable trade-off. Therefore, a more refined differentiation of unreliable pseudo-labels is warranted.

Therefore, we introduce the innovative Pseudo-label Dynamic Allocator, a module capable of autonomously adjusting thresholds based on the confidence distribution of pseudo-labels and categorizing them. The Pseudo-label Dynamic Allocator incorporates two dynamic thresholds: the high-confidence threshold and the low-confidence threshold.

The low-confidence threshold is set to 0.1 times the sum of confidences for all pseudo-labels generated in one iteration. Pseudo-labels with confidence below this threshold are considered unreliable and are directly discarded without participating in loss computation. After removing low-confidence pseudo-labels, the average confidence of all remaining pseudo-labels becomes the high-confidence threshold. Pseudo-labels with confidence above this threshold are considered reliable and directly contribute to loss computation. Pseudo-labels with confidence between the low- and high-confidence thresholds are considered uncertain. If an object's confidence in an uncertain pseudo-label exceeds 0.8, this pseudo-label participates in IoU loss computation. The specific process of the Pseudo-label Dynamic Allocator is outlined using pseudocode in Algorithm 1.

Algorithm 1 Pseudo-label Dynamic Allocator

```

1: for  $I = I_1, \dots, I_N$  do                                     ▷ In each iteration,  $I$  means one iteration.
2:    $\alpha = L_{obj} = L_{reg} = 0$ 
3:    $P^r = \emptyset$ 
4:   for  $P = P_1, \dots, P_N$  do                                   ▷  $P$  means pseudo-label.
5:      $\alpha = \alpha + P_{conf}$                                        ▷  $P_{conf}$  means confidence of pseudo-label.
6:   end for
7:    $\tau_l = \alpha * 0.1$                                            ▷ Calculate the low confidence threshold  $\tau_l$ .
8:   for  $P = P_1, \dots, P_N$  do
9:     if  $P_{conf} \geq \tau_l$  then
10:       $P^r \cup \{P\}$                                              ▷  $P^r$  means reliable pseudo-labels.
11:    end if
12:  end for
13:   $\tau_h = \frac{1}{N} \sum_{n=1}^N P_{conf}^{rn}$                                ▷ Calculate the high confidence threshold  $\tau_h$ .
14:  for  $P^r = P_1^r, \dots, P_N^r$  do
15:    if  $P_{Conf}^r \geq \tau_h$  then
16:       $L_{obj} = L_{obj} + CE(P^r)$                                    ▷  $L_{obj}$  means object loss.
17:       $L_{reg} = L_{reg} + CIoU(P^r)$                                ▷  $L_{reg}$  means regression loss.
18:    else if  $P_{objconf}^r \geq 0.8$  then                             ▷  $P_{objconf}^r$  means object confidence.
19:       $L_{reg} = L_{reg} + CIoU(P^r)$ 
20:    end if
21:  end for
22:  return  $L_{obj}, L_{reg}$ 
23: end for

```

With the introduction of the Pseudo-label Dynamic Allocator, the loss formulation for Wheat Teacher is as follows:

$$L = L_s + \lambda L_u \quad (1)$$

In the formulation, where L represents the total loss, L_s is the loss computed from labeled data, L_u is the loss computed from unlabeled data, and λ is a weighted coefficient used to balance the losses between full-supervision and semi-supervision, we set it to 3. It is important to note that, since Wheat Teacher only detects the “wheat head” object, class loss is not computed.

The definition of L_s is as follows:

$$L_s = \sum_{n=1}^N (CIoU(X_n^{reg}, Y_n^{reg}) + CE(X_n^{obj}, Y_n^{obj})) \quad (2)$$

$CIoU$ represents the Complete Intersection over Union loss function, CE represents the cross-entropy loss function, X_n represents the output of the student model, and Y_n represents the ground truth.

The definition of L_u is as follows:

$$L_u = L_u^{reg} + L_u^{obj} \tag{3}$$

$$L_u^{reg} = \sum_{n=1}^N (\mathbb{I}_{\{p_n \geq \tau_h \text{ or } (p_n \geq \tau_l \text{ and } o\hat{b}j_n > 0.8)\}} CIoU(X_n^{reg}, \hat{Y}_n^{reg})) \tag{4}$$

$$L_u^{obj} = \sum_{n=1}^N (\mathbb{I}_{\{p_n \geq \tau_h\}} CE(X_n^{obj}, \hat{Y}_n^{obj})) \tag{5}$$

\hat{Y}_n represents the pseudo-label, p_n represents the overall confidence of this pseudo-label, $o\hat{b}j_n$ represents the object confidence of this pseudo-label, τ_l represents the low-confidence threshold, τ_h represents the high-confidence threshold, $\mathbb{I}_{\{\cdot\}}$ represents an indicator function. If condition $\{\cdot\}$ is satisfied, the output is 1; otherwise, it is 0.

2.3. Loss Dynamic Threshold (LDT)

Even after filtering pseudo-labels through the Pseudo-label Dynamic Allocator, the remaining pseudo-labels are not entirely accurate because they are based on predictions from the teacher model, which is not flawless. Through experiments, we observed that in images treated as unlabeled data, for the same wheat head, the differences between the predictions of the student model, pseudo-labels, and ground truth are often minimal. However, the gap between the predictions of the student model and pseudo-labels can sometimes be larger than the gap between the predictions of the student model and the ground truth. We attribute this phenomenon to the fact that Wheat Teacher employs YOLOv5 as its backbone, which is a one-stage anchor-based detector. This type of detector outputs a large number of prediction boxes, necessitating post-processing to filter out predictions. YOLOv5 uses Non-Maximum Suppression (NMS) as its post-processing method, removing all candidate boxes except the optimal one. In the typical scenario of YOLOv5 forward inference, NMS has been empirically proven to be a highly effective post-processing technique. However, in the context of semi-supervised training, there is uncertainty regarding whether the bounding box selected by NMS as the optimal candidate, when compared to less optimal candidates, exhibits a minimal difference with the true ground truth.

The subtle variations in loss due to the uncertainty of pseudo-labels do not significantly benefit the performance of the student model. In fact, they may even mislead the student model, leading to unnecessary oscillations during the training process. Therefore, we innovatively propose the Loss Dynamic Threshold. In each iteration, it is crucial to determine which losses induced by pseudo-labels are necessary and which are unnecessary, in order to reduce unnecessary oscillations in the model training. We advocate for dynamically adjusting the Loss Dynamic Threshold based on the total loss value generated by pseudo-labels in each iteration. Subsequently, losses below this threshold should be discarded. However, if the model calculates the loss but does not immediately backpropagate and clear gradients after each loss computation, attempting to backpropagate all losses at once, the GPU must store the complete computation graph for each loss to ensure correct backpropagation. This results in a multiplicative increase in model memory usage. While setting the batch size for each iteration to 1 effectively addresses this issue, it leads to significant memory wastage, severely slowing down the model training speed. Therefore, to ensure training efficiency, we set the Loss Dynamic Threshold for each iteration to the peak value of the lowest 10% of losses generated by pseudo-labels in the previous iteration. Losses below the Loss Dynamic Threshold are considered unnecessary and are set to 0, while losses exceeding the threshold undergo normal backpropagation. The specific process of Loss Dynamic Threshold is outlined in Algorithm 2 using pseudocode.

Algorithm 2 Loss Dynamic Threshold

```

1: for  $l = I_1, \dots, I_N$  do                                ▷ In each iteration,  $l$  means one iteration.
2:   for  $l = l_1, \dots, l_N$  do
3:     if  $l \geq \tau$  then                                    ▷  $\tau$  means dynamic loss threshold.
4:       BackPropagation( $l$ )                               ▷ Backpropagate through the loss  $l$ .
5:        $L \cup \{l\}$                                          ▷  $L$  is used to record all losses.
6:     else
7:        $L \cup \{l\}$ 
8:     end if
9:   end for
10:  Sort( $L$ )                                               ▷ Sort  $L$  in ascending order.
11:   $\tau = L[\lceil 0.1 \times \text{length}(L) \rceil]$               ▷ Update  $\tau$ .
12:   $L = \emptyset$ 
13: end for

```

2.4. Backbone

The mainstream algorithms for general object detection can be broadly categorized into two types: two-stage detectors and one-stage detectors. Two-stage detectors, relying on region proposals, extract features through multi-stage networks, achieving higher accuracy at the cost of slower detection speed. Representative algorithms in this category include the R-FCN and R-CNN series (comprising R-CNN [46], Fast-RCNN [47], Faster-RCNN [48], Mask-RCNN [49]). On the other hand, one-stage detectors, based on region regression, directly extract features using a single-stage network, resulting in extremely fast detection. Notable algorithms in this category encompass SSD [50] and the YOLO series (encompassing YOLO [51], YOLO9000 [52], YOLOv3 [53], YOLOv4 [54], YOLOv5 (<https://github.com/ultralytics/yolov5> accessed on 10 September 2023), YOLOX [55], YOLOv6 [56], YOLOv7 [57], etc.).

Given the application scenario of this research, Wheat Teacher opted for YOLOv5 as the backbone. YOLOv5 (You Only Look Once version 5) is a one-stage anchor-based detector, primarily comprising four components: input, backbone, neck, and head. The input component is responsible for image preprocessing, including adaptive anchor computation, various data augmentations, image data scaling, and other operations. YOLOv5's backbone is CSPDarknet53, which includes three main modules: CBS module, CSP module, and SPPF module. The core CBS module consists of three parts: two-dimensional convolution (Conv), batch normalization (BN), and Sigmoid-weighted Linear Unit activation function (SiLU). The backbone extracts feature information from the image and combines it to form feature maps of different granularities. The neck section combines feature maps and extracts features through Upsample and Concatenate operations to enhance the robustness of the detection network. The head section outputs the target detection result, and the number of branches at the output varies depending on the detection scenario. Figure 2 illustrates the specific network structure of YOLOv5.

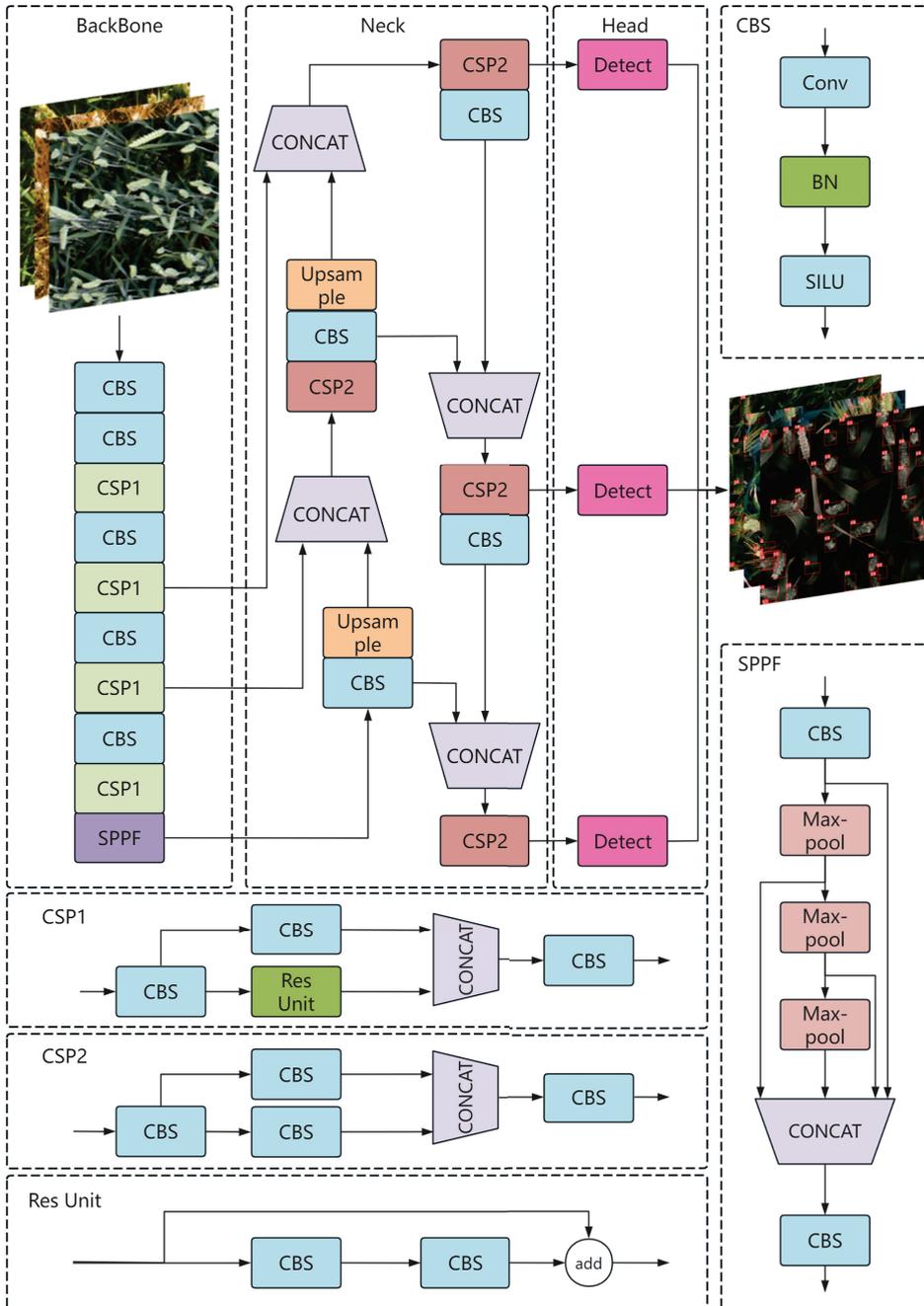


Figure 2. The network structure of YOLOv5.

3. Experimental Results

3.1. Dataset

In the specific domain of wheat head detection, research on existing technologies indicates limitations in proposed solutions, primarily addressing images from controlled

environments rather than those directly captured in the field [58]. Another limitation is that models in both the training and testing phases focus on the same type of wheat, leading to overfitting to that specific wheat type [7]. To overcome these limitations, this study decided to utilize the GWHD2021 dataset.

The Global Wheat Head Detection Dataset 2021 (GWHD2021 [19]) (<http://www.global-wheat.com/> accessed on 6 October 2023) is a large-scale wheat head object detection dataset based on optical images. It is currently the largest publicly available Wheat Head dataset. The Global Wheat Head Detection Competition was initiated by the Global Wheat Dataset consortium in 2020, providing the public dataset Global Wheat Head Detection Dataset 2020 (GWHD2020) for the competition. In 2021, the Global Wheat Dataset consortium launched the Global Wheat Challenge 2021 and expanded the GWHD2020 dataset in terms of wheat head diversity, label reliability, and data size, releasing GWHD2021.

GWHD2021 comprises 6515 RGB wheat head images collected from 11 different countries and regions worldwide, including Europe, North America, Asia, and Australia. Each image has a resolution of 1024×1024 pixels, and the labels include a total of over 270,000 wheat heads. The row spacing in the wheat fields varies from 12.5 cm to 30.5 cm. In addition to variations in row spacing, each field has different planting densities. Furthermore, the soil characteristics in the growth areas vary from mountainous regions to traditional irrigated farmlands, leading to differences in color and lighting conditions. These images were captured using various cameras at different distances from wheat heads (ranging from 1.8 m to 3 m). All these factors ensure the diversity of wheat heads in the GWHD2021 dataset. Some sample images from GWHD2021 are shown in Figure 3.

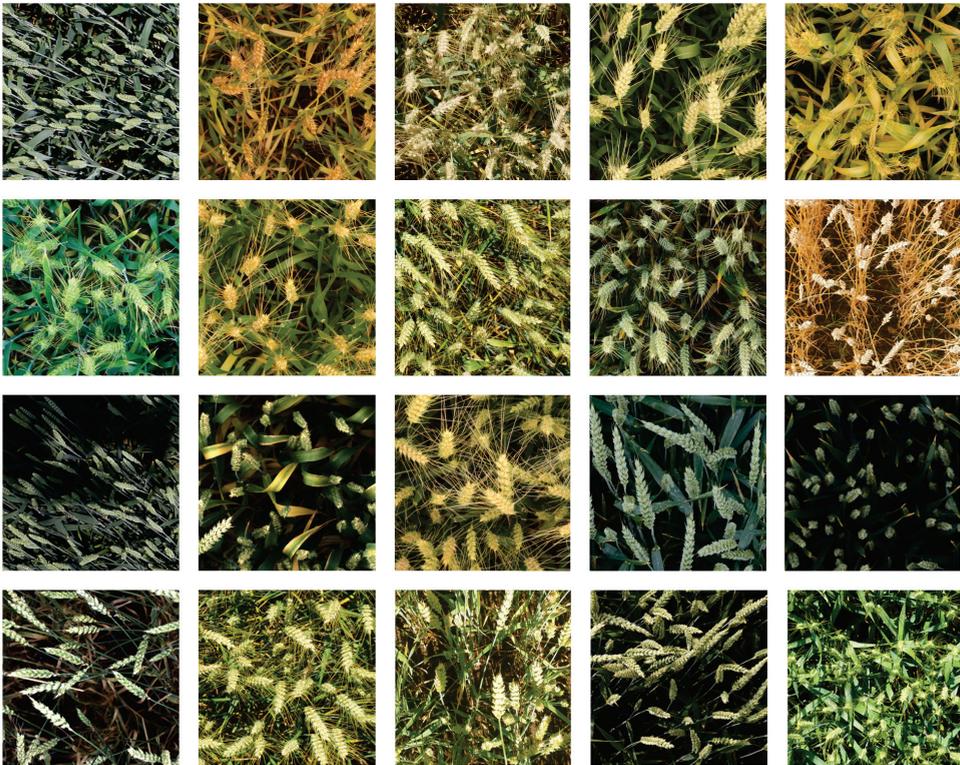


Figure 3. Some example images in the GWHD2021.

The semi-supervised training method requires a significant amount of images as unlabeled data. To ensure a fair comparison with other methods, this study did not introduce additional images beyond the GWHD2021 dataset for extra training. Instead, a total of 6000 images were randomly selected from the GWHD2021 dataset, which originally comprises 6515 images, to form a new training set. The remaining 515 images were used as a new validation set for reporting performance.

3.2. Evaluation Metric

To ensure the fairness and comparability of experimental results, widely used evaluation metrics in existing object detection methods were employed to assess the detection performance of the proposed model. These metrics include accuracy, recall, F1 score, and mean average precision (mAP). Specifically, accuracy measures the precision of the algorithm, while recall gauges the completeness of the image recognition results. The F1 score serves as a comprehensive evaluation metric for the model's detection accuracy, being the harmonic mean of precision and recall.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (6)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (7)$$

$$F1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (8)$$

In the above formulas, true positives (TP) represent the number of correctly classified wheat heads, where both the detection result and ground truth are wheat heads. False negatives (FN) indicate the number of incorrectly classified wheat heads, where the detection result is the background, and the ground truth is a wheat head. False positives (FP) denote the number of incorrectly classified backgrounds, where the detection result is a wheat head, and the ground truth is the background. True negatives (TN) signify the number of correctly classified backgrounds, where both the detection result and ground truth are backgrounds.

$$AP_n = \int_0^1 P(R) dR \quad (9)$$

$$mAP = \frac{1}{N} \sum_{n=1}^N AP_i \quad (10)$$

mAP (Mean Average Precision) is a crucial evaluation metric for measuring the performance of object detectors. It is defined as the area under the precision–recall curve, averaged across all classes, providing a measure of the overall performance of the model. mAP@0.5 assesses the model's performance on partially overlapping objects at an IoU (Intersection over Union) threshold of 0.5. mAP@0.5:0.95 further considers IoU thresholds ranging from 0.5 to 0.95, comprehensively evaluating the model's performance across various degrees of target overlap. As wheat head is the sole detection target in this study, the mAP value is equivalent to the AP value in this context.

3.3. Hyperparameter Settings

Wheat Teacher employs YOLOv5L 7.0 version as its backbone, with an input image size set to 1024 and a batch size of 6. The learning rate remains fixed at 0.01 throughout the training process. The rate for the exponential moving average (EMA) is set to 0.999. The training is conducted for a total of 350 epochs, comprising a Burn-in stage of 100 epochs during full-supervision training and a subsequent Semi-supervised stage of 150 epochs. In this study, for fair performance comparison, no transfer learning is applied, and no pre-trained weights are utilized. All models are initialized randomly.

3.4. Experimental Results Using Different Proportions of Labeled Data

In the experiment, we randomly partition the train set images into labeled and unlabeled data at different ratios (1% and 99%, 2% and 98%, 5% and 95%, 10% and 90%, 20% and 80%). The labels of unlabeled data are not used in the experiment. In a single experiment, Wheat Teacher first enters the burn-in stage, conducting fully supervised training using labeled data, and reports performance on the validation set after training. Subsequently, Wheat Teacher enters the semi-supervised stage, conducting training using both labeled and unlabeled data, and reports performance again on the validation set after training. The experimental results are shown in Table 1.

Table 1. Experimental results of labeled data with different proportions.

Labeled Data	Unlabeled Data	mAP@0.5	mAP@0.5:0.95	Precision	Recall	F1-Score
1%	—	14.5	5.1	22.1	24.0	23.0
1%	99%	40.3	17.6	57.0	38.2	45.7
2%	—	33.1	18.3	43.1	41.8	42.4
2%	98%	70.8	36.6	72.1	64.5	68.1
5%	—	78.6	38.9	81.3	70.9	75.7
5%	95%	83.7	45.5	84.3	78.9	81.5
10%	—	87.2	46.2	87.6	79.4	83.2
10%	90%	90.8	51.2	90.9	83.3	86.9
20%	—	90.7	49.3	90.9	84.3	87.4
20%	80%	92.8	53.1	92.5	86.9	89.6

The experimental results indicate that by employing semi-supervised training with unlabeled data, Wheat Teacher achieved a significant improvement in metrics compared to the fully supervised training stage at any data ratio. Specifically, when using only 1% labeled data, Wheat Teacher achieved a 25.8% improvement in mAP0.5, a 12.5% improvement in mAP0.5:0.95, a 34.9% improvement in Precision, a 14.2% improvement in Recall, and a 22.7% improvement in F1-Score. When using only 2% labeled data, Wheat Teacher achieved a 37.7% improvement in mAP0.5, a 18.3% improvement in mAP0.5:0.95, a 29% improvement in Precision, a 22.7% improvement in Recall, and an 25.7% improvement in F1-Score. With only 5% labeled data, Wheat Teacher obtained a 5.1% improvement in mAP0.5, a 6.6% improvement in mAP0.5:0.95, a 3% improvement in Precision, a 8% improvement in Recall, and a 5.8% improvement in F1-Score. Using 10% labeled data, Wheat Teacher achieved a 3.6% improvement in mAP0.5, a 5% improvement in mAP0.5:0.95, a 3.3% improvement in Precision, a 3.9% improvement in Recall, and a 3.7% improvement in F1-Score. With only 20% labeled data, Wheat Teacher obtained a 2.1% improvement in mAP0.5, a 3.8% improvement in mAP0.5:0.95, a 1.6% improvement in Precision, a 2.6% improvement in Recall, and a 2.2% improvement in F1-Score.

Figures 4–6 depict the variations of Loss, mAP@0.5, and mAP@0.5:0.95 throughout the experimental training process, respectively. It can be observed that after concluding the fully supervised training phase, at the commencement of the semi-supervised training phase, the model's accuracy undergoes significant oscillations. However, during the subsequent training, the accuracy rapidly rebounds, eventually surpassing the results of fully supervised training. The oscillations observed after entering the semi-supervised training phase are one of the characteristics of semi-supervised training methods. However, in the general object detection domain, the amplitude of these oscillations is generally limited and should not be as extensive as depicted in the above figures. We attribute this phenomenon to the specific application scenario of this research, characterized by a single, highly dense object class. Eliminating such oscillations will be a future research direction.

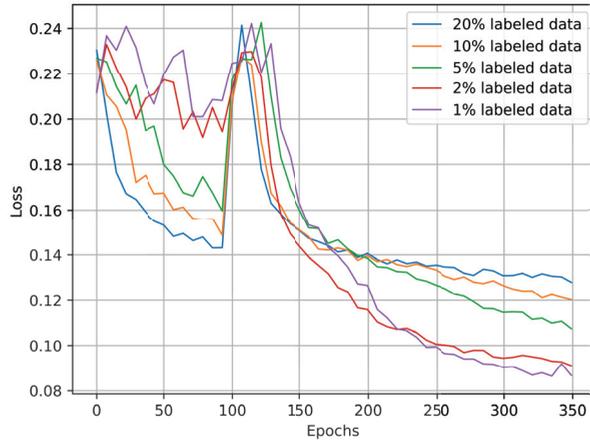


Figure 4. Graphs depicting loss curves.

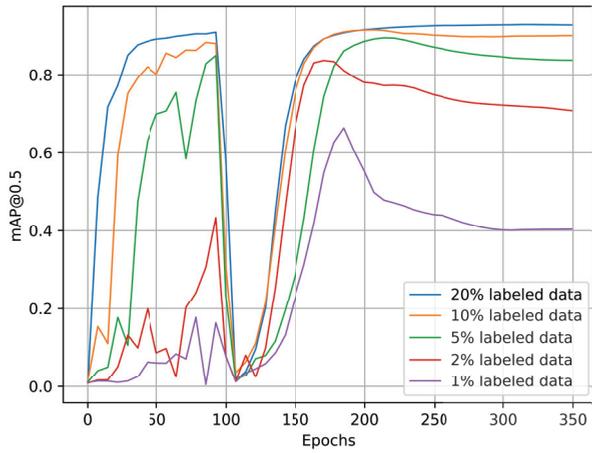


Figure 5. Graphs depicting mAP0.5 curves.

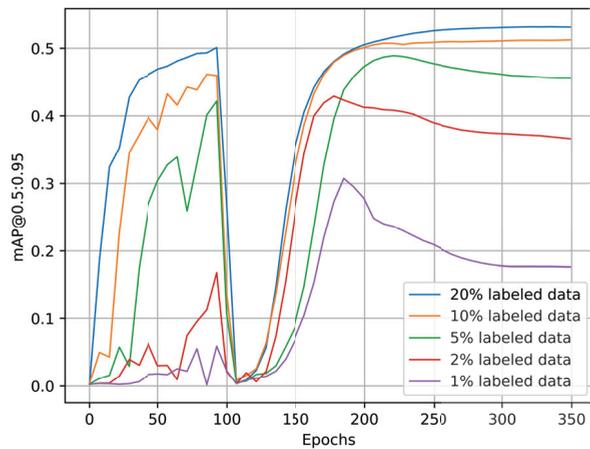


Figure 6. Graphs depicting mAP0.5:0.95 curves.

Figures 7 and 8 displays the precision–recall (PR) curve and the F1-Score curve after the completion of training. From the results above, it is evident that as the proportion of labeled data used in the fully supervised training phase increases, there is a noticeable improvement in the model’s accuracy. This indicates that the model has not overfit, and the semi-supervised approach indeed has the potential to further enhance the model’s accuracy. Semi-supervised methods show great promise in the field of wheat head detection.

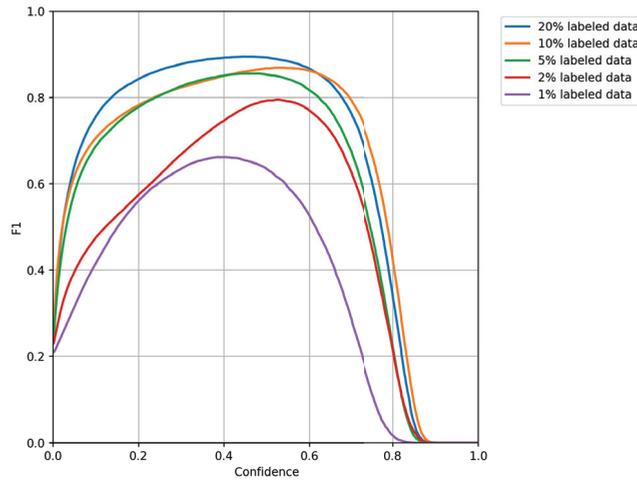


Figure 7. Graphs depicting F1 curves.

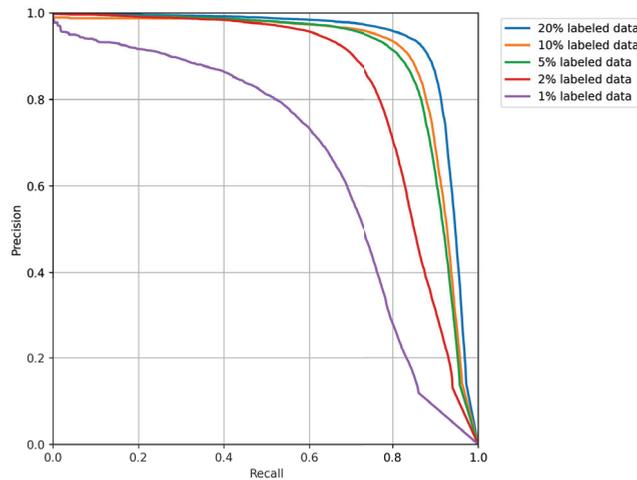


Figure 8. Graphs depicting PR curves.

3.5. Experimental Results Compared with Other Fully Supervised Object Detectors

The semi-supervised training method requires a substantial amount of images as unlabeled data. When compared with other fully supervised methods, we randomly divided the train set into labeled data and unlabeled data at a ratio of 20% to 80%. The labels of unlabeled data are not utilized by Wheat Teacher during the experiments. The experimental results are presented in Table 2.

We compared the predictions of Wheat Teacher with four fully supervised object detection models: Faster-RCNN, YOLOv5, YOLOX, and YOLOv7. The results show that

with only 20% of the training set labels, Wheat Teacher achieved highly competitive results. Wheat Teacher's metrics surpassed YOLOX, significantly outperformed Faster-RCNN, and the differences with YOLOv5 and YOLOv7 were within 1%, demonstrating the potential of semi-supervised methods in addressing wheat head detection. Figure 9 provides examples of prediction differences between Wheat Teacher, YOLOv5, and YOLOv7.

Table 2. Experimental results compared with other fully supervised object detectors on GWHD2021.

Type	Method	mAP@0.5	Precision	Recall	F1-Score
Two-stage Anchor-based	Faster-RCNN	79.5	77.5	83.1	80.2
One-stage Anchor-based	YOLOv7	93.6	92.5	87.4	89.8
One-stage Anchor-free	YOLOX	90.6	91.7	85.2	88.3
Backbone	YOLOv5	93.1	92.2	87.1	89.5
Ours	Wheat Teacher	92.9	92.4	86.8	89.5

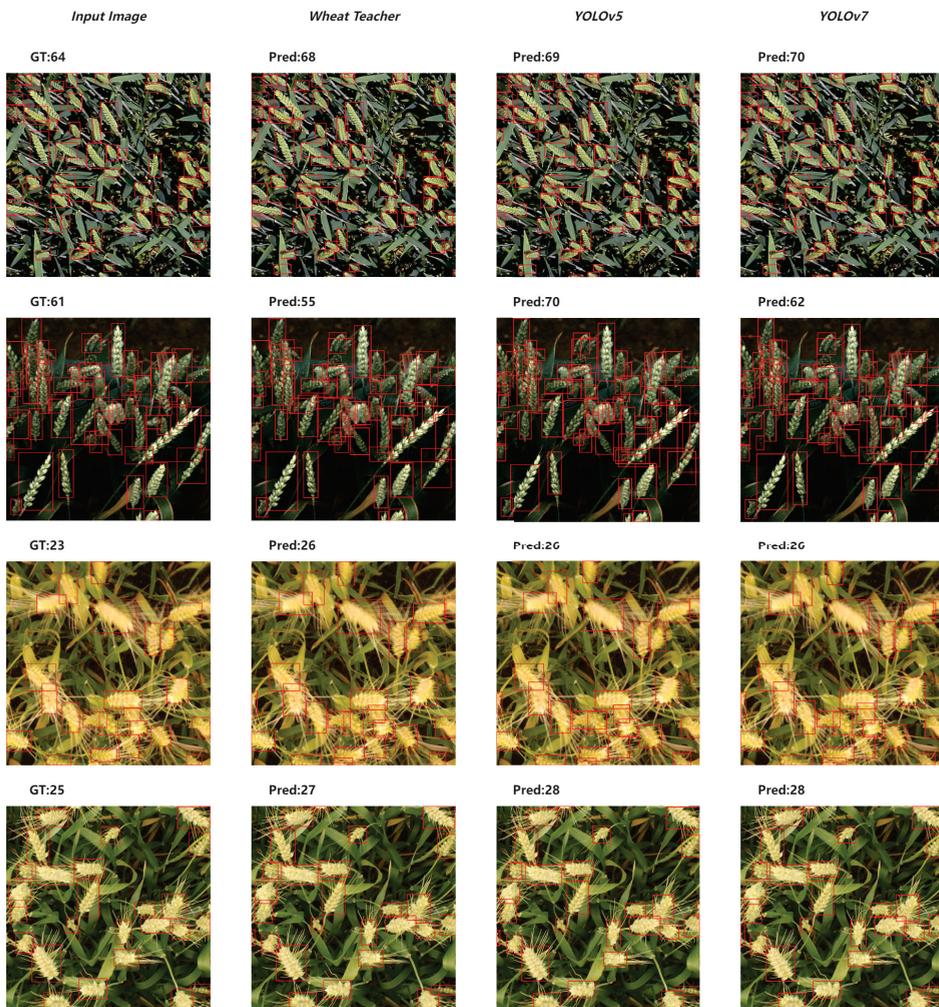


Figure 9. Differences in predictions under four different situations. The red box represents the wheat head. GT represents the number of wheat heads ground truth, and Pred represents the number of wheat heads predicted by the model.

3.6. Experimental Results Compared with Other Semi-Supervised Object Detectors

We conducted comparative experiments between Wheat Teacher and two existing semi-supervised object detection methods, Unbiased Teacher [25] and Efficient Teacher [32], at GWHD2021. The dataset was randomly sampled to consist of 20% labeled data and 80% unlabeled data, with all models utilizing the same random sampling scheme. The experimental results are presented in Table 3.

Table 3. Experimental results compared with other semi-supervised object detectors on GWHD2021.

Method	mAP@0.5	Precision	Recall	F1-Score	Training Time
Unbiased Teacher [25]	90.8	90.5	83.9	87.1	29 h
Efficient Teacher [32]	91.1	91.1	85.9	88.4	20 h
Wheat Teacher	92.9	92.4	86.8	89.5	22 h

From the experimental results, it can be observed that our method outperforms both Unbiased Teacher and Efficient Teacher across all metrics, except for a slightly longer training time compared to Efficient Teacher. This improvement may be attributed to the additional processing of the student network's predictions in our method, such as filtering losses, compared to Efficient Teacher.

3.7. Ablation Experiment

To validate the impact of our proposed Pseudo-label Dynamic Allocator and Loss Dynamic Threshold, we conducted a series of ablation experiments. We first removed the Loss Dynamic Threshold from Wheat Teacher and replaced the Pseudo-label Dynamic Allocator with a static threshold (set to 0.7), then reported the performance. Next, we replaced the static threshold with the Pseudo-label Dynamic Allocator and reported the performance. Finally, we introduced the Loss Dynamic Threshold into the network and reported the performance. All ablation experiments were conducted using 10% labeled data and 90% unlabeled data for training, and performance was evaluated on the validation set. The experimental results are presented in Table 4.

Table 4. Ablation experiment results.

Pseudo-Label Threshold	Loss Threshold	mAP@0.5
Static-Threshold (0.7)	×	88.3
PDA	×	89.9
PDA	LDT	90.8

The results indicate that by introducing the Pseudo-label Dynamic Allocator, Wheat Teacher's mAP0.5 improved by 1.6%. Additionally, with the introduction of the Loss Dynamic Threshold, Wheat Teacher's mAP0.5 further increased by 0.9%, confirming the effectiveness of both the Pseudo-label Dynamic Allocator and the Loss Dynamic Threshold.

4. Discussion

Wheat Teacher is implemented based on a one-stage anchor-based detector, whereas most semi-supervised object detection methods are implemented on one-stage anchor-free detectors and two-stage anchor-based detectors. The reason for not using one-stage anchor-based detectors in many semi-supervised object detection methods lies in the fact that, compared to anchor-free detectors with no anchor design and two-stage detectors with multi-level filtering, one-stage anchor-based detectors tend to output more densely packed prediction boxes. This leads to a severe imbalance between positive and negative samples during semi-supervised training, and the issue of imbalance becomes more pronounced in detection scenarios with denser targets, such as wheat fields. To address this imbalance, we propose the Pseudo-label Dynamic Allocator and Loss Dynamic Threshold.

Before the one-stage anchor-based detector outputs the final predictions, it undergoes non-maximum suppression (NMS) to remove most irrelevant anchors. In essence, NMS can be considered as a filtering process on pseudo-labels based on a threshold, where only bounding boxes passing through the filter are deemed trustworthy pseudo-labels. Similar to NMS, which compares bounding boxes with each other, we believe that, instead of using a fixed static threshold set by hyperparameters, a more effective threshold should be determined by comparing the differences between bounding boxes. We propose the Pseudo-label Dynamic Allocator, which filters pseudo-labels based on the relationship between the confidence of pseudo-labels and the sum of pseudo-label confidences. Specifically, we set 0.1 times the sum of pseudo-label confidences as the low-confidence threshold and the mean of pseudo-label confidences as the high-confidence threshold.

Even after filtering, the remaining pseudo-labels inevitably contain errors. In dense target prediction scenarios, the model generates numerous bounding boxes. These bounding boxes, influenced by each other during non-maximum suppression, might replace more accurate bounding boxes due to subtle differences in confidence. While such differences might be inconsequential in other application scenarios, in semi-supervised training, these subtle differences can accumulate and mislead the student model throughout the training process. The objective of semi-supervised training is to enable the student model to learn correct features from unlabeled data. If the learned features are very small or even incorrect, we should abandon learning these features during training. Therefore, we propose the Loss Dynamic Threshold, which considers the peak in the lowest 10% of losses for each iteration as the threshold. This threshold is then used to filter out losses lower than the threshold in the next iteration. The effectiveness of Wheat Teacher in wheat head detection is adequately demonstrated in the experiments, and the roles of Pseudo-label Dynamic Allocator and Loss Dynamic Threshold are extensively validated in the ablation studies.

During the experimental process, we observed limitations when applying semi-supervised training methods to wheat head detection. If the labeled data is insufficient, it results in the initial model trained in the fully supervised training phase being suboptimal. Consequently, even when transitioning to semi-supervised training, the model's performance ceiling remains low. However, when an ample amount of labeled data is used in the fully supervised training phase, semi-supervised training methods demonstrate their potential. In comparison to other fully supervised object detection models that converge rapidly, semi-supervised training methods can leverage a large quantity of unlabeled data to continuously enhance their detection performance, exhibiting a higher performance ceiling than the backbone alone. Addressing the challenge of training a superior initial model with fewer labeled data points is expected to be a crucial avenue for future research.

5. Conclusions

This article delves into the significance and challenges of wheat head object detection. It introduces semi-supervised object detection methods and their applications in agriculture, subsequently proposing a one-stage anchor-based semi-supervised wheat head detector named "Wheat Teacher." Wheat Teacher amalgamates two semi-supervised methods, pseudo-labeling and consistency regularization, and integrates two novel dynamic threshold components, namely the Pseudo-label Dynamic Allocator and Loss Dynamic Threshold. We conducted detailed experiments on the largest public wheat head dataset, GWHD2021. Compared with various types of detectors, Wheat Teacher achieved an mAP_{0.5} of 92.8% with only 20% labeled data. This result surpassed the test outcomes of two fully supervised object detection models trained with 100% labeled data, and the difference with the other two fully supervised models trained with 100% labeled data was within 1%. Furthermore, Wheat Teacher shows improvements of 2.1%, 3.6%, 5.1%, 37.7%, and 25.8% in mAP_{0.5} across different labeled data usage ratios of 20%, 10%, 5%, 2%, and 1%, respectively. These experiments validate the effectiveness of our semi-supervised approach, highlighting the significant potential of Wheat Teacher in wheat head detection.

Author Contributions: Conceptualization, R.Z. and Y.S.; methodology, R.Z.; software, R.Z.; validation, R.Z.; formal analysis, R.Z.; investigation, R.Z.; resources, R.Z.; data curation, R.Z.; writing—original draft preparation, R.Z.; visualization, R.Z.; writing—review and editing, Y.S., M.Y., Z.Q., L.Z. and W.L.; supervision, Y.S.; project administration, Y.S.; funding acquisition, Y.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China under Grant 61972147.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Reynolds, M.; Chapman, S.; Crespo-Herrera, L.; Molero, G.; Mondal, S.; Pequeno, D.N.; Pinto, F.; Pinera-Chavez, F.J.; Poland, J.; Rivera-Amado, C.; et al. Breeder friendly phenotyping. *Plant Sci.* **2020**, *295*, 110396. [CrossRef]
2. Crain, J.; Mondal, S.; Rutkoski, J.; Singh, R.P.; Poland, J. Combining high-throughput phenotyping and genomic information to increase prediction and selection accuracy in wheat breeding. *Plant Genome* **2018**, *11*, 170043. [CrossRef]
3. Hund, A.; Kronenberg, L.; Anderegg, J.; Yu, K.; Walter, A. Non-invasive field phenotyping of cereal development. In *Advances in Breeding Techniques for Cereal Crops*; Burleigh Dodds Science Publishing: Sawston, UK, 2019; pp. 249–292.
4. Walter, A.; Liebisch, F.; Hund, A. Plant phenotyping: From bean weighing to image analysis. *Plant Methods* **2015**, *11*, 14. [CrossRef]
5. Hasan, M.M.; Chopin, J.P.; Laga, H.; Miklavcic, S.J. Detection and analysis of wheat spikes using convolutional neural networks. *Plant Methods* **2018**, *14*, 100. [CrossRef] [PubMed]
6. Jin, X.; Liu, S.; Baret, F.; Hemerlé, M.; Comar, A. Estimates of plant density of wheat crops at emergence from very low altitude UAV imagery. *Remote Sens. Environ.* **2017**, *198*, 105–114. [CrossRef]
7. Madec, S.; Jin, X.; Lu, H.; De Solan, B.; Liu, S.; Duyme, F.; Heritier, E.; Baret, F. Ear density estimation from high resolution RGB imagery using deep learning technique. *Agric. For. Meteorol.* **2019**, *264*, 225–234. [CrossRef]
8. David, E.; Madec, S.; Sadeghi-Tehran, P.; Aasen, H.; Zheng, B.; Liu, S.; Kirchgessner, N.; Ishikawa, G.; Nagasawa, K.; Badhon, M.A.; et al. Global Wheat Head Detection (GWHD) dataset: A large and diverse dataset of high-resolution RGB-labelled images to develop and benchmark wheat head detection methods. *Plant Phenom.* **2020**, *2020*, 3521852. [CrossRef] [PubMed]
9. Xu, X.; Li, H.; Yin, F.; Xi, L.; Qiao, H.; Ma, Z.; Shen, S.; Jiang, B.; Ma, X. Wheat ear counting using K-means clustering segmentation and convolutional neural network. *Plant Methods* **2020**, *16*, 1106. [CrossRef] [PubMed]
10. Wang, D.; Zhang, D.; Yang, G.; Xu, B.; Luo, Y.; Yang, X. SSRNet: In-field counting wheat ears using multi-stage convolutional neural network. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 4403311. [CrossRef]
11. Khaki, S.; Safaei, N.; Pham, H.; Wang, L. WheatNet: A lightweight convolutional neural network for high-throughput image-based wheat head detection and counting. *Neurocomputing* **2022**, *489*, 78–89. [CrossRef]
12. Zaji, A.; Liu, Z.; Xiao, G.; Bhowmik, P.; Sangha, J.S.; Ruan, Y. AutoOLA: Automatic object level augmentation for wheat spikes counting. *Comput. Electron. Agric.* **2023**, *205*, 107623. [CrossRef]
13. Wang, D.; He, D. Channel pruned YOLO V5s-based deep learning approach for rapid and accurate apple fruitlet detection before fruit thinning. *Biosyst. Eng.* **2021**, *210*, 271–281. [CrossRef]
14. Sozzi, M.; Cantalamessa, S.; Cogato, A.; Kayad, A.; Marinello, F. Automatic bunch detection in white grape varieties using YOLOv3, YOLOv4, and YOLOv5 deep learning algorithms. *Agronomy* **2022**, *12*, 319. [CrossRef]
15. He, M.X.; Hao, P.; Xin, Y.Z. A robust method for wheat ear detection using UAV in natural scenes. *IEEE Access* **2020**, *8*, 189043–189053. [CrossRef]
16. Gong, B.; Ergu, D.; Cai, Y.; Ma, B. Real-time detection for wheat head applying deep neural network. *Sensors* **2020**, *21*, 191. [CrossRef]
17. Zhao, J.; Zhang, X.; Yan, J.; Qiu, X.; Yao, X.; Tian, Y.; Zhu, Y.; Cao, W. A wheat spike detection method in UAV images based on improved YOLOv5. *Remote Sens.* **2021**, *13*, 3095. [CrossRef]
18. Meng, X.; Li, C.; Li, J.; Li, X.; Guo, F.; Xiao, Z. YOLOv7-MA: Improved YOLOv7-Based Wheat Head Detection and Counting. *Remote Sens.* **2023**, *15*, 3770. [CrossRef]
19. David, E.; Serouart, M.; Smith, D.; Madec, S.; Velumani, K.; Liu, S.; Wang, X.; Espinosa, F.P.; Shafiee, S.; Tahir, I.S.; et al. Global wheat head dataset 2021: More diversity to improve the benchmarking of wheat head localization methods. *arXiv* **2021**, arXiv:2105.07660.
20. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In *Proceedings of the Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014; Proceedings, Part V 13*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 740–755.
21. Patricio, D.I.; Rieder, R. Computer vision and artificial intelligence in precision agriculture for grain crops: A systematic review. *Comput. Electron. Agric.* **2018**, *153*, 69–81. [CrossRef]

22. Oliver, A.; Odena, A.; Raffel, C.A.; Cubuk, E.D.; Goodfellow, I. Realistic evaluation of deep semi-supervised learning algorithms. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 3239–3250.
23. Yang, X.; Song, Z.; King, I.; Xu, Z. A survey on deep semi-supervised learning. *IEEE Trans. Knowl. Data Eng.* **2022**, *35*, 8934–8954. [CrossRef]
24. Sohn, K.; Zhang, Z.; Li, C.L.; Zhang, H.; Lee, C.Y.; Pfister, T. A simple semi-supervised learning framework for object detection. *arXiv* **2020**, arXiv:2005.04757.
25. Liu, Y.C.; Ma, C.Y.; He, Z.; Kuo, C.W.; Chen, K.; Zhang, P.; Wu, B.; Kira, Z.; Vajda, P. Unbiased teacher for semi-supervised object detection. *arXiv* **2021**, arXiv:2102.09480.
26. Tarvainen, A.; Valpola, H. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 1195–1204.
27. Xu, M.; Zhang, Z.; Hu, H.; Wang, J.; Wang, L.; Wei, F.; Bai, X.; Liu, Z. End-to-end semi-supervised object detection with soft teacher. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 3060–3069.
28. Tang, Y.; Chen, W.; Luo, Y.; Zhang, Y. Humble teachers teach better students for semi-supervised object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 3132–3141.
29. Zhou, H.; Ge, Z.; Liu, S.; Mao, W.; Li, Z.; Yu, H.; Sun, J. Dense teacher: Dense pseudo-labels for semi-supervised object detection. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 35–50.
30. Liu, Y.C.; Ma, C.Y.; Kira, Z. Unbiased teacher v2: Semi-supervised object detection for anchor-free and anchor-based detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 9819–9828.
31. Chen, B.; Chen, W.; Yang, S.; Xuan, Y.; Song, J.; Xie, D.; Pu, S.; Song, M.; Zhuang, Y. Label matching semi-supervised object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 14381–14390.
32. Xu, B.; Chen, M.; Guan, W.; Hu, L. Efficient Teacher: Semi-Supervised Object Detection for YOLOv5. *arXiv* **2023**, arXiv:2302.07577.
33. Wang, X.; Yang, X.; Zhang, S.; Li, Y.; Feng, L.; Fang, S.; Lyu, C.; Chen, K.; Zhang, W. Consistent-Teacher: Towards Reducing Inconsistent Pseudo-Targets in Semi-Supervised Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 3240–3249.
34. Kerdegari, H.; Razaak, M.; Argyriou, V.; Remagnino, P. Smart monitoring of crops using generative adversarial networks. In Proceedings of the Computer Analysis of Images and Patterns: 18th International Conference, CAIP 2019, Salerno, Italy, 3–5 September 2019; Proceedings, Part I 18; Springer: Berlin/Heidelberg, Germany, 2019; pp. 554–563.
35. Jiang, H.; Zhang, C.; Qiao, Y.; Zhang, Z.; Zhang, W.; Song, C. CNN feature based graph convolutional network for weed and crop recognition in smart farming. *Comput. Electron. Agric.* **2020**, *174*, 105450. [CrossRef]
36. Shorewala, S.; Ashfaq, A.; Sidharth, R.; Verma, U. Weed density and distribution estimation for precision agriculture using semi-supervised learning. *IEEE Access* **2021**, *9*, 27971–27986. [CrossRef]
37. Menezes, G.K.; Astolfi, G.; Martins, J.A.C.; Tetila, E.C.; Junior, A.d.S.O.; Gonçalves, D.N.; Junior, J.M.; Silva, J.A.; Li, J.; Gonçalves, W.N.; et al. Pseudo-label semi-supervised learning for soybean monitoring. *Smart Agric. Technol.* **2023**, *4*, 100216. [CrossRef]
38. Liu, T.; Jin, X.; Zhang, L.; Wang, J.; Chen, Y.; Hu, C.; Yu, J. Semi-supervised learning and attention mechanism for weed detection in wheat. *Crop Prot.* **2023**, *174*, 106389. [CrossRef]
39. Benchallal, F.; Hafiane, A.; Ragot, N.; Canals, R. ConvNeXt based semi-supervised approach with consistency regularization for weeds classification. *Expert Syst. Appl.* **2024**, *239*, 122222. [CrossRef]
40. Khaki, S.; Pham, H.; Han, Y.; Kuhl, A.; Kent, W.; Wang, L. Deepcorn: A semi-supervised deep learning method for high-throughput image-based corn kernel counting and yield estimation. *Knowl.-Based Syst.* **2021**, *218*, 106874. [CrossRef]
41. Casado-García, A.; Heras, J.; Milella, A.; Marani, R. Semi-supervised deep learning and low-cost cameras for the semantic segmentation of natural images in viticulture. *Precis. Agric.* **2022**, *23*, 2001–2026. [CrossRef]
42. Xu, X.; Wang, L.; Liang, X.; Zhou, L.; Chen, Y.; Feng, P.; Yu, H.; Ma, Y. Maize Seedling Leave Counting Based on Semi-Supervised Learning and UAV RGB Images. *Sustainability* **2023**, *15*, 9583. [CrossRef]
43. Johanson, R.; Wilms, C.; Johannsen, O.; Frintrop, S. S3AD: Semi-Supervised Small Apple Detection in Orchard Environments. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 4–8 January 2024; pp. 7076–7085.
44. Fourati, F.; Mseddi, W.S.; Attia, R. Wheat head detection using deep, semi-supervised and ensemble learning. *Can. J. Remote Sens.* **2021**, *47*, 198–208. [CrossRef]
45. Tian, Z.; Shen, C.; Chen, H.; He, T. Fcos: Fully convolutional one-stage object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, South Korea, 27 October–2 November 2019; pp. 9627–9636.
46. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
47. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.

48. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 91–99. [CrossRef] [PubMed]
49. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
50. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part I 14; Springer: Berlin/Heidelberg, Germany, 2016; pp. 21–37.
51. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
52. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
53. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
54. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
55. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. Yolox: Exceeding yolo series in 2021. *arXiv* **2021**, arXiv:2107.08430.
56. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; et al. YOLOv6: A single-stage object detection framework for industrial applications. *arXiv* **2022**, arXiv:2209.02976.
57. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 7464–7475.
58. Pound, M.P.; Atkinson, J.A.; Wells, D.M.; Pridmore, T.P.; French, A.P. Deep learning for multi-task plant phenotyping. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Venice, Italy, 22–29 October 2017; pp. 2055–2063.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.



Article

Apple Varieties Classification Using Deep Features and Machine Learning

Alper Taner ^{1,*}, Mahtem Teweldemedhin Mengstu ^{2,†}, Kemal Çağatay Selvi ¹, Hüseyin Duran ¹, İbrahim Gür ³ and Nicoleta Ungureanu ^{4,*}

¹ Department of Agricultural Machinery and Technologies Engineering, Faculty of Agriculture, Ondokuz Mayıs University, 55139 Samsun, Turkey; kselvi@omu.edu.tr (K.Ç.S.); huseyin.duran@omu.edu.tr (H.D.)

² Department of Agricultural Engineering, Hamelmalo Agricultural College, Keren P.O. Box 397, Eritrea; mahtem23@gmail.com

³ Fruit Research Institute, 32500 Isparta, Turkey; igur03@gmail.com

⁴ Department of Biotechnical Systems, Faculty of Biotechnical Systems Engineering, National University of Science and Technology Politehnica Bucharest, 060042 Bucharest, Romania

* Correspondence: alper.taner@omu.edu.tr (A.T.); nicoleta.ungureanu@upb.ro (N.U.)

† These authors contributed equally to this work.

Abstract: Having the advantages of speed, suitability and high accuracy, computer vision has been effectively utilized as a non-destructive approach to automatically recognize and classify fruits and vegetables, to meet the increased demand for food quality-sensing devices. Primarily, this study focused on classifying apple varieties using machine learning techniques. Firstly, to discern how different convolutional neural network (CNN) architectures handle different apple varieties, transfer learning approaches, using popular seven CNN architectures (VGG16, VGG19, InceptionV3, MobileNet, Xception, ResNet150V2 and DenseNet201), were adopted, taking advantage of the pre-trained models, and it was found that DenseNet201 had the highest (97.48%) classification accuracy. Secondly, using the DenseNet201, deep features were extracted and traditional Machine Learning (ML) models: support vector machine (SVM), multi-layer perceptron (MLP), random forest classifier (RFC) and K-nearest neighbor (KNN) were trained. It was observed that the classification accuracies were significantly improved and the best classification performance of 98.28% was obtained using SVM algorithms. Finally, the effect of dimensionality reduction in classification performance, deep features, principal component analysis (PCA) and ML models was investigated. MLP achieved an accuracy of 99.77%, outperforming SVM (99.08%), RFC (99.54%) and KNN (91.63%). Based on the performance measurement values obtained, our study achieved success in classifying apple varieties. Further investigation is needed to broaden the scope and usability of this technique, for an increased number of varieties, by increasing the size of the training data and the number of apple varieties.

Keywords: transfer learning; deep features; principal component analysis; machine learning; apple

Citation: Taner, A.; Mengstu, M.T.; Selvi, K.Ç.; Duran, H.; Gür, İ.; Ungureanu, N. Apple Varieties Classification Using Deep Features and Machine Learning. *Agriculture* **2024**, *14*, 252. <https://doi.org/10.3390/agriculture14020252>

Academic Editors: Francesco Marinello, Gniewko Niedbala, Sebastian Kujawa, Magdalena Piekutowska and Tomasz Wojciechowski

Received: 22 December 2023

Revised: 28 January 2024

Accepted: 1 February 2024

Published: 3 February 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Apples are among the most consumed and produced temperate fruit crops in the world; in 2022, over 95 million tons of apples were produced [1]. Due to their taste and nutritional value, there is an ever-increasing demand for apples that requires continuous production and supply. In this context, the implementation of innovative solutions in the systems of agricultural production and marketing is much required [2].

Identifying and sorting apples has several limitations because of its subjective nature, and it is a relatively complex problem due to the impressive number of apple varieties [3]. Human perception can be influenced when assessing the texture, color pattern, smell and other characteristics of apples. Likewise, manual inspections are highly dependent on the experience, training and duration of work of the personnel, as well as the environmental and psychological conditions, which can cause inconsistencies and variations in the results

or processing time [4,5]. However, automated recognition and classification systems can play a vital role in reducing labor costs and enhancing the economic efficiency of fruit, right from their harvest to the market [6]. Having the advantages of speed, suitability and high accuracy, computer vision has been effectively utilized as a non-destructive approach to automatically recognize and classify fruits and vegetables to meet the increased demand for food quality-sensing devices.

In the field of computer vision, image classification is a widely studied topic. In traditional machine learning, extracting features before training models on these features is required. Thus, the quality of the extracted features has a significant impact on a given classifier [7]. Deep learning has gained much popularity in image recognition and classification tasks of fruits and vegetables, as computing power and algorithms to process big data are emerging [8]. Because the convolutional neural network (CNN) model serves to automatically extract and classify features [9], it has been effectively utilized as a non-destructive approach to automatically classify fruits and vegetables to meet the increased demand for food quality-sensing devices [10], avoiding the need to manually or separately extract image features or representations [9].

In studies [11–15], convolutional neural networks (CNNs), which are deep learning-based, have shown excellent outcomes in a wide range of food and agricultural tasks, namely grading and sorting, varieties classification and disease detection. In grading bio-colored apples, a study [16] employed CNN, using multispectral images, to ensure the quality grading of apples. Likewise, CNN was also used to differentiate mature apples from immature ones in apple trees [17]. In fruit sorting, CNN models were applied to detect defective apples [18], while a similar work identified bruised apples in the investigation of automated sorting, by fusing deep features [19]. A great performance of the suppression mask R-CNN was reported in the classification of Gala and Blondee apple varieties [20].

Integrating CNNs and a convolution autoencoder, the authors of study [21] classified 26 different fruits, out of which nine classes were apples. In a similar classification problem, a CNN model was trained utilizing 30 types of leaf images from various growth periods [22]. Recently, the successful application of transfer learning to identify and classify 13 apple varieties using publicly available image datasets was reported [23].

The application of transfer learning, using models pre-trained on images from the internet, in diverse tasks such as the classification of crops and fruits, has been increasingly applicable and effective. As a result, applying a pre-trained network to learn new patterns with new data is beneficial. Furthermore, it is helpful when there are relatively small data to train a model. Thus, employing a pre-trained model is a typical solution [24]. This study firstly aims to test the performance of transfer learning in apple varieties and to investigate the impact of principal components (PC) of deep features coupled with traditional machine learning models. To the best of our knowledge, the integration of PC, deep features and machine learning has not been tested in apple varieties classification. Accordingly, this study aims to assess how these diverse components can jointly interact to enhance the accuracy and efficacy of apple variety classification.

The objectives of this study are: (I) to apply the transfer learning approach to develop an apple varieties classifier using pre-trained popular CNN architectures; (ii) to train and evaluate machine learning (ML) models using deep features obtained using best-performing models from the transfer learning approach; and (iii) to assess the effect of principal component analysis (PCA) on the performance of ML models trained using deep features.

2. Materials and Methods

2.1. Image Data

Ten apple varieties obtained from the Ministry of Agriculture and Fruit Research Institute of the Republic of Turkey were used in this study (Figure 1). The images were captured with a 20-megapixel resolution camera from a uniform distance and light set-up.

A total of 5808 images were captured from three views. Of these images, 70% were used in training, 15% in testing and 15% in validation, respectively.

		CNN Models						
Varieties		VGG16	VGG19	InceptionV3	Xception	Mobile Net	ResNet 150V2	DenseNet 201
FUJI								
GOLDEN REINDERS								
GRANNY SMITH								
KASEL 37								
MONDIAL GALA								
RED BREABURN								
RED CHIEF								
SCARLET SPUR								
STAR KRIMSON								
STARSPUR G. DELICIOUS								

Figure 1. Activations of the first layer of all models.

2.2. Transfer Learning

Generally, the training of CNNs is performed using large amount of data. Their deeper and interconnected layers are the reason for their good performance. Nonetheless, CNNs are often trained on a small dataset, which can simply result in the overfitting of the networks, leading to poor and unconvincing performances [23]. Likewise, training deep CNNs contributes to significant weaknesses, namely high processing costs and slow-running processes [25]. Thus, to solve the aforementioned difficulties, transfer learning is widely adopted. Transfer learning is a deep learning approach that utilizes the information acquired from a well-established model to reinstate with a new problem [26]. Because it

transfers the relevant knowledge from the learned model to the new task, transfer learning reduces the training time [27].

Thus, in this study, a transfer learning approach was adopted; seven popular architectures (VGG16, VGG19, InceptionV3, MobileNet, Xception, ResNet150V2 and DenseNet201) were trained using the Keras library in Google Colab, keeping the weights of the convolutional base and modifying only the final output layer, according to the number of classes in our study. This method utilizes the knowledge gained by these architectures from previously learned tasks, enhancing the model's performance on the target task. The description of each CNN architecture is as follows: VGG16 comprises thirteen convolutional layers and five Maxpooling layers; and VGG19 has sixteen convolutional layers and five Maxpooling layers. The VGG networks are typically designed successively [28]. From the Inception family, the InceptionV3 and Xception networks were adopted, which are characterized by multiple parallel convolutional operations, known as inception modules [29], with 48 convolutional layers and 71 layers, respectively. MobileNet is known for its depthwise separable convolution, having 53 layers [30]. From networks characterized by residual connections, ResNet150V2, with 150 layers [31], and DenseNet201, with dense connectivity between layers [32], with 201 layers, were used. In all the above models, the convolutional base was adopted excluding the top layer. According to our task, one hidden layer of 256 neurons with ReLu activation and a final layer of 10 represents the number of the apple classes or the final model prediction.

To discern how different architectures handle different apple varieties, the activations of the first layer of each model used have been provided in Figure 1. The activations represent the availability of specific features in the input data that serve as building blocks for successive layers to detect more complex features. Every model was trained for the same training, validation and test proportions. During the training, the hyperparameters were fixed to 100 batch size, 100 patience and 200 epochs. Early stopping and model checkpoint callbacks were also adopted to stop the training and save the weights of the best model, while monitoring the validation loss for a given training argument. This strategy facilitates the training, as it stops the training before all specified epochs are undertaken when the validation loss stops to decrease for a considerable amount of time.

2.3. Deep Features

Deep features are a set of high-level representations of input data, such as images or text, that are extracted from a deep neural network (DNN) such as MobileNet and DenseNet that is pre-trained on a large dataset. DNN is typically a multi-layered neural network that learns to identify patterns and features in the input data by successively transforming it through multiple layers of non-linear functions. The final output of the DNN, which is often a classification decision, is produced by a classifier that is trained on the deep features extracted from the input data. Deep features are more robust and informative than traditional hand-crafted features, as they capture more complex and nuanced relationships between the input data and the output labels [33].

Deep features are useful in a variety of machine learning tasks, such as image classification, object detection and natural language processing. In this study, after the transfer learning strategy was implemented, we took the best-performing model, which was DenseNet201, extracted 1920 deep features and used them to train traditional machine learning models (Figure 2).

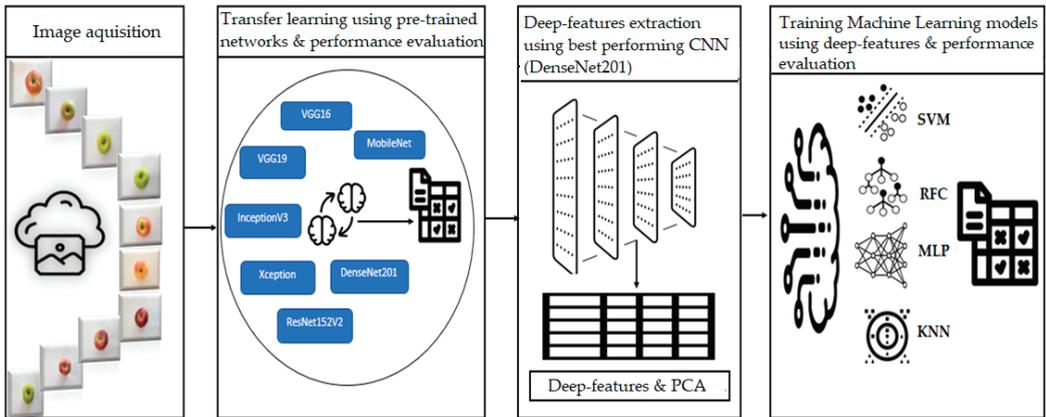


Figure 2. Diagram representation of the proposed approach.

2.4. Machine Learning Models

To train ML models, the 1920 deep features extracted using the best-performing CNN architecture, DenseNet201, were used. Four ML models, namely support vector machine (SVM), random forest classifier (RFC), multi-layer perceptron (MLP) and K-nearest neighbor (KNN) were trained with 10-fold stratified cross-validation (Skfold), as shown in Figure 3.

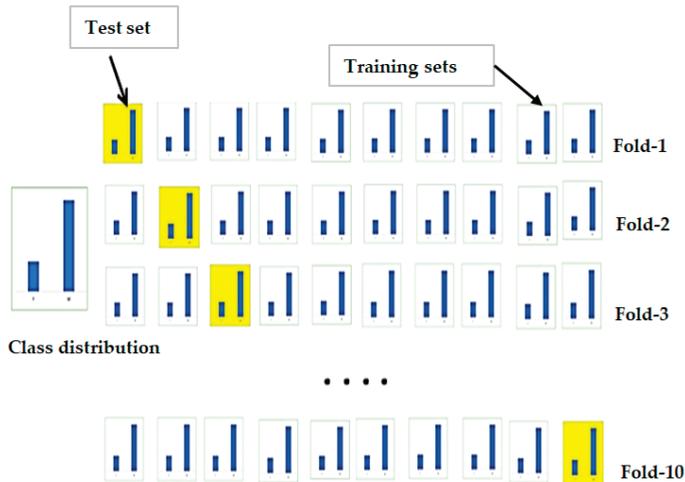


Figure 3. Tenfold stratified cross validation.

As depicted in Figure 3, the utilization of Skfold becomes vital when dealing with imbalanced data distributions among classes. In conditions where the frequency of different classes varies significantly, only K-fold cross-validation may result in unequal representations of classes across folds. Skfold, however, addresses this case by ensuring that each fold maintains a proportional representation of the various classes available in the dataset. This approach is mainly helpful in machine learning task where maintaining the balance of class distribution is decisive for model training. By stratifying the folds based on class labels, Skfold improves the robustness of the model evaluation, avoiding biased performance metrics that could appear from uneven class representation in traditional cross-validation. This approach promotes a more reliable assessment of the model’s generalization capabilities across diverse class distributions, ultimately contributing to a more accurate and

unbiased evaluation of the model's performance. To fine-tune the models, a grid search was implemented and final models were trained using the best parameters.

2.5. Principal Component Analysis

To investigate the effect of dimensionality reduction in classification performance, all ML models were also trained with deep features after implementing principal component analysis (PCA). PCA is a method that takes multi-dimensional data and gives it components by using the dependencies between the variables representing it in a more manageable and lower-dimensional form, without losing too much information. The fundamental idea of PCA is to minimize the dimensionality of a data set that is composed of a large number of interrelated variables, while retaining the variation present in the data to a feasible extent [34].

PCA is a linear transformation of data that minimizes the redundancy measured through covariance and maximizes the information that is measured through variance. PCA diminishes the number of given variables by reducing the last principal components that do not significantly contribute to the observed variability [35].

In the analysis, new elements, known as principal components and ranked by their Eigen values, are created. Principal components (PC) are new variables with two properties: firstly, each PC is a linear combination of the original variables; and secondly, PCs are uncorrelated to each other and the redundant information is removed [36].

As a multivariate unsupervised statistical procedure, PCA is widely used as a data exploratory tool in conditions that require feature selection such as data compression, image analysis, visualization, pattern recognition, regression and time series prediction. In our case, using the Python PCA library, we performed a PCA for the 1920 deep features. After examining the eigenvalues which told us the amount of variance explained by a single component, 262 features with eigenvalues greater than one were selected to train SVM, RFC, MLP and KNN models.

2.6. Performance Evaluation

To evaluate the performance of all models' accuracy, precision, recall, specificity, F1-score, Cohen's kappa, Matthews correlation coefficient, area under the receiver operating characteristic curve (AUC-ROC) the trade-off between true positive rate and false positive rate, area under the precision-recall curve (AUC-PR) and the trade-off between precision and recall for different classification thresholds, performance metrics were used in this study. The equations of all metrics are provided below, where TP: true positive, TN: true negative, FP: false positive, FN: false negative, Po: relative observed agreement among raters and Pe: relative observed agreement among raters.

1. Accuracy (Acc) is the ratio of the number of correctly classified samples to the total number of samples:

$$\text{Acc} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (1)$$

2. Precision (Pre) is the proportion of true positives out of total predicted positives, also known as positive predicted value:

$$\text{Pre} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (2)$$

3. Recall (Rec) is the proportion of positive samples classified as true. Recall is referred to as a true positive rate:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (3)$$

4. Specificity (Spec) is the proportion of negative samples classified as true. Known as a true negative rate:

$$\text{Spec} = \frac{\text{TN}}{\text{TN} + \text{FP}} \quad (4)$$

5. F1-Score (FS) is the harmonic mean of recall and precision:

$$\text{FS} = 2 \cdot \frac{\text{Pre} \cdot \text{Recall}}{\text{Pre} + \text{Recall}} \quad (5)$$

6. Cohen's kappa (K) is a measure of inter-rater agreement that considers the agreement that would be expected by chance:

$$\text{K} = \frac{\text{Po} - \text{Pe}}{1 - \text{Pe}} \quad (6)$$

7. Matthews correlation coefficient (MCC) is the correlation between the predicted and actual classifications:

$$\text{MCC} = \frac{[(\text{TP} \cdot \text{TN}) - (\text{FP} \cdot \text{FN})]}{\sqrt{[(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})]}} \quad (7)$$

3. Results

In this study, three methods were proposed: (i) the adoption of transfer learning using the popular CNN model; (ii) extracting deep features and training traditional ML models making use of the best performing CNN model, which was the DenseNet201; (iii) applying PCA to the deep features and training ML models using the PCs with eigenvalues greater than one. The performance metrics of each proposed method are presented in Tables 1–3, respectively.

Table 1. Performance of popular CNN models using Transfer Learning.

CNN Model	Performance Metrics								
	Acc	Pre	Rec	Spec	FS	AUC -ROC	AUC -PR	K	MCC
VGG16	91.87	92.449	91.87	99.09	91.85	99.72	97.97	90.95	91.02
VGG19	94.16	94.55	94.16	99.35	94.18	99.76	98.38	93.49	93.54
InceptionV3	92.44	92.74	92.51	99.16	92.45	99.77	98.13	91.59	91.63
MobileNet	96.45	96.61	96.45	99.61	96.44	99.96	99.67	96.05	96.07
DenseNet201	97.48	97.54	97.48	99.72	97.48	99.98	99.86	97.19	97.20
Xception	93.47	93.97	93.47	99.27	93.39	99.79	98.69	92.73	92.80
ResNet152V2	94.84	95.26	94.84	99.42	94.76	99.89	99.17	94.26	94.32

As depicted in Table 1, in the results of performance metrics of the seven pre-trained CNN models, DenseNet201 outperformed all other models with an accuracy of 97.48% when classifying 10 apple varieties.

Similarly, the performance of the machine learning models, namely SVM, RFC, MLP and KNN, which were trained and tested using 1920 features obtained from DenseNet201 CNN (presented in Table 2), showed that SVM obtained an accuracy of 98.28%, outperforming all other models. However, as PCA was applied and features were reduced to 262, MLP outperformed all models with a classification accuracy of 99.77% (presented in Table 3).

Table 2. Performance metrics of ML models.

Model	Performance Metrics								
	Acc	Pre	Rec	Spec	FS	AUC-ROC	AUC-PR	K	MCC
SVM	98.28	98.32	98.10	97.67	98.18	99.88	98.71	98.08	98.09
RFC	91.86	91.65	91.69	90.14	91.62	99.56	96.60	90.94	90.95
MLP	98.05	97.99	97.95	96.47	97.96	99.96	99.70	97.83	97.83
KNN	89.33	91.59	88.93	98.46	89.04	98.59	95.04	88.13	88.17

Table 3. Performance metrics of ML models trained with PCA.

Model	Performance Metrics								
	Acc	Pre	Rec	Spec	FS	AUC-ROC	AUC-PR	K	MCC
SVM	99.08	99.06	99.05	97.70	99.05	99.93	99.43	98.97	98.98
RFC	99.54	99.57	99.55	100	99.99	99.99	99.94	99.49	99.49
MLP	99.77	99.78	99.75	100	99.99	99.99	99.99	99.74	99.75
KNN	91.63	92.35	91.26	98.70	99.70	99.70	97.09	90.68	90.73

The confusion matrices containing the classification results of the test data of each model trained using 262 PCs are depicted in the figures below.

Applying the SVM model, a few misclassifications were observed in Fuji, Golden Reinders, Kasel37 and Mondial Gala apple varieties, as shown in Figure 4. The model scored an excellent classification performance of 99.08%.

SVM	Predicted										
Actual	FUJI	GOLDEN REINDERS	GRANNY SMITH	KASEL 37	MONDIAL GALA	RED BREABURN	RED CHIEF	SCARLET SPUR	STARKRIMSON	STARSPUR	G.DELICIOUS
FUJI	72	0	0	0	0	0	0	0	0	0	0
GOLDEN REINDERS	1	92	0	0	0	0	0	0	0	0	0
GRANNY SMITH	0	0	88	0	0	0	0	0	0	0	0
KASEL 37	0	0	0	79	0	0	0	0	0	0	0
MONDIAL GALA	0	0	0	0	72	0	0	0	0	0	0
RED BREABURN	0	0	0	1	1	85	0	0	0	0	0
RED CHIEF	0	0	0	0	0	0	91	1	0	0	0
SCARLET SPUR	0	0	0	0	0	0	0	96	0	0	0
STARKRIMSON	1	0	0	1	0	0	0	0	91	0	0
STARSPUR G.DELICIOUS	1	1	0	0	0	0	0	0	0	98	0

Figure 4. SVM confusion matrix.

Figure 5 presents the confusion matrix according to the MLP model. Out of the four ML models, MLP has the highest performance of 99.77%; out of 782 instances of given test data, only four misclassifications were observed.

MLP	Predicted									
Actual	FUJI	GOLDEN REINDERS	GRANNY SMITH	KASEL 37	MONDIAL GALA	RED BREABURN	RED CHIEF	SCARLET SPUR	STARKRIMSON	STARKSPUR G.DELICIOUS
FUJI	72	0	0	0	0	0	0	0	0	0
GOLDEN REINDERS	0	92	0	0	0	1	0	0	0	0
GRANNY SMITH	0	0	88	0	0	0	0	0	0	0
KASEL 37	0	0	0	79	0	0	0	0	0	0
MONDIAL GALA	0	0	0	0	72	0	0	0	0	0
RED BREABURN	0	0	0	0	0	87	0	0	0	0
RED CHIEF	0	0	0	0	0	0	91	1	0	0
SCARLET SPUR	0	0	0	0	0	0	1	95	0	0
STARKRIMSON	0	1	0	0	0	0	0	0	92	0
STARKSPUR G.DELICIOUS	0	0	0	0	0	0	0	0	0	100

Figure 5. MLP confusion matrix.

With the highest confusion between Kase137 and Red Braeburn, the RFC model performed with an overall classification accuracy of 99.54%, where eight misclassifications were observed (Figure 6).

RFC	Predicted									
Actual	FUJI	GOLDEN REINDERS	GRANNY SMITH	KASEL 37	MONDIAL GALA	RED BREABURN	RED CHIEF	SCARLET SPUR	STARKRIMSON	STARKSPUR G.DELICIOUS
FUJI	72	0	0	0	0	0	0	0	0	0
GOLDEN REINDERS	0	92	0	0	0	0	0	0	0	1
GRANNY SMITH	0	0	88	0	0	0	0	0	0	0
KASEL 37	0	0	0	76	0	3	0	0	0	0
MONDIAL GALA	0	0	0	0	72	0	0	0	0	0
RED BREABURN	0	0	0	0	0	86	0	0	0	1
RED CHIEF	0	0	0	1	1	0	90	0	0	0
SCARLET SPUR	0	0	0	0	0	0	0	95	1	0
STARKRIMSON	0	0	0	0	0	0	0	0	93	0
STARKSPUR G.DELICIOUS	0	1	0	0	0	0	0	0	0	99

Figure 6. RFC confusion matrix.

Among the four models, KNN appeared to have the lowest performance, with 90 misclassifications, which is very significant compared to the other three models. A total of 15 misclassifications were also observed between Fuji and Red Braeburn apple varieties, followed by seven misclassifications between Red Braeburn and Kasel37 varieties; or, out of the total misclassifications, 22 or 24% were wrongly classified as Red Braeburn, which shows that a close observation in the features of Red Braeburn variety is needed (Figure 7).

KNN		Predicted									
Actual	FUJI	GOLDEN REINDERS	GRANNY SMITH	KASEL 37	MONDIAL GALA	RED BRAEBURN	RED CHIEF	SCARLET SPUR	STARKRIMSON	STARKSPUR	G.DELICIOUS
FUJI	47	0	0	4	1	15	1	0	1	3	
GOLDEN REINDERS	0	77	2	1	0	0	0	0	0	13	
GRANNY SMITH	0	1	86	0	0	0	0	0	0	1	
KASEL 37	0	0	1	73	0	2	2	0	0	1	
MONDIAL GALA	1	0	0	4	63	4	0	0	0	0	
RED BRAEBURN	0	1	0	7	0	77	0	0	2	0	
RED CHIEF	0	0	0	4	0	0	82	6	0	0	
SCARLET SPUR	0	0	0	2	1	0	5	88	0	0	
STARKRIMSON	0	0	0	3	0	1	4	2	83	0	
STARKSPUR	0	2	0	0	0	0	0	0	0	98	

Figure 7. KNN confusion matrix.

4. Discussion

A deep learning-based convolutional neural network for apple classification was presented in a previous study [37]. The model used a network with four layers to classify unforeseen apple images, and the CNN model was trained and tested using images of 13 apple varieties. At the test phase, the model achieved an accuracy of 90%. In another study, by proposing an integrated CNN and certainty factor, a model with a dataset containing images of six apple varieties was trained, and the image classification performance scored an excellent result of 99.78% [38]. When classifying a group of 26 different fruits, which included nine apple classes, the authors of study [21] also obtained an accuracy of 95.67%, employing CNN and the autoencoder. In study [39], a shallow CNN was employed to classify six apple varieties; the authors collected and labeled apple images to train a model along with data augmentation. As the training and model parameter optimization were performed using the Caffe framework, at the test stage, the model accuracy performance was 92%. A CNN-based deep learning model was developed in study [22] to classify thirty apple varieties under complex natural environments, which contributed to the currently available means of apple variety classification; the model's accuracy was 93.14% using the test set.

Applying the ability of the CNN model to extract features, a method to identify 14 apple varieties was proposed by the authors of study [40]. Compared to transfer learning techniques, such as ResNet50, VGG16, MobileNet and EfficientNetB0, their approach attained an improved test accuracy of 99.59%.

Recently, in the successful application of transfer learning to identifying and classifying 13 apple varieties using publicly available image datasets with accuracies of 96–100%, the use of different models was reported [23]. The summary of recently related works, including our proposed model, has been provided in Table 4.

Table 4. Performance comparison of the proposed methods with related works.

Task	Model and Accuracy	References
Classification of 13 apple varieties	CNN, 90%	[37]
Classification of 6 apple varieties	CNN and certainty factor, 99.78%	[38]
Classification of 26 fruits	CNN and autoencoder, 95.67%	[21]
Classification of 6 apple varieties	Shallow CNN, 92%	[39]
Classification of 30 apple varieties	CNN, 93.14%	[22]
Classification of 14 apple varieties	Lightweight CNN, 99.59%	[40]
Classification of 14 apple varieties	CNN, 96.1–100%	[23]
Classification of 10 apple varieties	Transfer learning, 97.48%	Proposed Method 1
Classification of 10 apple varieties	Deep features and ML, 98.28%	Proposed Method 2
Classification of 10 apple varieties	Deep features, PCA and ML, 99.77%	Proposed Method 3

In this study, compared to relevant and related works, it can be concluded that the results of our proposed models are in agreement with the classification accuracy indicating the relevance of all the techniques, and especially the third method, which is the integration of deep features, PCA and ML, which performed as excellently as state-of-the-art models and even outperformed some of them, likely because of the use of PCA.

Additionally, in the research of apple varieties' classifications, to the best of our knowledge, the integration of deep features and PCA has not yet been adopted; therefore, the approach of coupling deep features, PCA and machine learning models in the area of apple varieties classification is an innovative approach that this study has investigated and demonstrated. Furthermore, the adoption of Skfold validation is a new addition to the literature that this investigation has undertaken.

According to the performance metrics (Table 1), DenseNet201 has the highest classification accuracy (97.48%) among popular CNN models applied in the transfer learning approach, indicating its potential to be applied in apple varieties classification. The investigation into the integration of deep features and traditional ML models shows the benefits and advantages of coupling deep features and ML features to have improved classification accuracy; SVM scored 98.28%, but ultimately, the incorporation of PCA increased the model's performance up to 99.77%, as achieved by MLP model. Additionally, observing other performance metrics, MLP has the highest values—a precision of 99.78%, a recall of 99.75% and an F1 score is 99.76%. Besides, its discrimination power was observed in the results of AUC-ROC and AUC-PR, which were 99.99% and 99.99%, respectively. Moreover, Cohen's Kappa and MCC metrics, which examine the agreement between predicted and actual classes, are 99.75% and 99.74%, outperforming all the ML models, as shown in Table 3.

Thus, based on our proposed technique, integrating deep features, PCA and the ML models MLP and SVM can classify ten apple varieties with excellent performance.

5. Conclusions

In solving agricultural problems such as fruit varieties' classification and grading, the application of machine learning has a significant role. As indicated in our work, and also in other recent and related works, transfer learning is commonly used in classifying different fruit types, and specifically apple varieties. Our work trained and tested seven popular CNN architectures from different families, which all of them performed well, with above

90% classification accuracy. To further and discover a new dimension in apple varieties' classification, our investigation focused on the application of deep features coupled with PCA and traditional ML algorithms, for which the Skfold stratified cross-validation was adopted during the training.

All four models, namely SVM, RFC, MLP and KNN, were tested with separate test data and achieved classification accuracies of 99.08%, 99.54%, 99.77% and 91.63%, respectively. The study confirmed the increase in performance by coupling deep features and PCA for the given image dataset. This study was undertaken on ten apple varieties; however, to prove the efficiency of the proposed technique, increasing the size of the training data and increasing the number of apple varieties should be further studied and examined. Besides, to develop robust and versatile models, we suggest the use of random images, containing scattered apple images taken in real-time, such as on conveyors and other production environments. In future investigations, images captured in various acquisition setups and light conditions that represent actual and complicated environments should be considered. One of the challenges in apple varieties is the occurrence of variability within classes, which contributes to confusion and misclassification. Although it is difficult to provide conclusive reasons based on experiments, the misclassifications indicated in the results might be due to variability within classes. Hence, future research endeavors should explore and identify algorithms capable of handling such complexities. Prioritizing the discovery and implementation of algorithms that are robust to variations within classes will enhance the model's ability to accurately discriminate between intricate and closely related instances, contributing to an improved classification performance in challenging scenarios. On top of the varieties, the difference or similarities of agricultural products such as fruits is highly impacted by the stage of ripening. Two different varieties might be highly similar and highly different at various stages of their ripening, which significantly contribute to the challenges in the development of sorting, grading or classification models. Therefore, during data collection, such things need to be addressed and considered. Moreover, apart from classification models, the use of object detection algorithms, including prominent ones like YOLO (You Only Look Once), needs to be researched, to further strengthen and enhance the practical and real-time application of models. Object detection goes beyond classifying entire images, allowing for the precise identification and localization of multiple objects within an image.

Author Contributions: Conceptualization, A.T. and M.T.M.; methodology, A.T.; software, M.T.M.; validation, K.Ç.S., A.T. and H.D.; formal analysis, A.T. and M.T.M.; investigation, A.T.; resources, İ.G.; data curation, A.T.; writing—original draft preparation, A.T., N.U. and H.D.; writing—review and editing, N.U. and İ.G.; visualization, N.U. and M.T.M.; supervision, A.T., K.Ç.S. and H.D.; project administration, A.T.; funding acquisition, N.U. All authors have read and agreed to the published version of the manuscript.

Funding: The APC was funded by National University of Science and Technology Politehnica Bucharest, Romania, within the PubArt Program.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding authors.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Food and Agriculture Organization of the United Nations. *FAOSTAT Statistical Database*; FAO: Rome, Italy, 2022. Available online: <https://www.fao.org/faostat/en/#data/QCL/visualize> (accessed on 15 January 2024).
2. Rachmawati, E.; Supriana, I.; Khodra, M.L. Toward a new approach in fruit recognition using hybrid RGBD features and fruit hierarchy property. In Proceedings of the 4th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), Yogyakarta, Indonesia, 19–21 September 2017; pp. 1–6. [CrossRef]
3. Hameed, K.; Chai, D.; Rassau, A. A comprehensive review of fruit and vegetable classification techniques. *Image Vis Comput.* **2018**, *80*, 24–44. [CrossRef]

4. Barbelian, M.A.; Bălan, C.V. Fault tree event classification by neural network analysis. *UIPB Sci. Bull. Series D Mech. Eng.* **2017**, *79*, 55–66.
5. Bhargava, A.; Bansal, A.; Goyal, V. Machine learning-based detection and sorting of multiple vegetables and fruits. *Food Anal. Methods* **2022**, *15*, 228–242. [CrossRef]
6. Tian, H.; Wang, T.; Liu, Y.; Qiao, X.; Li, Y. Computer vision technology in agricultural automation—A review. *Inf Process Agric.* **2020**, *7*, 1–19. [CrossRef]
7. Taner, A.; Mengstu, M.T.; Selvi, K.Ç.; Duran, H.; Kabaş, Ö.; Gür, İ.; Karaköse, T.; Gheorghită, N.-E. Multiclass apple varieties classification using machine learning with histogram of oriented gradient and color moments. *Appl. Sci.* **2023**, *13*, 7682. [CrossRef]
8. Srivalli, D.S.; Geetha, A. Fruits, vegetable and plants category recognition systems using convolutional neural networks: A review. *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol.* **2019**, *5*, 452–462. [CrossRef]
9. Iosif, A.; Maican, E.; Biriş, S.; Popa, L. Automated quality assessment of apples using convolutional neural networks. *INMATEH -Agric. Eng.* **2023**, *71*, 483–498. [CrossRef]
10. Abasi, S.; Minaei, S.; Jamshidi, B.; Fathi, D. Dedicated non-destructive devices for food quality measurement: A review. *Trends Food Sci. Technol.* **2018**, *78*, 197–205. [CrossRef]
11. Gikunda, P.K.; Jouandeau, N. State-of-the-art convolutional neural networks for smart farms: A review. In Proceedings of the Intelligent Computing Conference, London, UK, 16–17 July 2019; pp. 763–775. [CrossRef]
12. Hamid, Y.; Wani, S.; Soomro, A.B.; Alwan, A.A.; Gulzar, Y. Smart Seed Classification System based on MobileNetV2 Architecture. In Proceedings of the 2nd International Conference on Computing and Information Technology (ICIT), Tabuk, Saudi Arabia, 25–27 January 2022; pp. 217–222. [CrossRef]
13. Kaur, P.; Harnal, S.; Tiwari, R.; Upadhyay, S.; Bhatia, S.; Mashat, A.; Alabdali, A.M. Recognition of leaf disease using hybrid convolutional neural network by applying feature reduction. *Sensors* **2022**, *22*, 575. [CrossRef]
14. Nagaraju, M.; Chawla, P.; Upadhyay, S.; Tiwari, R. Convolution network model based leaf disease detection using augmentation techniques. *Expert Syst.* **2022**, *39*, e12885. [CrossRef]
15. Nagaraju, M.; Chawla, P.; Tiwari, R. An effective image augmentation approach for maize crop disease recognition and classification. In *Computational Intelligence and Smart Communication*; ICCISC 2022. Communications in Computer and Information Science; Mehra, R., Meesad, P., Peddoju, S.K., Rai, D.S., Eds.; Springer: Cham, Switzerland, 2022; Volume 1672. [CrossRef]
16. Unay, D. Deep learning based automatic grading of bi-colored apples using multispectral images. *Multimed. Tools Appl.* **2022**, *81*, 38237–38252. [CrossRef]
17. Lu, S.; Chen, W.; Zhang, X.; Karkee, M. Canopy-attention-YOLOv4-based immature/mature apple fruit detection on dense-foliage tree architectures for early crop load estimation. *Comput. Electron. Agric.* **2022**, *193*, 106696. [CrossRef]
18. Fan, S.; Li, J.; Zhang, Y.; Tian, X.; Wang, Q.; He, X.; Zhang, C.; Huang, W. On line detection of defective apples using computer vision system combined with deep learning methods. *J. Food Eng.* **2020**, *286*, 110102. [CrossRef]
19. Hu, Z.; Tang, J.; Zhang, P.; Jiang, J. Deep learning for the identification of bruised apples by fusing 3D deep features for apple grading systems. *Mech. Syst. Signal Process.* **2020**, *145*, 106922. [CrossRef]
20. Chu, P.; Li, Z.; Lammers, K.; Lu, R.; Liu, X. Deep learning-based apple detection using a suppression mask R-CNN. *Pattern Recognit. Lett.* **2021**, *147*, 206–211. [CrossRef]
21. Xue, G.; Liu, S.; Ma, Y. A hybrid deep learning-based fruit classification using attention model and convolution autoencoder. *Complex Intell. Syst.* **2023**, *9*, 2209–2219. [CrossRef]
22. Chen, J.; Han, J.; Liu, C.; Wang, Y.; Shen, H.; Li, L. A deep-learning method for the classification of apple varieties via leaf images from different growth periods in natural environment. *Symmetry* **2022**, *14*, 1671. [CrossRef]
23. Yu, F.; Lu, T.; Xue, C. Deep learning-based intelligent apple variety classification system and model interpretability analysis. *Foods* **2023**, *12*, 885. [CrossRef]
24. Naranjo-Torres, J.; Mora, M.; Hernández-García, R.; Barrientos, R.J.; Fredes, C.; Valenzuela, A. A review of convolutional neural network applied to fruit image processing. *Appl. Sci.* **2020**, *10*, 3443. [CrossRef]
25. Canziani, A.; Paszke, A.; Culurciello, E. An analysis of deep neural network models for practical applications. *arXiv* **2016**, arXiv:1605.07678. [CrossRef]
26. Behera, S.K.; Rath, A.K.; Sethy, P.K. Maturity status classification of papaya fruits based on machine learning and transfer learning approach. *Inf. Process Agric.* **2020**, *8*, 244–250. [CrossRef]
27. Pan, S.J.; Yang, Q. A survey on transfer learning. *IEEE Trans Knowl. Data Eng.* **2010**, *22*, 1345–1359. [CrossRef]
28. Xiao, J.; Wang, J.; Cao, S.; Li, B. Application of a novel and improved VGG-19 network in the detection of workers wearing masks. *J. Phys. Conf. Ser.* **2020**, *1518*, 012041. [CrossRef]
29. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826. [CrossRef]
30. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv:1704.04861. [CrossRef]
31. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.

32. Huang, G.; Liu, Z.; van der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
33. Lee, S.H.; Chan, C.; Mayo, S.J.; Remagnino, P. How deep learning extracts and learns leaf features for plant classification. *Pattern Recognit.* **2017**, *71*, 1–13. [CrossRef]
34. Jolliffe, I.T. Principal Component Analysis. In *Springer Series in Statistics*; Springer: New York, NY, USA, 2002; ISBN 0-387-95442-2.
35. Ma, J.; Yuan, Y. Dimension reduction of image deep feature using PCA. *J. Vis. Commun. Image Represent.* **2019**, *63*, 102578. [CrossRef]
36. Khalid, S.; Khalil, T.; Nasreen, S. A survey of feature selection and feature extraction techniques in machine learning. In Proceedings of the Science and Information (SAI) Conference, London, UK, 27–29 August 2014; pp. 372–378. [CrossRef]
37. Al-Shawwa, M.; Abu-Naser, S.S. Knowledge based system for apple problems using CLIPS. *Int. J. Acad. Eng. Res.* **2019**, *3*, 1–11.
38. Katarzyna, R.; Paweł, M. A vision-based method utilizing deep convolutional neural networks for fruit variety classification in uncertainty conditions of retail sales. *Appl. Sci.* **2019**, *9*, 3971. [CrossRef]
39. Li, J.; Xie, S.; Chen, Z.; Liu, H.; Kang, J.; Fan, Z.; Li, W. A shallow convolutional neural network for apple classification. *IEEE Access* **2020**, *8*, 111683–111692. [CrossRef]
40. Shruthi, U.; Narmadha, K.S.; Meghana, E.; Meghana, D.N.; Lakana, K.P.; Bhuvan, M.P. Apple varieties classification using light weight CNN Model. In Proceedings of the 4th International Conference on Circuits, Control, Communication and Computing (I4C), Bangalore, India, 21–23 December 2022; pp. 68–72. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.



Article

Insights into Drought Tolerance of Tetraploid Wheat Genotypes in the Germination Stage Using Machine Learning Algorithms

Berk Benlioglu ¹, Fatih Demirel ², Aras Türkoğlu ^{3,*}, Kamil Haliloğlu ⁴, Hamdi Özaktan ⁵, Sebastian Kujawa ⁶, Magdalena Piekutowska ⁷, Tomasz Wojciechowski ⁶ and Gniewko Niedbala ^{6,*}

¹ Department of Field Crops, Faculty of Agriculture, Ankara University, 06110 Ankara, Türkiye; benlioglu@ankara.edu.tr

² Department of Agricultural Biotechnology, Faculty of Agriculture, Iğdır University, 76000 Iğdir, Türkiye; fatih.demirel@igdir.edu.tr

³ Department of Field Crops, Faculty of Agriculture, Necmettin Erbakan University, 42310 Konya, Türkiye

⁴ Department of Field Crops, Faculty of Agriculture, Atatürk University, 25240 Erzurum, Türkiye; kamilh@atauni.edu.tr

⁵ Department of Field Crops, Faculty of Agriculture, Erciyes University, 38000 Kayseri, Türkiye; ozaktan_03@hotmail.com

⁶ Department of Biosystems Engineering, Faculty of Environmental and Mechanical Engineering, Poznań University of Life Sciences, Wojska Polskiego 50, 60-627 Poznań, Poland; sebastian.kujawa@up.poznan.pl (S.K.); tomasz.wojciechowski@up.poznan.pl (T.W.)

⁷ Department of Botany and Nature Protection, Institute of Biology, Pomeranian University in Słupsk, 22b Arciszewskiego St., 76-200 Słupsk, Poland; magdalena.piekutowska@ups.edu.pl

* Correspondence: aras.turkoglu@erbakan.edu.tr (A.T.); gniewko.niedbala@up.poznan.pl (G.N.)

Abstract: Throughout germination, which represents the initial and crucial phase of the wheat life cycle, the plant is notably susceptible to the adverse effects of drought. The identification and selection of genotypes exhibiting heightened drought tolerance stand as pivotal strategies aimed at mitigating these effects. For the stated objective, this study sought to evaluate the responses of distinct wheat genotypes to diverse levels of drought stress encountered during the germination stage. The induction of drought stress was achieved using polyethylene glycol at varying concentrations, and the assessment was conducted through the application of multivariate analysis and machine learning algorithms. Statistical significance ($p < 0.01$) was observed in the differences among genotypes, stress levels, and their interaction. The ranking of genotypes based on tolerance indicators was evident through a principal component analysis and biplot graphs utilizing germination traits and stress tolerance indices. The drought responses of wheat genotypes were modeled using germination data. Predictions were then generated using four distinct machine learning techniques. An evaluation based on R-square, mean square error, and mean absolute deviation metrics indicated the superior performance of the elastic-net model in estimating germination speed, germination power, and water absorption capacity. Additionally, in assessing the criterion metrics, it was determined that the Gaussian processes classifier exhibited a better performance in estimating root length, while the extreme gradient boosting model demonstrated superior performance in estimating shoot length, fresh weight, and dry weight. The study's findings underscore that drought tolerance, susceptibility levels, and parameter estimation for durum wheat and similar plants can be reliably and efficiently determined through the applied methods and analyses, offering a fast and cost-effective approach.

Citation: Benlioglu, B.; Demirel, F.; Türkoğlu, A.; Haliloğlu, K.; Özaktan, H.; Kujawa, S.; Piekutowska, M.; Wojciechowski, T.; Niedbala, G. Insights into Drought Tolerance of Tetraploid Wheat Genotypes in the Germination Stage Using Machine Learning Algorithms. *Agriculture* **2024**, *14*, 206. <https://doi.org/10.3390/agriculture14020206>

Academic Editors: Valya Vassileva and Hyeon Tae Kim

Received: 4 December 2023

Revised: 16 January 2024

Accepted: 20 January 2024

Published: 27 January 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: tetraploid wheat; drought stress; germination; stress tolerance; modeling

1. Introduction

Wheat holds the distinction of being the primary staple food [1]. Tetraploid wheat is a species that displays heightened vulnerability to abiotic pressures, particularly showing susceptibility to drought [2]. This species exhibits a discerning preference for specific climatic and soil conditions to attain optimal yield and quality and is more responsive to

environmental challenges compared to bread wheat [3]. In 2021, durum wheat production in Türkiye decreased by around 21% compared to the previous year [4].

The anticipated climate change and global warming are expected to exacerbate the magnitude of stressors [5]. Plants commonly face a combination of biotic and abiotic stressors within their native environment [6]. However, among all the stressors influenced by climate change, drought emerges as the primary factor hindering plant productivity [7]. Climate change may have varying impacts on crop performance depending on the timing and duration of drought events, as well as whether drought stress occurs alone or in combination with heat stress. In the Mediterranean region, low rainfall and irregularities in the rainfall regime can cause significant yield losses for crops grown under rainfed conditions, such as durum wheat [8]. Drought stress during germination can have severe consequences for the success of the plant's life cycle, as an inadequate water supply at this stage can hinder the robust growth of roots and shoots, resulting in significant crop losses [9,10].

The global population is undergoing a significant and rapid growth trajectory, with projections estimating that it will reach approximately 9.74 billion individuals by the year 2050 [11]. Simultaneously, there will be an escalating demand for sustenance. To meet the nutritional requirements, it is crucial to cultivate new cultivars that demonstrate high productivity and resilience to both biotic and abiotic challenges [12]. To facilitate the development of novel drought-tolerant cultivars, it is essential to ascertain the tolerance status of existing genotypes [13].

According to Rai et al. [14], obtaining accurate and dependable outcomes may be achieved by performing selection during the first development phase in controlled laboratory settings. In laboratory settings, NaCl, polyethylene glycol (PEG), sorbitol, and mannitol are often used to induce drought stress in plants [15]. This manipulation serves to augment the dry conditions within the plant's growth environment, hence impeding water uptake by the plant [16]. High molecular weight polyethylene glycols (PEGs) are often used as stress agents in various studies [17]. This is mostly attributed to their water-soluble nature, lack of toxicity, and inability to be absorbed by plant roots [18].

The examination of yield components, the assessment of yield stability, and the enhancement of stress tolerance represent conventional methodologies in plant breeding [19]. It is paramount to establish a clear and comprehensive understanding of the correlation between improved agricultural features and their reciprocal impacts [20]. In controlled breeding research, the effects of input elements (genotype and treatment factors) on observed plant characteristics (outputs) have been extensively investigated [21,22]. Traditional statistical approaches have been predominantly utilized for the examination and interpretation of outcome variables [23,24]. Analysis of variance, principal component analysis (PCA), and linear regression models are commonly employed techniques to determine the associations between independent input factors and dependent output variables [25].

Machine learning (ML) algorithms are increasingly being employed in various aspects of plant science and agriculture. In addition to these effective approaches, it is noteworthy that ML algorithms, as a subfield of artificial intelligence, possess the capacity to make precise predictions and enhance the efficacy of various intricate biological systems. Frequently employed for acquiring knowledge and constructing models optimized for specific tasks, ML algorithms undergo a learning process from data. Their objective is to provide predictions for a designated target variable using knowledge acquired from the data's properties [20,26,27]. Presently, the determination of drought tolerance, susceptibility levels, and the estimation of observed parameters in plants can be reliably accomplished using ML systems in the fastest, most cost-effective, and practical manner through the applied methods and analyses.

The objective of this study was to investigate common tetraploid wheat genotypes in Türkiye to determine their drought tolerance at the initial growth stage using a multivariate analysis and stress tolerance indices. Furthermore, the study aimed to employ ML algorithms to predict the parameters observed during the early development phase

of tetraploid wheat under drought conditions and to present a range of available ML models. Consequently, this research sought to unveil the drought tolerance of varieties and characterize genotypes suitable as parental plants for future breeding-focused studies.

2. Materials and Methods

2.1. Plant Materials

This research was conducted at Ankara University, Faculty of Agriculture, Department of Field Crops. The study utilized eleven varieties extensively employed in durum wheat cultivation in Türkiye, along with one hulled tetraploid wheat genotype, emmer (*Triticum dicoccum*), sourced from the Kars province of Türkiye (Table 1). The selected genotypes in our study have been registered in arid and semi-arid regions of Türkiye.

Table 1. Tetraploid wheat genotypes used in the research.

Genotype	Type	Registration Year	Growth Habit	Breeding Company
Altın-40/98	Cultivar	1998	Alternative	Field Crops Central Research Institute, Ankara, Türkiye
Artuklu	Cultivar	2008	Spring	GAP International Agricultural Research and Training Center, Diyarbakir, Türkiye
Çakmak-79	Cultivar	1979	Alternative	Field Crops Central Research Institute, Ankara, Türkiye
Çeşit-1252	Cultivar	1999	Alternative	Field Crops Central Research Institute, Ankara, Türkiye
Eminbey	Cultivar	2009	Winter	Field Crops Central Research Institute, Ankara, Türkiye
Kızıltan-91	Cultivar	1991	Alternative	Field Crops Central Research Institute, Ankara, Türkiye
Kunduru-1149	Cultivar	1967	Winter	Field Crops Central Research Institute, Ankara, Türkiye
Meram-2002	Cultivar	2002	Alternative	Bahri Dagdas International Agricultural Research Institute, Konya, Türkiye
Mirzabey-2000	Cultivar	2000	Alternative	Field Crops Central Research Institute, Ankara, Türkiye
Sarıçanak 98	Cultivar	1998	Spring	GAP International Agricultural Research and Training Center, Diyarbakir, Türkiye
Selçuklu-97	Cultivar	1997	Alternative	Bahri Dagdas International Agricultural Research Institute, Konya, Türkiye
<i>T. dicoccum</i> (Emmer)	Landrace	-	Alternative	Collected from Kars Province, Türkiye

2.2. Treatment Conditions and Plant Growth

The seeds underwent sterilization in 5% commercial bleach (NaClO) for a duration of twenty minutes, followed by thorough washing with distilled water. Germination was carried out in dark conditions at a temperature of 25 ± 1 °C. Drought stress was induced using high molecular weight polyethylene glycol (PEG 6000) following the protocol outlined by Michel and Kaufmann [28]. The levels of drought intensity (-0.50 , -1.48 , -2.95 , and -4.91 bar) were determined based on the work of previous researchers who deemed them appropriate [29]. Data collection was conducted in accordance with the measurements and counts established on the fourth and eighth days, following the guidelines provided by the International Seed Testing Association (ISTA) [30].

Germination was considered complete when the radicles reached a length of 2 mm. Subsequently, on the fourth day, the number of germinated seeds was recorded, and the germination rate (GS) was calculated. On the eighth day, the germination strength (GP) was determined by counting the germinated seeds. Additionally, the root length (RL) was measured as the longest root formed by the seed, the shoot length (SL) was measured as the endpoint of the plumule emerging from the coleoptile, and the fresh weight (FW) was determined by weighing all embryonic roots, the coleoptile, and the plumule immediately [31,32]. To calculate the dry weight (DW) of the plants, fresh samples

underwent a drying process at 105 °C for two hours, and the subsequent mass was recorded. The water absorption capacity (WAC) was determined by calculating the difference between the FW and DW, and the percentage rate was subsequently computed.

Fischer and Maurer [33] proposed the stress susceptibility index (SSI) as a means of measuring trait stability, considering changes in both potential and actual traits in variable environments. Clarke et al. [34] evaluated the drought tolerance of wheat genotypes using the SSI and found different annual variations in the SSI for different genotypes, allowing a ranking of their patterns. An SSI > 1 in spring wheat cultivars indicates an above-average susceptibility to drought stress, according to Guttieri et al. [35]. A smaller value of SSI is preferred as larger values indicate a greater sensitivity to stress. Fernandez [36] proposed the Stress Tolerance Index (STI) as a tool for identifying genotypes that exhibit high grain yield under contrasting conditions. The STI is designed to identify genotypes with favorable characteristics for the examined traits under both stressful and normal conditions.

Y_{pi} = The value of observed feature of each genotype under normal conditions (control)

Y_{si} = The value of the feature of each genotype under stressful conditions

Y_p = The mean of the features of the genotypes examined under normal conditions (control)

Y_s = The mean of the features of the genotypes examined under stressful conditions

- Stress tolerance index: $STI = \frac{Y_{pi} \times Y_{si}}{Y_p^2}$
- Stress intensity: $SI = 1 - \frac{Y_s}{Y_p}$
- Stress susceptibility index: $(SSI) = \frac{1 - (\frac{Y_{si}}{Y_{pi}})}{SI}$

2.3. Experiment Design and Statistical Analysis

A completely randomized design was employed with four replications per genotype and per stress level. A unit of replication were a Petri dish containing 50 seeds. Statistical analyses of the acquired data were conducted using JMP 13.2.0 with SAS software version 9.4. The impacts of genotype and stress level on germination responses were assessed through analysis of variance and Duncan tests. PCA was carried out using standardized average values for each germination parameter and the STI value for genotypes, providing insight into their tolerance or susceptibility to drought stress. The dissimilarity between genotypes was calculated using Ward's method of clustering. Correlations between the examined traits of tetraploid wheat genotypes under control and drought stress treatments were computed using JMP 13.2.0 [37].

2.4. Machine Learning Analysis and Model Assessment

The primary aim of this study was to determine the relationships between input variables (genotype and drought stress) and the output variable (observed germination parameters) to develop a predictive model. Four ML techniques were employed, namely support vector machines (SVM) [38], extreme gradient boosting (XGBoost) [39], elastic-net (ELANET) [40], and the Gaussian processes classifier (GPC) [41]. The evaluation of algorithm performance utilized three key metrics: root mean square error (MSE), R-squared (R^2), and mean absolute deviation (MAD). The coefficient of determination, denoted as R^2 , gauges the extent to which the model (Equation (1)) replicates the observed data. MSE measures the proximity between the predicted and actual values, as expressed by Equation (2). Additionally, the root MAD characterizes the overall distribution of prediction errors, as outlined in Equation (3) [20,26,27]. The dataset was randomly divided into two sets using the caret package in the R software (version 4.3.2): the training set (70%) and the testing set (30%). The Grid Search Cross-Validation (GCV) method was employed to identify the optimal hyperparameters for each ML model, as indicated by Equation (4) [42].

$$R^2 = 1 - \left(\frac{\sum_{i=1}^n (y_i - y_{ip})^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \right) \quad (1)$$

$$\text{MSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - y_{ip})^2} \quad (2)$$

$$\text{MAD} = \frac{1}{n} \sum_{i=1}^n |y_i - y_{ip}| \quad (3)$$

$$\text{GCV}(\lambda) = \frac{\sum_{i=1}^n (y_i - y_{ip})^2}{\left[1 - \frac{M(\lambda)}{n}\right]^2} \quad (4)$$

where n is the training/testing sample size in the dataset, y_i is the measured real value, y_{ip} is the predicted value, and \bar{y} is the measured values mean. $M(\lambda)$ is the penalty function for the complexity of the model covering λ terms. The R program was used for the computation of ML algorithms and performance metrics [26,27].

3. Results

According to the variance analysis of the data, it was determined that the differences among genotypes and stress levels were significant at $p < 0.01$ for all examined parameters. Additionally, the Genotype \times Stress interaction was found to have a significant effect at $p < 0.01$ on all parameters, except for germination power (Table 2). Figure 1 displays the mean, minimum, and maximum values resulting from the interaction between genotype and stress.

Table 2. Variance analysis of tetraploid wheat genotypes' germination performances under different drought stress levels.

Variation Source	df	Mean Square						
		GS	GP	RL	SL	FW	DW	WAC
Genotype (G)	11	184.833 **	97.391 **	17.668 **	10.879 **	0.290 **	0.047 **	242.055 **
Stress level (S)	4	2981.008 **	642.359 **	628.871 **	522.911 **	5.938 **	0.057 **	9299.92 **
G \times S	44	35.308 **	16.267 ns	2.936 **	6.641 **	0.053 **	0.005 **	72.837 **

GS: germination speed; GP: germination power; RL: root length; SL: shoot length; FW: fresh weight; DW: dry weight; WAC: water absorption capacity. ** 0.01 significant at the probability level; ns: non-significant.

3.1. Germination Speed and Germination Power

The results suggest that the impact of drought stress on germination rate is more significant than its effect on GP. Specifically, compared to the control group, the germination rate of genotypes under -1.48 bar drought stress was significantly reduced. Notably, Artuklu exhibited the highest GS, particularly at -4.91 bar, representing the highest level of drought stress. At this maximum stress level, *T. dicoccum* displayed the lowest GS (66.6%) (Supplementary Table S1). Considering stress-tolerance indices, the varieties demonstrating the highest tolerance in terms of GS were Eminbey, Kızıltan-91, and Mirzabey 2000 at -0.50 bar drought stress, Altın-40/98 at -1.48 bar drought stress, and Artuklu at -2.95 and -4.91 bar drought stress (Supplementary Table S2). Conversely, based on the SSI, Kunduru-1149, Eminbey, Selçuk-97, and Meram-2002 were significantly affected by drought stress, resulting in a notable reduction in GS. The highest stress-susceptibility indices were observed in Kunduru-1149 at -0.50 bar stress, Eminbey and Selçuklu-97 at -1.48 bar stress, and Meram-2002 at -2.95 bar and -4.91 bar stress levels. Moreover, drought stress had a comparatively lower impact on the germination rate of tetraploid wheat genotypes. It was observed that the GP of these varieties was not affected by the -0.50 bar level, representing the lowest drought stress level, and remained comparable to the control group (Supplementary Table S1). Across the different levels of drought stress, Altın-40/98 displayed the highest average GP, while *T. dicoccum* exhibited the lowest. According to the STI, Altın-40/98 achieved the highest tolerance values at all stress levels. Conversely, based on the SSI, the varieties whose germination power was most affected by drought

stress were identified as Artuklu, Kunduru-1149, and Selçuklu-97 at -0.50 bar, Selçuklu-97 at -1.48 bar, Eminbey at -2.95 bar, and Eminbey and Kunduru-1149 at -4.91 bar (Supplementary Table S2). When the genotypes were analyzed in terms of GS and GP, it became evident that there were significant decreases, especially at the fourth and fifth stress levels.

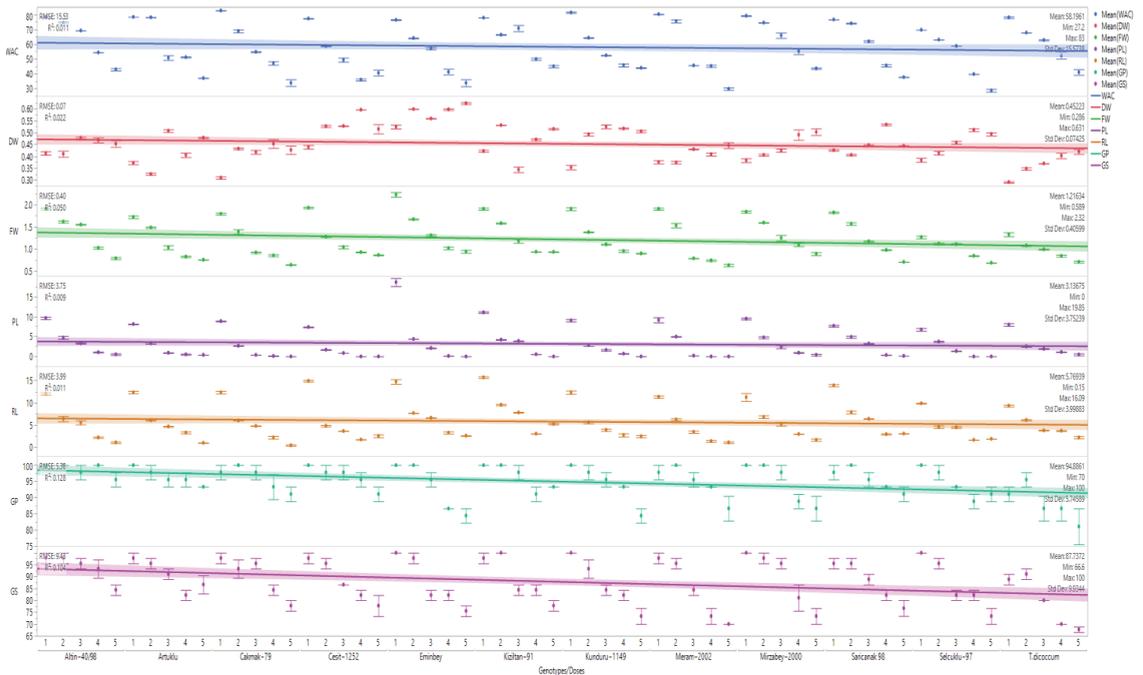


Figure 1. The meaning of parameters examined according to genotype and stress levels. GS: germination speed; GP: germination power; RL: root length; SL: shoot length; FW: fresh weight; DW: dry weight; WAC: water absorption capacity; RMSE: root mean square error; R^2 : coefficient of determination.

Even though -0.50 bar represented the lowest drought stress level, it caused a significant reduction in the root length of all genotypes compared to the control group. The decreases in root length persisted with increasing stress levels. Notably, Kızıltan-91 and Sarıçanak 98 exhibited the highest mean root length at different drought stresses. Specifically, Kızıltan-91, at -0.50 bar, -1.48 bar, and -4.91 bar stress levels, and *T. dicoccum* at the -2.95 bar stress level showed the most robust root formation (Supplementary Table S2). The most tolerant genotypes in terms of root development, according to the STI, were Kızıltan-91 at -0.50 bar and -1.49 bar stresses, Eminbey at -2.95 bar stress, and Kızıltan-91 at -4.91 bar stress. On the other hand, based on the SSI, the genotypes where drought stress significantly affected root development were Çeşit-125 at the -0.50 bar, -1.48 bar, and -2.95 bar drought levels, and Altın-40/98 and Çakmak-79 at -4.91 bar stress (Supplementary Table S2).

Moreover, drought stress had a more pronounced impact on shoot development compared to root development in tetraploid wheat genotypes. When examining the average shoot length of the genotypes, there was an approximately 61% decrease compared to the control group at -0.50 bar, the lowest drought level in the study. The sharp declines in shoot length continued with increasing stress levels. Shoots did not occur in some genotypes at drought stresses of -2.95 bar (Çeşit-1252, Meram-2002, and Selçuklu-97) and -4.91 bar (Çakmak-79, Çeşit-1252, Eminbey, Kızıltan-91, Meram-2002, and Selçuklu-97). Eminbey formed the highest shoot length in terms of the general shoot length average of

genotypes but could not produce shoots at high-stress levels. However, according to the results under stress pressure, the genotype that produced the most shoots was Sarıçanak 98. When stress tolerance values were considered based on shoot-forming abilities, Eminbey at -0.50 bar stress, Kızıltan-91 at -1.49 bar stress, Altın-40/98 at -2.95 bar stress, and Altın-40/98 and Sarıçanak 98 varieties at -4.91 bar stress had the highest index values. However, according to the SSI, at -0.50 bar drought, Çeşit-1252; at -1.49 bar, Meram-2002; at -2.95 bar, Çeşit-1252, Meram-2002, and Selçuklu-97; at -4.91 bar, Çakmak-79, Çeşit-1252, Eminbey, Kızıltan-91, Kunduru-1149, Meram-2002, and Selçuklu-97 were identified as the most susceptible varieties in terms of shoot elongation (Supplementary Table S2).

3.2. Fresh Dry Weight and Water Absorption Capacity

The FW of tetraploid wheat genotypes were not as significantly affected as root and shoot formation under the pressure of drought. The highest mean FW was observed in the control group, with gradual decreases occurring with increasing stress levels. Specifically, the highest average FW was observed in Eminbey, while the lowest was in Selçuklu-97. FW was measured to be the highest in the control group and at -0.50 bar for Eminbey, at -1.49 bar for Altın 40/98, and at -2.95 and -4.91 bar for Mirzabey-2000 (Supplementary Table S1). According to the STI, the genotype most tolerant regarding FW was Eminbey at -0.50 bar, -2.95 bar, and -4.91 bar stresses, and Altın-40/98 at -1.48 bar stress. On the other hand, based on the SSI in terms of FW, the most vulnerable genotypes were Çeşit-1252 at the -0.50 bar drought level, Meram 2002 at the -1.48 bar, -2.95 bar, and -4.91 bar drought levels (Supplementary Table S2). The DW average was highest in Eminbey and lowest in *T. dicoccum*. Eminbey consistently had the highest DW values in both the control group and under all stress conditions (Supplementary Table S1). According to the STI, Eminbey displayed the highest tolerance values at all stresses regarding DW. However, based on the SSI, it showed the highest susceptibility in terms of DW, with Çakmak-79 being the most susceptible at the -0.50 bar, -2.95 bar, and -4.91 bar drought levels, and Kunduru-1149 being the most susceptible at the -1.49 bar and 2.95 bar drought levels (Supplementary Table S2).

Furthermore, the water absorption abilities of the tetraploid wheat genotypes exhibited a significant decrease with the increasing pressure of drought because of drought stress. There was a 53% decrease in water absorption averages of the genotypes between stress-free conditions and the highest stress level. The WAC averages at all stress levels were highest for Mirzabey 2000 and lowest for Selçuklu 97. According to the STI, the most tolerant genotypes regarding WAC were Artuklu at -0.50 bar, Altın-40/98 at -1.48 bar, and Mirzabey 2000 at -2.95 bar and -4.91 bar stress. The STI for WAC was highest for Artuklu at -0.50 bar stress, Altın-40/98 at -1.48 bar stress, and Mirzabey 2000 at -2.95 bar and -4.91 bar stresses. Conversely, based on the SSI, the most sensitive varieties in terms of water absorption were Çeşit-1252 at the -0.50 bar and -2.95 bar drought levels, Meram 2002 at the -1.48 bar drought level, and Eminbey at the -4.91 bar drought level.

3.3. Multivariate Analysis

The STI and SSI demonstrated inversely proportional values. Therefore, the STI was used in the first PCA. The PCA revealed a high level of variation among the tetraploid wheat genotypes. The distribution of tetraploid wheat genotypes concerning the first two principal components under different drought levels, based on morphological characteristics and STI values, was tabulated in a biplot (Figure 2). The variation examined with PCA indicated that the first two principal components contributed 75.20% of the total variance among the seven variable germination traits under normal conditions. Furthermore, the first two principal components contributed to 77.75%, 68.47%, 67.20%, and 63.37% of the total variance among the seven germination traits and seven STI values for the -0.50 bar, -1.48 bar, -2.95 bar, and -4.91 bar drought stress levels, respectively. Taking into consideration the means of the fourteen values examined, it was determined that the first two principal components constituted 69.63% of the total variation (Supplementary Table S3).

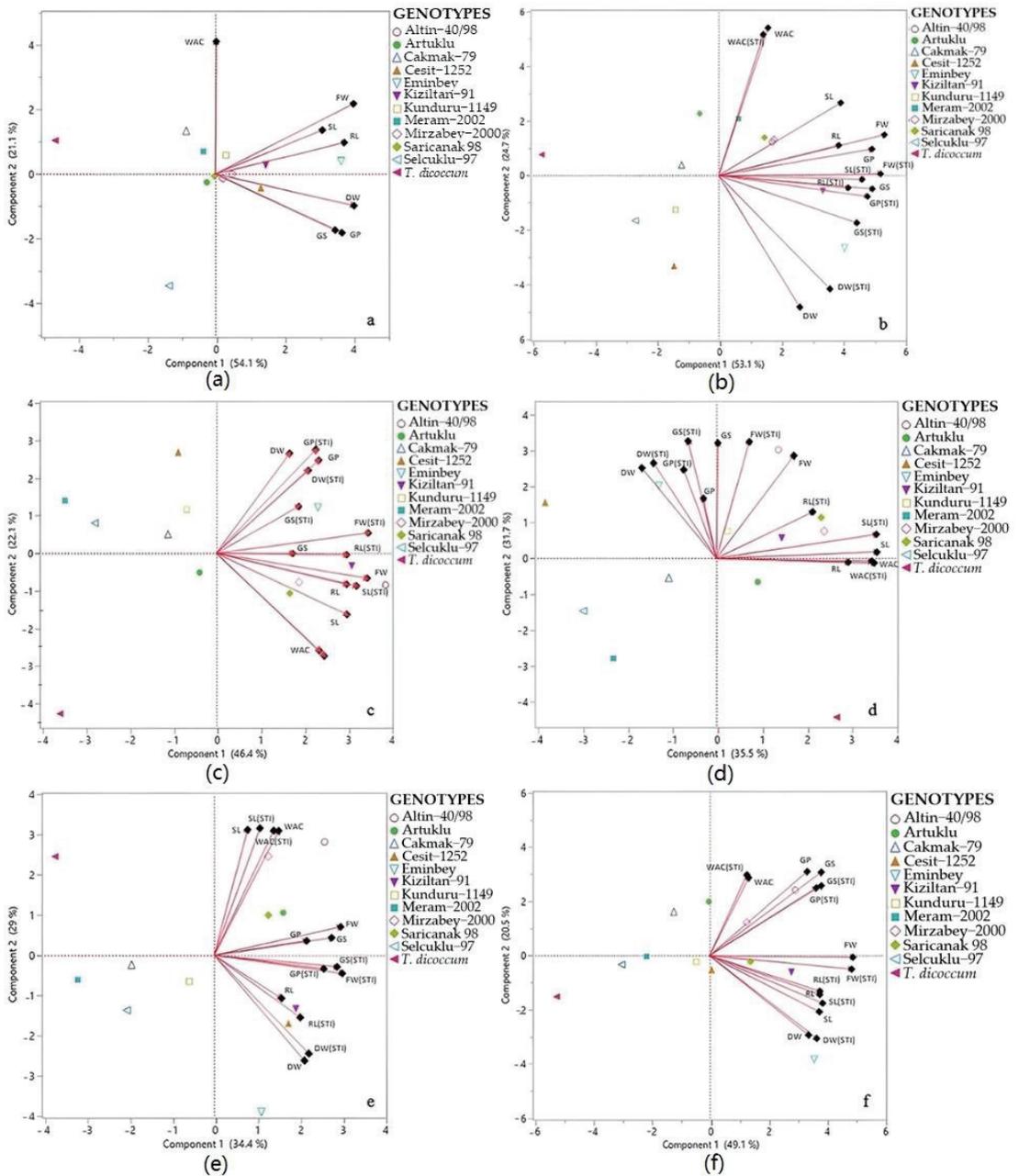


Figure 2. Classification of tetraploid wheat genotypes for different drought stress levels along with the first and second principal components on biplots ((a) control, (b) -0.50 bar, (c) -1.48 bar, (d) -2.95 bar, (e) -4.91 bar, and (f) mean of all). GS: germination speed; GP: germination power; RL: root length; SL: shoot length; FW: fresh weight; DW: dry weight; WAC: water absorption capacity; STI: stress tolerance index.

The first two principal components were graphically plotted to illustrate the similarities among genotypes at different drought stress levels (Figure 2). The biplot graphs were designed by computing each feature individually to separate the stress levels, demonstrating the variability of genotypes for the seven morphological traits and seven STI values in the study. The graphical representation on the biplot indicates a wide genetic variability among the genotypes based on their distribution model under different drought stresses (Figure 2). Considering the positive and high values of these two principal components on the biplot, genotypes located near the investigated traits are likely to be highly efficient under both stressed and unstressed conditions. Notably, in the biplot graphs, Selçuklu-97 and Meram-2002 are seen to be in the negative direction of the first two principal components. This suggests that these genotypes are more susceptible to drought stress than others. Throughout all the biplot graphs, *T. dicoccum* was observed to be located at a distance from the other genotypes, indicating a potential difference in its genetic structure. In the biplot charts, it is evident that Artuklu, Eminbey, Kızıltan-91, and Sarıçanak-98 were distributed near tolerance indicators. This suggests that these genotypes may possess a higher stress tolerance compared to others. The biplot analysis provides valuable insights into the genetic variability and stress response of the tetraploid wheat genotypes under different drought conditions.

In the second biplot analysis (Figure 2), conducted to reveal the relationships among genotypes and all parameters examined, a total of 11 principal component axes were obtained, with five principal component axes having an eigenvalue greater than 1.0. The eigenvalues of these five principal component axes, which collectively account for 91.85% of the total variation, range between 1.06 and 7.45. Upon examining the angles between the axes, a high angle among the SSI(WAC), SSI(GS), and SSI(SL) axes and the STI(WAC), WAC axes suggests a highly negative correlation among these parameters. Conversely, the slight angle between the GP axis and the axes of STI(GS), STI(GP), and GS in the same region reveals a highly positive relationship among these parameters. Considering the axis regions and lengths of all parameters, as well as their angles with each other and the positions of genotypes, *T. dicoccum* is located alone and in a region opposite to the axes. Considering this analysis, *T. dicoccum* appears to have the lowest values regarding the examined properties compared to the other genotypes. The biplot analysis provides a comprehensive view of the relationships among the genotypes and various parameters, aiding in the interpretation of the dataset and highlighting the characteristics that contribute most to the observed variability.

Additionally, according to Figure 3, *T. dicoccum* is positioned in the highest class in parameters close to the Eminbey axis tip, clustered in the same region with the Kızıltan 91 and Altın 40/98 varieties. Upon examining the WAC and STI (WAC) axes, Mirzabey-2000 and Sarıçanak 98 emerge as the leading genotypes (Figure 3, Supplementary Table S2).

To analyze the interactions between genotypes throughout the germination period and to decide which genotypes are appropriate for future breeding programs, a cluster analysis was conducted using the evaluated characteristics of the genotypes that were subjected to drought stress. Understanding the diversity in parents is crucial for the improvement of breeding programs aimed at developing new, tolerant varieties. According to the cluster analysis, the genotypes with the least genetic diversity distance between them concerning GP and stress indexes were identified as Çeşit-1252 and Kunduru-1149. On the other hand, the genotypes with the furthest genetic diversity distance between them were Artuklu and *T. dicoccum* (Figure 4). The findings suggest that *T. dicoccum* exhibits distinctive genetic characteristics compared to the other genotypes, particularly in terms of germination performance under drought stress. This differentiation is further emphasized by both the principal component analysis and cluster analysis. These insights can inform future breeding strategies for developing improved, drought-tolerant wheat varieties.

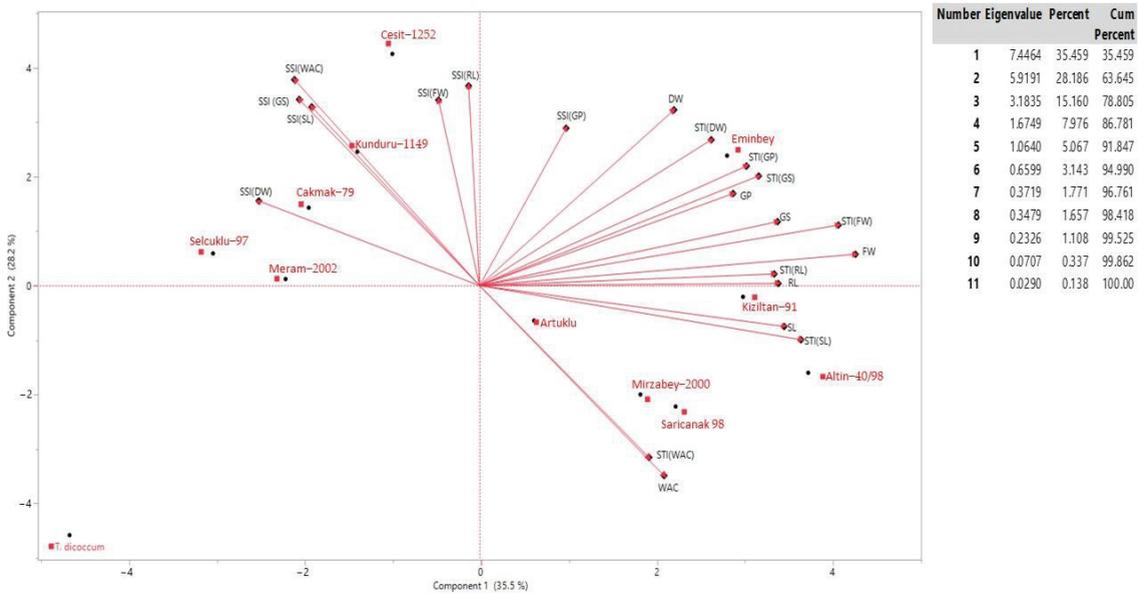


Figure 3. Principal component analysis (biplot) and output values of genotypes as well as examined parameters. GS: germination speed; GP: germination power; RL: root length; SL: shoot length; FW: fresh weight; DW: dry weight; WAC: water absorption capacity; STI: stress tolerance index.

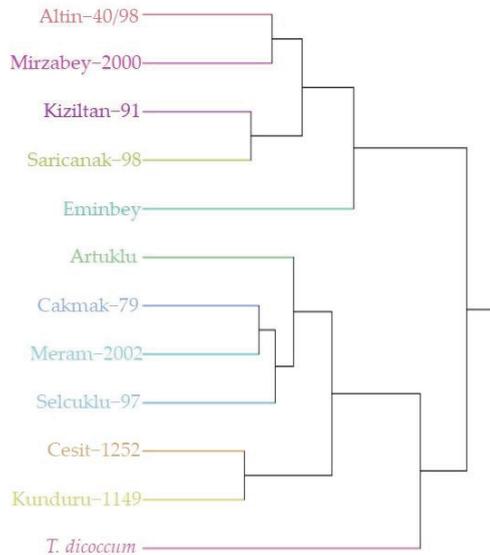


Figure 4. Phylogenetic tree of tetraploid wheat genotypes according to germination performance and tolerance indices under different drought stresses.

The correlation analysis revealed statistically significant relationships among all parameters. According to the analysis, the highest positive correlation among the quantitative traits was identified between FW and RL (0.917). Conversely, the highest negative correlation was observed between WAC and DW (−0.549). Notably, there was a consistent negative correlation between DW and all the other parameters (Figure 5).



Figure 5. Heat map showing the correlation between germination parameters in tetraploid wheat genotypes under various levels of drought stress. * Significant at the 0.05 probability level; ** significant at the 0.01 probability level; ns: non-significant. GS: germination speed; GP: germination power; RL: root length; SL: shoot length; FW: fresh weight; DW: dry weight; WAC: water absorption capacity; r: Pearson’s correlation.

3.4. Machine Learning (ML) Analysis

The responses of *Triticum durum* wheat to drought stress were modeled with respect to various variables, including GS, GP, RL, SL, FW, DW, and WAC. Predictions were generated using four different ML techniques: SVM, XGBoost, Elastic Net (ELNET), and GPC. The modeling approach employed a total of twelve wheat genotypes (Table 1) and involved five levels of drought stress (0, -0.50, -1.48, -2.95, and -4.91 bar) applied to the genotypes. Separate models were developed for each of the seven variables studied as output parameters. The study’s findings are presented comprehensively in Table 3, providing an overview of the outcomes generated by the ML models utilized in the investigation. Evaluation metrics, such as MSE and MAD, were employed to assess the overall performance of the algorithms. A reduction in the values of these metrics indicates that the model predictions are becoming closer to the actual observed values. Additionally, the study evaluated the extent to which the R² model could elucidate the variability observed between the independent factors and the dependent variable under investigation. The performances of the SVM, XGBoost, ELNET, and GPC models were assessed using a GCV approach. Among the various evaluation metrics, the XGBoost model exhibited the lowest MSE and MAD values, indicating a superior predictive accuracy for the training data. The MSE values for the variables GS, GP, RL, SL, FW, DW, and WAC were found to be 3.280, 2.291, 0.355, 0.263, 0.058, 0.012, 0.008, and 1.259, respectively. Additionally, the MAD values for the same variables were determined to be 2.591, 1.426, 0.245, 0.157, 0.039, and 0.795, respectively. Furthermore, this model demonstrated the highest R² coefficient while making predictions for the variables (GS, GP, RL, SL, FW, DW, and WAC) using the training dataset (0.890, 0.789, 0.992, 0.995, 0.980, 0.974, and 0.993, respectively). For the training dataset, the XGBoost model emerged as the top-performing model compared to the other models (Table 3). Each trained ML model underwent evaluation by making predictions based on the test dataset (Table 3). This helps estimate how well the model is likely to perform on new, unseen data. Based on the evaluation of the R², MSE, and MAD

metrics, the ELNET model demonstrated a superior performance in predicting GS, GP, and WAC. Furthermore, while evaluating the criteria metrics, it was concluded that the GPC model had a better performance in predicting RL, but the XGBoost model had a better performance in estimating SL, FW, and DW. Figure 6 displays the linear regression graph depicting the projected values of the models that provide the most accurate prediction model for the variables under examination, alongside the observed actual values.

Table 3. Algorithms’ goodness-of-fit criteria for prediction of variables.

Observed Variable	ML Criterion	SVM		XGBoost		ELNET		GPC	
		Train	Test	Train	Test	Train	Test	Train	Test
GS ¹	R ²	0.801	0.600	0.890	0.730	0.796	0.762	0.871	0.715
	MSE	4.406	6.320	3.280	5.190	4.455	4.873	3.540	5.339
	MAD	3.389	4.831	2.591	3.861	3.608	3.755	2.990	4.266
GP	R ²	0.655	0.310	0.789	0.352	0.621	0.514	0.758	0.349
	MSE	2.934	5.894	2.291	5.713	3.074	4.946	2.455	5.725
	MAD	1.794	4.264	1.426	3.915	2.267	3.581	1.730	4.092
RL	R ²	0.866	0.736	0.992	0.980	0.944	0.949	0.987	0.981
	MSE	1.454	2.072	0.355	0.571	0.936	0.915	0.449	0.552
	MAD	0.802	1.217	0.245	0.409	0.706	0.705	0.329	0.407
SL	R ²	0.779	0.673	0.995	0.962	0.887	0.852	0.990	0.942
	MSE	1.723	2.244	0.263	0.761	1.234	1.512	0.368	0.944
	MAD	0.758	1.147	0.157	0.457	0.794	0.912	0.230	0.501
FW	R ²	0.903	0.759	0.980	0.962	0.915	0.892	0.974	0.945
	MSE	0.128	0.193	0.058	0.077	0.120	0.129	0.066	0.092
	MAD	0.080	0.125	0.039	0.056	0.093	0.106	0.047	0.070
DW	R ²	0.901	0.807	0.974	0.944	0.733	0.793	0.935	0.886
	MSE	0.023	0.033	0.012	0.018	0.038	0.034	0.019	0.025
	MAD	0.015	0.022	0.008	0.014	0.029	0.028	0.014	0.020
WAC	R ²	0.924	0.830	0.993	0.891	0.949	0.902	0.989	0.880
	MSE	4.309	6.750	1.259	5.419	3.505	5.140	1.657	5.684
	MAD	2.903	4.738	0.795	3.914	2.844	2.492	1.238	3.235

¹ GS: germination speed; GP: germination power; RL: root length; SL: shoot length; FW: fresh weight; DW: dry weight; WAC: water absorption capacity; MSE: root mean squared error; MAD: mean absolute deviation; SVM: support vector machines; XGBoost: extreme gradient boosting; ELNET: elastic-net; GPC: Gaussian processes classifier.

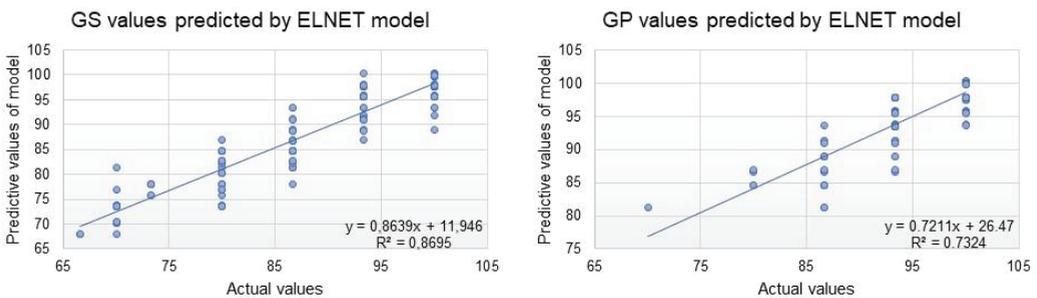


Figure 6. Cont.

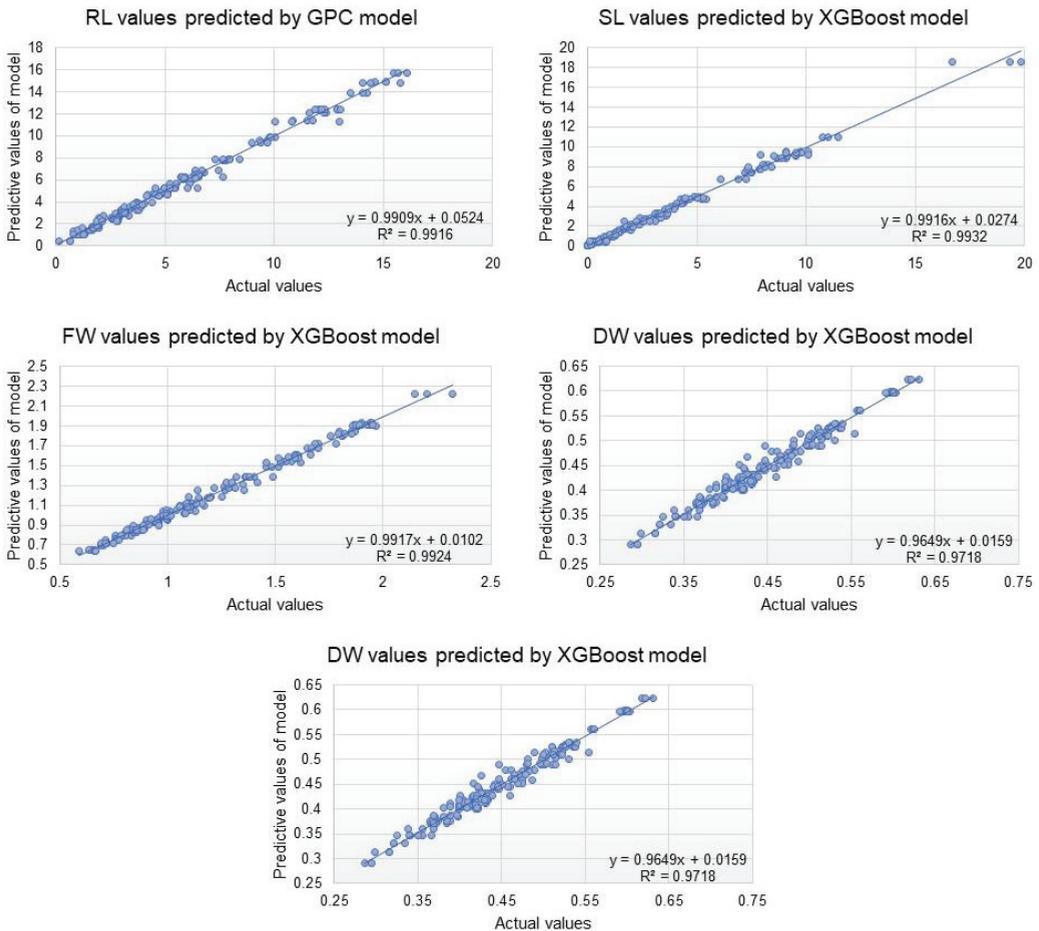


Figure 6. Based on the test set forecast, linear regression of the best models' expected values and their real values. GS: germination speed; GP: germination power; RL: root length; SL: shoot length; FW: fresh weight; DW: dry weight; WAC: water absorption capacity; SVM: support vector machines; XGBoost: extreme gradient boosting; ELNET: elastic-net; GPC: Gaussian processes classifier.

4. Discussion

The successful cultivation of tetraploid wheat, with its high yields, hinges upon its ability to withstand drought stress during early development. A pivotal aspect in achieving resilient strains is the identification of drought tolerance levels in existing varieties, a crucial step in effective breeding programs. This emphasis on drought tolerance is underscored by prior research conducted by Sayar et al. [43] and Aslan et al. [44], where the assessment of wheat responses to drought stress during germination, using PEG like our current study, aligns with the present research focus. Both studies emphasized the significance of determining tolerance levels, particularly during germination, as an essential measure in addressing challenges posed by drought. Notably, Badr et al. [45] stressed the importance of studying stress-responsive traits in genotype germination performance as a swift and effective screening method for identifying drought-resistant genotypes. The current research aimed to expand upon these insights by delving into the drought tolerance of diverse tetraploid wheat genotypes during the initial growth stage, employing a multivariate analysis and ML algorithms.

The research findings highlight a noticeable impairment in the germination performance of tetraploid wheat genotypes with increasing drought stress. This aligns with the observations of Aslan et al. [44] and Benlioglu and Ozgen [2], who similarly reported a negative impact of drought on the germination of tetraploid wheat. Root length emerges as a critical factor influencing wheat tolerance to drought during the early growth stages [44]. A longer root allows the plant to penetrate deeper into the soil, accessing more water resources. Varieties that excel in root formation during germination in arid conditions tend to exhibit better shoot development in subsequent growth stages. Therefore, both RL and SL are important features in evaluating drought tolerance [46]. In this study, Artuklu, Eminbey, Kızıltan-91, Sarıçanak-98, and *T. dicoccum* stood out in terms of root development under high drought stress. The results indicate that drought stress significantly affected shoot development compared to root growth, consistent with previous studies [47]. Sarıçanak 98 and Mirzabey 2000 outperformed other genotypes, demonstrating superior shoot formation and development, particularly under conditions of high drought stress. The increasing drought pressure in the external environment hampers seed absorption of sufficient water for germination [48]. Thus, drought stress hinders the use of stored nutrients in the seed for shoot and root development [49]. In this study, an increase in dry weight during the germination phases of tetraploid wheat varieties was observed to coincide with escalating stress levels. This phenomenon may be attributed to limited water access due to the heightened drought stress, which prevents the optimal utilization of storage nutrients [50]. Consequently, the impact of drought stress on FW was more pronounced than on DW, which is consistent with the findings of Sayar et al. [43]. In contrast to studies conducted on barley [50] and wheat [46], our results suggest a tendency for DW to increase with elevated stress levels.

Many studies conducted at different ploidy levels have consistently indicated that drought stress significantly reduces grain yield, yield components, harvest index, plant height, leaf area, and DW [51]. The utilization of STI and SSI provides a clear differentiation among tetraploid wheat genotypes based on their germination performance under varying drought stress levels [52]. However, it is important to note that STI and SSI may not always yield parallel results, especially when there is no statistically significant difference between genotypes and treatments. This discrepancy was evident in our study, particularly in the GP parameter, where the difference between genotypes and treatments was statistically insignificant. Similarly, in the DW parameter, which exhibits an inverse relationship with stress levels, these indexes may be deemed inappropriate for assessing tolerance. This discrepancy in results could be attributed to the sensitivity of these indices to statistical variations and their effectiveness in capturing differences under specific conditions. Researchers should be cautious and consider alternative indices or additional statistical analyses when dealing with parameters that may not align well with STI and SSI, especially in cases where there is no statistically significant difference between genotypes and treatments.

Selection based on a combination of indices can be a valuable strategy for developing drought tolerance in tetraploid wheat. Utilizing methods like PCA is crucial for characterizing genotypes under stress conditions. A narrow vector angle of axes in the same region indicates significant positive relationships among these elements [53]. Moreover, for the efficient use of PCA and accurate interpretation of results, it is recommended that the first two or three principal components explain at least 25% of the total variation [25]. The biplot graphs generated in our study revealed that Artuklu, Eminbey, Kızıltan-91, and Sarıçanak-98 were positioned near the tolerance indicators, indicating a higher drought stress tolerance compared to the other genotypes. Conversely, Selçuklu-97, Meram-2002, and *T. dicoccum* were located further away from tolerance indicators, suggesting a lower drought tolerance. This observation aligns with the findings of [54], who noted that domestication, selection, and breeding of wheat have positively influenced the improvement of above-ground biomass, ultimately increasing wheat yield. Similarly, our study indicated that *T. dicoccum* exhibited a low drought tolerance during the germination period. The

germination performance of tetraploid wheat varieties under different drought stress levels unveiled a wide variation among genotypes, emphasizing the importance of a multifaceted approach for selecting genotypes with robust drought tolerance.

Methods based on ML offer the advantage of creating nonlinear correlations between yield factors and independent samples [20,26]. ML has demonstrated significant advancements compared to traditional regression models that rely on linear associations [27,32,55]. This technique allows for a comprehensive examination of multitemporal field sample data, facilitating the optimization of plant growing conditions and crop output [56–58]. The utilization of historical sampling and prediction models is crucial for enhancing future crop management strategies, providing valuable insights into the impact of previous farming practices and abiotic variables on observed target metrics [5]. Moreover, the selection of algorithms used in the estimation process holds significant importance. Various measures, such as MSE, MAD, and R^2 , are employed to determine the most optimal and superior performing algorithms [27]. Based on the results obtained from this research, it can be concluded that the ELNET model demonstrated a higher level of effectiveness in forecasting GS, GP, and WAC. The analysis also revealed that the GPC model exhibited superior predictive performance in relation to RL, while the XGBoost model demonstrated a better predictive performance for SL, FW, and DW. The models that exhibited the highest predictive accuracy in relation to the observed variables had a range of values between 0.7324 and 0.9932, as indicated by the linear regression R^2 metric (Figure 6). The SL parameter estimate yielded the highest accuracy in the R^2 value measurement. The use of these algorithms in forthcoming breeding investigations has the potential to enhance the decision-making process in tetraploid wheats under drought situations using data-driven methodologies. Furthermore, this study provides a fast and cost-effective method to determine the tolerance of genotypes to drought stress.

5. Conclusions

The responses and tolerance levels of tetraploid wheat genotypes to drought stress during the early period were assessed based on germination performance, STI, SSI, and a multivariate analysis. All PEG treatments led to reduced germination rates and delayed seedling growth. Severe drought stress levels of -2.95 bar and -4.91 bar in the early period inhibited further growth in some genotypes, indicating the sensitivity of these genotypes to these stress levels. The STI and SSI effectively conveyed the tolerance and susceptibility levels of the genotypes. The PCA provided valuable insights into tolerance levels and relationships among different parameters by analyzing the distribution of genotypes in the direction of tolerance indicators. The cluster analysis confirmed the results of the PCA, demonstrating a divergence across genotypes with respect to the characteristics under investigation. Moreover, the study suggests that employing a combined ML technique, potentially incorporating additional ML methods, could provide a reliable approach to establish the relationship between tetraploid wheats subjected to drought stress and their observed parameters during the early phases of growth. This integrated approach may offer a time-saving and cost-effective means of determining drought tolerance and susceptibility levels in tetraploid wheat. Overall, the methods and analyses applied in this study present a comprehensive and efficient approach for assessing drought stress in tetraploid wheat genotypes.

Supplementary Materials: The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/agriculture14020206/s1>, Table S1. The mean values and Duncan groups of germination parameters of tetraploid wheat genotypes under different drought stress levels. Table S2: Stress tolerance index (STI) and stress sensitive index (SSI) values of tetraploid wheat genotypes. Table S3: Eigenvalues and percent of total variation for the principal components.

Author Contributions: Conceptualization, B.B., F.D. and G.N.; methodology, B.B., F.D., A.T., K.H., H.Ö. and G.N.; software, B.B., A.T., F.D., S.K., M.P., T.W. and G.N.; validation, A.T., K.H., F.D., M.P. and G.N.; formal analysis, A.T., K.H., F.D. and G.N.; investigation, B.B., F.D. and G.N.; resources,

B.B., F.D. and G.N.; data curation, A.T., K.H., F.D., S.K., M.P., T.W. and G.N.; writing—original draft preparation, B.B., F.D., A.T. and G.N.; writing—review and editing, B.B., F.D., A.T., K.H., H.Ö., S.K., M.P., T.W. and G.N.; visualization, B.B., F.D. and G.N.; supervision, B.B., H.Ö. and G.N.; project administration, B.B., F.D., A.T. and G.N.; funding acquisition, A.T. and K.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: All data supporting the conclusions of this article are included in this article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

- Loura, D.; Dhankar, A.; Kumar, S. Weed management practices in wheat (*Triticum aestivum* L.): A review. *Agric. Rev.* **2023**, *44*, 1–11.
- Benlioglu, B.; Ozgen, M. In vitro selection of drought tolerant regenerants in durum wheat (*Triticum durum* desf.). *Appl. Ecol. Environ. Res.* **2021**, *19*, 1813–1825. [CrossRef]
- Zingale, S.; Guarnaccia, P.; Matarazzo, A.; Lagioia, G.; Ingraio, C. A systematic literature review of life cycle assessments in the durum wheat sector. *Sci. Total Environ.* **2022**, *844*, 157230. [CrossRef] [PubMed]
- TUIK. *TSI: 2022 Türkiye Agricultural Production Values*; TUIK: Ankara, Türkiye, 2023.
- Varga, I.; Radočaj, D.; Jurišić, M.; Kulundžić, A.M.; Antunović, M. Prediction of sugar beet yield and quality parameters with varying nitrogen fertilization using ensemble decision trees and artificial neural networks. *Comput. Electron. Agric.* **2023**, *212*, 108076. [CrossRef]
- Balestrini, R.; Chitarra, W.; Ghirardo, A.; Nardini, A.; Nerva, L. A stressful life: How plants cope with multiple biotic and abiotic adverse factors. *Plant Stress* **2022**, *5*, 100095. [CrossRef]
- Cohen, I.; Zandalinas, S.I.; Huck, C.; Fritschi, F.B.; Mittler, R. Meta-analysis of drought and heat stress combination impact on crop yield and yield components. *Physiol. Plant.* **2021**, *171*, 66–76. [CrossRef] [PubMed]
- De Santis, M.A.; Soccio, M.; Laus, M.N.; Flagella, Z. Influence of drought and salt stress on durum wheat grain quality and composition: A review. *Plants* **2021**, *10*, 2599. [CrossRef] [PubMed]
- Gitz, V.; Meybeck, A.; Lipper, L.; Young, C.D.; Braatz, S. *Climate Change and Food Security: Risks and Responses*; FAO: Rome, Italy, 2016.
- Yu, H.; Zhang, Q.; Sun, P.; Song, C. Impact of droughts on winter wheat yield in different growth stages during 2001–2016 in Eastern China. *Int. J. Disaster Risk Sci.* **2018**, *9*, 376–391. [CrossRef]
- Desa, U. *World Population Prospects 2019: Highlights*; United Nations Department for Economic and Social Affairs: New York, NY, USA, 2019; Volume 11, p. 125.
- Özgen, M.; Birsin, M.A.; Benlioglu, B. Biotechnological characterization of a diverse set of wheat progenitors (*Aegilops* sp. and *Triticum* sp.) using callus culture parameters. *Plant Genet. Res.* **2017**, *15*, 45–50. [CrossRef]
- Ahmad, A.; Aslam, Z.; Javed, T.; Hussain, S.; Raza, A.; Shabbir, R.; Mora-Poblete, F.; Saeed, T.; Zulfiqar, F.; Ali, M.M. Screening of wheat (*Triticum aestivum* L.) genotypes for drought tolerance through agronomic and physiological response. *Agronomy* **2022**, *12*, 287. [CrossRef]
- Rai, R.; Agrawal, M.; Agrawal, S. Threat to food security under current levels of ground level ozone: A case study for Indian cultivars of rice. *Atmos. Environ.* **2010**, *44*, 4272–4282. [CrossRef]
- Gopal, J.; Iwama, K. In vitro screening of potato against water-stress mediated through sorbitol and polyethylene glycol. *Plant Cell Rep.* **2007**, *26*, 693–700. [CrossRef] [PubMed]
- Verslues, P.E.; Agarwal, M.; Katiyar-Agarwal, S.; Zhu, J.; Zhu, J.K. Methods and concepts in quantifying resistance to drought, salt and freezing, abiotic stresses that affect plant water status. *Plant J.* **2006**, *45*, 523–539. [CrossRef] [PubMed]
- Mustamu, N.E.; Tampubolon, K.; Basyuni, M.; Al-Taey, D.K.; Janabi, H.J.K.A.; Mehdizadeh, M. Drought stress induced by polyethylene glycol (PEG) in local maize at the early seedling stage. *Heliyon* **2023**, *9*, e20209. [CrossRef] [PubMed]
- Caruso, A.; Chefdor, F.; Carpin, S.; Depierreux, C.; Delmotte, F.M.; Kahlem, G.; Morabito, D. Physiological characterization and identification of genes differentially expressed in response to drought induced by PEG 6000 in *Populus canadensis* leaves. *J. Plant Physiol.* **2008**, *165*, 932–941. [CrossRef]
- Türkoğlu, A.; Bolouri, P.; Haliloğlu, K.; Eren, B.; Demirel, F.; Işık, M.I.; Piekutowska, M.; Wojciechowski, T.; Niedbała, G. Modeling callus induction and regeneration in hypocotyl explant of fodder pea (*Pisum sativum* var. *arvense* L.) using machine learning algorithm method. *Agronomy* **2023**, *13*, 2835. [CrossRef]
- Demirel, F.; Eren, B.; Yilmaz, A.; Türkoğlu, A.; Haliloğlu, K.; Niedbała, G.; Bujak, H.; Jamshidi, B.; Pour-Aboughadareh, A.; Bocianowski, J. Prediction of grain yield in wheat by CHAID and MARS algorithms analyses. *Agronomy* **2023**, *13*, 1438. [CrossRef]
- Niazian, M.; Niedbała, G. Machine learning for plant breeding and biotechnology. *Agriculture* **2020**, *10*, 436. [CrossRef]

22. Wahab, A.; Abdi, G.; Saleem, M.H.; Ali, B.; Ullah, S.; Shah, W.; Mumtaz, S.; Yasin, G.; Muresan, C.C.; Marc, R.A. Plants' physio-biochemical and phyto-hormonal responses to alleviate the adverse effects of drought stress: A comprehensive review. *Plants* **2022**, *11*, 1620. [CrossRef]
23. Kırnak, H.; Uzun, S.; Irik, H.A.; Özaktan, H.; Arslan, M. Water–yield relations of drip-irrigated peas under semi-arid climate condition. *Int. J. Agric. Nat. Sci.* **2021**, *14*, 85–95.
24. Sivakumar, J.; Prashanth, J.E.P.; Rajesh, N.; Reddy, S.M.; Pinjari, O.B. Principal component analysis approach for comprehensive screening of salt stress-tolerant tomato germplasm at the seedling stage. *J. Biosci.* **2020**, *45*, 141. [CrossRef]
25. Özaktan, H. Technological characteristics of chickpea (*Cicer arietinum* L.) cultivars grown under natural conditions. *Turkish J. Field Crops* **2021**, *26*, 235–243. [CrossRef]
26. Demirel, F.; Uğur, R.; Popescu, G.C.; Demirel, S.; Popescu, M. Usage of Machine learning algorithms for establishing an effective protocol for the in vitro micropropagation ability of black chokeberry (*Aronia melanocarpa* (Michx.) Elliott). *Horticulturae* **2023**, *9*, 1112. [CrossRef]
27. Eren, B.; Türkoğlu, A.; Haliloğlu, K.; Demirel, F.; Nowosad, K.; Özkan, G.; Niedbała, G.; Pour-Aboughadareh, A.; Bujak, H.; Bocianowski, J. Investigation of the influence of polyamines on mature embryo culture and DNA methylation of wheat (*Triticum aestivum* L.) using the machine learning algorithm method. *Plants* **2023**, *12*, 3261. [CrossRef] [PubMed]
28. Michel, B.E.; Kaufmann, M.R. The osmotic potential of polyethylene glycol 6000. *Plant Physiol.* **1973**, *51*, 914–916. [CrossRef] [PubMed]
29. El-Rawy, M.A.; Hassan, M.I. A diallel analysis of drought tolerance indices at seedling stage in bread wheat (*Triticum aestivum* L.). *Plant Breed. Biotechnol.* **2014**, *2*, 276–288. [CrossRef]
30. ISTA. *International Rules for Seed Testing. Intentional Seed Testing Association*; ISTA: Bassersdorf, Switzerland, 2013.
31. Haliloğlu, K.; Türkoğlu, A.; Aydin, M. Determination of imazamox herbicide dose in in vivo selection in wheat (*Triticum aestivum* L.). *Eregli J. Agric. Sci.* **2022**, *2*, 1–11.
32. Türkoğlu, A.; Tosun, M.; Haliloğlu, K.; Karagöz, H. Effects of early drought stress on germination and seedling growth parameters of Kirik bread wheat (*Triticum aestivum* L.). *Eregli J. Agric. Sci.* **2022**, *2*, 75–80. [CrossRef]
33. Fernandez, G.C. Stress tolerance index—a new indicator of tolerance. *HortScience* **1992**, *27*, 626d. [CrossRef]
34. Clarke, J.M.; DePauw, R.M.; Townley-Smith, T.F. Evaluation of methods for quantification of drought tolerance in wheat. *Crop Sci.* **1992**, *32*, 723–728. [CrossRef]
35. Guttieri, M.J.; Stark, J.C.; O'Brien, K.; Souza, E. Relative sensitivity of spring wheat grain yield and quality parameters to moisture deficit. *Crop Sci.* **2001**, *41*, 327–335. [CrossRef]
36. Fernandez, G.C. Effective selection criteria for assessing plant stress tolerance. In Proceedings of the International Symposium on Adaptation of Vegetables and Other Food Crops in Temperature and Water Stress, Shanhua, Taiwan, 13–16 August 1992; pp. 257–270.
37. Yuan, Y. Multiple imputation using SAS software. *J. Stat. Softw.* **2011**, *45*, 1–25. [CrossRef]
38. Noble, W.S. What is a support vector machine? *Nat. Biotechnol.* **2006**, *24*, 1565–1567. [CrossRef]
39. Chen, T.; Guestrin, C. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd ACM Sigkdd International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794.
40. John, M.; Haselbeck, F.; Dass, R.; Malisi, C.; Ricca, P.; Dreischer, C.; Schultheiss, S.J.; Grimm, D.G. A comparison of classical and machine learning-based phenotype prediction methods on simulated data and three plant species. *Front. Plant Sci.* **2022**, *13*, 932512. [CrossRef] [PubMed]
41. Rasmussen, C.E. Gaussian processes in machine learning. In *Summer School on Machine Learning*; Springer: Cambridge, MA, USA, 2003; pp. 63–71.
42. Camacho-Pérez, E.; Lugo-Quintal, J.M.; Tirink, C.; Aguilar-Quiñonez, J.A.; Gastelum-Delgado, M.A.; Lee-Rangel, H.A.; Roque-Jiménez, J.A.; García-Herrera, R.A.; Chay-Canul, A.J. Predicting carcass tissue composition in Blackbelly sheep using ultrasound measurements and machine learning methods. *Trop. Anim. Health Prod.* **2023**, *55*, 300. [CrossRef] [PubMed]
43. Sayar, R.; Bchini, H.; Mosbahi, M.; Ezzine, M. Effects of salt and drought stresses on germination, emergence and seedling growth of durum wheat (*Triticum durum* Desf.). *J. Agric. Res.* **2010**, *5*, 2008–2016.
44. Aslan, D.; Aktaş, H.; Ordu, B.; Zencirci, N. Evaluation of bread and einkorn wheat under in vitro drought stress. *J. Anim. Plant Sci.* **2017**, *27*, 1974–1983.
45. Badr, A.; El-Shazly, H.H.; Tarawneh, R.A.; Börner, A. Screening for drought tolerance in maize (*Zea mays* L.) germplasm using germination and seedling traits under simulated drought conditions. *Plants* **2020**, *9*, 565. [CrossRef] [PubMed]
46. Datta, J.; Mondal, T.; Banerjee, A.; Mondal, N. Assessment of drought tolerance of selected wheat cultivars under laboratory condition. *J. Agric. Technol.* **2011**, *7*, 383–393.
47. Benlioglu, B.; Ozkan, U. The influence of salinity and drought stress on some oat cultivars (*Avena sativa* L.) by determining some stress indexes and growth performances at the germination stage. *Fresenius Environ. Bull.* **2021**, *31*, 771–778.
48. Yang, X.; Lu, M.; Wang, Y.; Wang, Y.; Liu, Z.; Chen, S. Response mechanism of plants to drought stress. *Horticulturae* **2021**, *7*, 50. [CrossRef]
49. Khaeim, H.; Kende, Z.; Jolánkai, M.; Kovács, G.P.; Gyuricza, C.; Tarnawa, Á. Impact of temperature and water on seed germination and seedling growth of maize (*Zea mays* L.). *Agronomy* **2022**, *12*, 397. [CrossRef]

50. Benlioğlu, B. Determination of responses of some barley cultivars (*Hordeum vulgare* L.) to salt stress in different doses at the germination period. *J. Cent. Res. Inst. Field Crops* **2015**, *24*, 109–114.
51. Wang, J.Y.; Turner, N.C.; Liu, Y.X.; Siddique, K.H.; Xiong, Y.C. Effects of drought stress on morphological, physiological and biochemical characteristics of wheat species differing in ploidy level. *Funct. Plant Biol.* **2016**, *44*, 219–234. [CrossRef]
52. Zencirci, N.; Ulukan, H.; Bülent, O.; Aslan, D.; Mutlu, H.T.; Örguç, M. Salt, cold, and drought stress on einkorn and bread wheat during germination. *Int. J. Second. Metab.* **2019**, *6*, 113–128. [CrossRef]
53. Ozaktan, H.; Doymaz, A. Mineral composition and technological and morphological performance of beans as influenced by organic seaweed-extracted fertilizers applied in different growth stages. *J. Food Compos. Anal.* **2022**, *114*, 104741. [CrossRef]
54. Wang, J.Y.; Xiong, Y.C.; Li, F.M.; Siddique, K.H.; Turner, N.C. Effects of drought stress on morphophysiological traits, biochemical characteristics, yield, and yield components in different ploidy wheat: A meta-analysis. *Adv. Agron.* **2017**, *143*, 139–173.
55. Aasim, M.; Katurcı, R.; Akgur, O.; Yildirim, B.; Mustafa, Z.; Nadeem, M.A.; Baloch, F.S.; Karakoy, T.; Yılmaz, G. Machine learning (ML) algorithms and artificial neural network for optimizing in vitro germination and growth indices of industrial hemp (*Cannabis sativa* L.). *Ind. Crops Prod.* **2022**, *181*, 114801. [CrossRef]
56. Jafari, M.; Daneshvar, M.H.; Jafari, S.; Hesami, M. Machine learning-assisted in vitro rooting optimization in *Passiflora caerulea*. *Forests* **2022**, *13*, 2020. [CrossRef]
57. Kirtis, A.; Aasim, M.; Katurcı, R. Application of artificial neural network and machine learning algorithms for modeling the in vitro regeneration of chickpea (*Cicer arietinum* L.). *Plant Cell Tissue Organ Cult.* **2022**, *150*, 141–152. [CrossRef]
58. Pepe, M.; Hesami, M.; Small, F.; Jones, A.M.P. Comparative analysis of machine learning and evolutionary optimization algorithms for precision micropropagation of *Cannabis sativa*: Prediction and validation of in vitro shoot growth and development based on the optimization of light and carbohydrate sources. *Front. Plant Sci.* **2021**, *12*, 757869.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.



Article

A Lightweight Detection Method for Blueberry Fruit Maturity Based on an Improved YOLOv5 Algorithm

Feng Xiao, Haibin Wang *, Yueqin Xu and Zhen Shi

College of Mechanical and Electrical Engineering, Northeast Forestry University, Harbin 150040, China; xiaofeng@nefu.edu.cn (F.X.)

* Correspondence: whb_nefu@nefu.edu.cn

Abstract: In order to achieve accurate, fast, and robust recognition of blueberry fruit maturity stages for edge devices such as orchard inspection robots, this research proposes a lightweight detection method based on an improved YOLOv5 algorithm. In the improved YOLOv5 algorithm, the ShuffleNet module is used to achieve lightweight deep-convolutional neural networks. The Convolutional Block Attention Module (CBAM) is also used to enhance the feature fusion capability of lightweight deep-convolutional neural networks. The effectiveness of this method is evaluated using the blueberry fruit dataset. The experimental results demonstrate that this method can effectively detect blueberry fruits and recognize their maturity stages in orchard environments. The average recall (R) of the detection is 92.0%. The mean average precision (mAP) of the detection at a threshold of 0.5 is 91.5%. The average speed of the detection is 67.1 frames per second (fps). Compared to other detection algorithms, such as YOLOv5, SSD, and Faster R-CNN, this method has a smaller model size, smaller network parameters, lower memory usage, lower computation usage, and faster detection speed while maintaining high detection performance. It is more suitable for migration and deployment on edge devices. This research can serve as a reference for the development of fruit detection systems for intelligent orchard devices.

Citation: Xiao, F.; Wang, H.; Xu, Y.; Shi, Z. A Lightweight Detection Method for Blueberry Fruit Maturity Based on an Improved YOLOv5 Algorithm. *Agriculture* **2024**, *14*, 36. <https://doi.org/10.3390/agriculture14010036>

Academic Editors:
Gniewko Niedbala,
Sebastian Kujawa, Tomasz
Wojciechowski and Magdalena
Piekutowska

Received: 21 November 2023
Revised: 20 December 2023
Accepted: 20 December 2023
Published: 24 December 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: blueberry fruit; deep learning; machine vision; object detection; YOLOv5

1. Introduction

Blueberry fruits are rich in many nutrients, particularly vitamin C, vitamin K, and manganese. In addition, they are rich in fiber, anthocyanins, antioxidants, and other bioactive compounds. These ingredients provide blueberry fruits with a variety of health benefits, including the ability to enhance immune function and improve cardiovascular health. Figure 1 shows blueberry plants and their fruits. As people pay increasing attention to their health, the demand for blueberry fruits is also rising. Consequently, the commercial cultivation areas for blueberry fruits are expanding [1].

Blueberry fruit harvesting directly impacts the yield and income generated from blueberry cultivation. Blueberry fruits have a short ripening period, which typically occurs during the rainy season. The contradiction between the high demand for labor and the shortage of labor in the traditional blueberry production modes has become increasingly evident. The labor cost of blueberry harvesting has reached 30–50% of the total production cost [2]. To some extent, it has hindered the development of the blueberry industry [3] and agricultural production. Developing precision agriculture and smart agriculture technology is an important step in addressing the labor-intensive nature of the blueberry industry [4–6]. Accurately, quickly, and robustly detecting blueberry fruits and providing information on their maturity stages are essential requirements for efficient and timely harvesting. Currently, researchers have conducted extensive studies on fruit detection and recognition using deep learning (DL) technology [7–9]. Based on the YOLOv2 algorithm, Xiong J. et al. [10] detected green mangoes on the surface of tree crowns. They

achieved a precision of 96.1% and a recall of 89.0%. Based on the YOLOv3 algorithm, Zhang W. et al. [11] counted the number of citrus fruits in video sequences. They resolved the issue of double-counting fruit in overlapping situations. Based on the YOLOv4 algorithm, Gao F. et al. [12] detected and counted apples by tracking their stems. They achieved a mean average precision of 99.35% for fruits and trunks. Miao et al. [13] integrated classical image processing techniques with the YOLOv5 algorithm. The detection and recognition performance of the YOLOv5 algorithm has been significantly improved. In addition, Yu Y. et al. [14] detected strawberries using the Mask R-CNN algorithm, while Jia W. et al. [15] detected apples.



Figure 1. Blueberry plants and their fruits in an orchard.

In order to achieve accurate, fast, and robust recognition of blueberry fruit maturity stages for orchard inspection robots while also enhancing the deployment capability of the blueberry fruit detection algorithm on edge devices such as agricultural unmanned aerial vehicles (UAVs) and agricultural unmanned ground vehicles (UGVs), this research proposes a lightweight detection method based on an improved YOLOv5 algorithm. It uses 680 images of blueberry fruits, including 9935 target blueberry fruits, to evaluate the effectiveness of the improved YOLOv5 algorithm and compares its performance with other advanced fruit detection algorithms, such as YOLOv5, SSD, and Faster R-CNN.

Section 2 introduces the production of the blueberry fruit dataset, the YOLOv5 algorithm, and the improved YOLOv5 algorithm. Section 3 demonstrates the detection performance of the improved YOLOv5 algorithm. To investigate the impact of each module changed in the improved YOLOv5 algorithm, Section 3 also conducted a comprehensive ablation study. Section 4 concludes the paper.

2. Materials and Methods

2.1. Dataset Production

2.1.1. Data Acquisition

The blueberry fruit dataset, which serves as the signal source guiding deep convolutional neural networks for blueberry fruit detection, plays a significant role in determining the overall performance of algorithms. The blueberry fruit images in the dataset came from two sources. Some were collected from the published dataset in Reference [5]. The others were collected from an orchard by the camera of an iPhone 11 set to fully automatic mode. The orchard is located in the Jizhou District, Tianjin, China.

Considering the planting spacing of blueberry plants in the orchard, the photographers positioned themselves at a distance of 0.4–0.9 m from the blueberry plants. The distance between the camera and the blueberry plants is 0.2–0.5 m, and the camera is approximately 1.2 m above the ground. The three shooting distances of near, medium, and far correspond to 0.2–0.3 m, 0.3–0.4 m, and 0.4–0.5 m, respectively, from the camera to the blueberry plants. Examples of blueberry fruit images at the three shooting distances are shown in Figure 2.

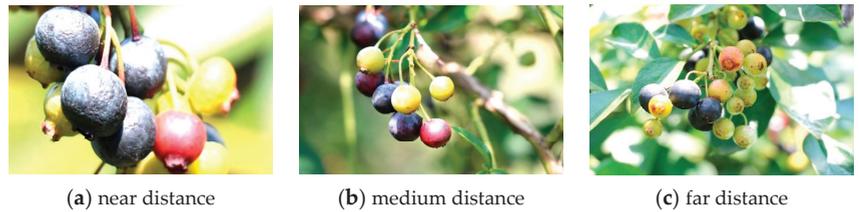


Figure 2. Examples of blueberry fruit images at the three shooting distances.

The working environment of edge devices in orchards is also affected by external conditions, such as the presence of branches and leaves. As shown in Figure 3, the blueberry fruit images in the dataset can be classified into various types. These types include mild clustering, severe clustering, mild occlusion, severe occlusion, backlighting, and blurred background. In addition, the blueberry fruit images in the dataset include different time periods (such as morning and afternoon) and various weather conditions (such as sunny and cloudy).

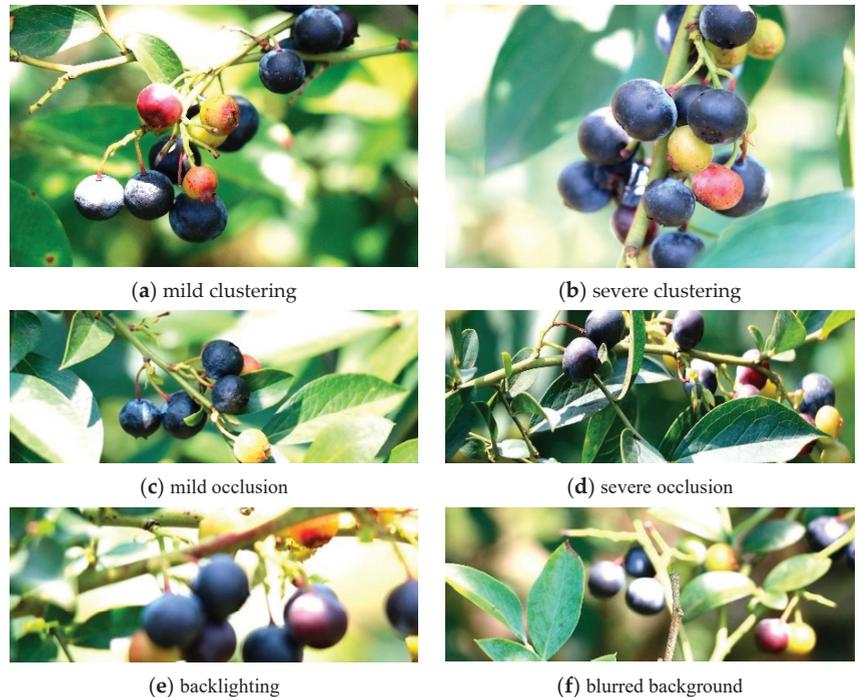


Figure 3. Examples of blueberry fruit images in different conditions.

2.1.2. Data Preprocessing

The ripeness of blueberry fruits is mainly judged based on their color. According to the sensory evaluation method and the expertise of blueberry cultivation experts, blueberry fruits typically go through three stages of ripening: fully ripe, semi-ripe, and immature. The color of immature blueberry fruits is similar to that of the branches and leaves of blueberry plants, while the color of fully ripe blueberry fruits is similar to that of the soil. Labellmg is a graphical image annotation tool. It was created by Tzutalin at National Taiwan University with the help of dozens of contributors. In this research, Labellmg is used to label the original blueberry fruit images according to the labeling format of the

Pascal VOC dataset and generate .xml-type labeling files. Fully ripe blueberry fruits are labeled as “blueberry”, semi-ripe blueberry fruits are labeled as “blueberry-halfripe”, and immature blueberry fruits are labeled as “blueberry-unripe”. When labeling, the bounding box used is the smallest rectangle that completely encloses the entire blueberry fruit. This is done to minimize the inclusion of unnecessary pixels from the background.

Training deep convolutional neural networks requires a large amount of data [16]. Too little data can result in underfitting or overfitting of deep convolutional neural networks. Therefore, data augmentation needs to be performed on the original blueberry fruit images. As shown in Figure 4, this research employs various methods, including mirroring, rotation, scaling, adding noise, and adjusting brightness, to enhance the diversity of the blueberry fruit images that we have collected. The annotation files corresponding to each image are transformed simultaneously. The augmented blueberry fruit dataset contains 680 images and can improve the robustness of deep convolutional neural networks in detecting the ripeness of blueberry fruits in orchards.



Figure 4. Data augmentation.

The blueberry fruit dataset is randomly divided into a training set and a validation set [17]. The data distribution is shown in Table 1. The training set contains 544 images of blueberry fruits, including 7895 target blueberry fruits. Among these, there are 4310 fully mature blueberry fruits, 655 semi-ripe blueberry fruits, and 2930 immature blueberry fruits. The validation set contains 136 images of blueberry fruits, including 2040 target blueberry fruits. Among these images, there are 1169 fully mature blueberry fruits, 172 semi-ripe blueberry fruits, and 699 immature blueberry fruits.

Table 1. Data distribution.

Datasets	Number of Blueberry Fruit Images	Number of Target Blueberry Fruits		
		Total	Types	Number
blueberry fruit dataset	680	9935	mature	5479
			semi-ripe	827
			immature	3629
training set	544	7895	mature	4310
			semi-ripe	655
			immature	2930
validation set	136	2040	mature	1169
			semi-ripe	172
			immature	699

2.2. The YOLOv5 Algorithm

The YOLOv5 algorithm is one of the algorithms in the YOLO series [18]. It is an improvement based on the YOLOv4 algorithm. The YOLOv5 algorithm consists of 10 detectors: YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x, and others. The main differences among them lie in the number of convolutional layers and optimal application scenarios. As the number of convolutional layers increases, the model size gradually increases, while the detection performance improves and the detection speed decreases. This research focuses on the YOLOv5s 7.0 model as the subject. As shown in Figure 5, the overall network structure of the YOLOv5s 7.0 model can be divided into four parts: the input layers, the backbone feature extraction networks, the neck enhancement feature extraction networks, and the output layers.

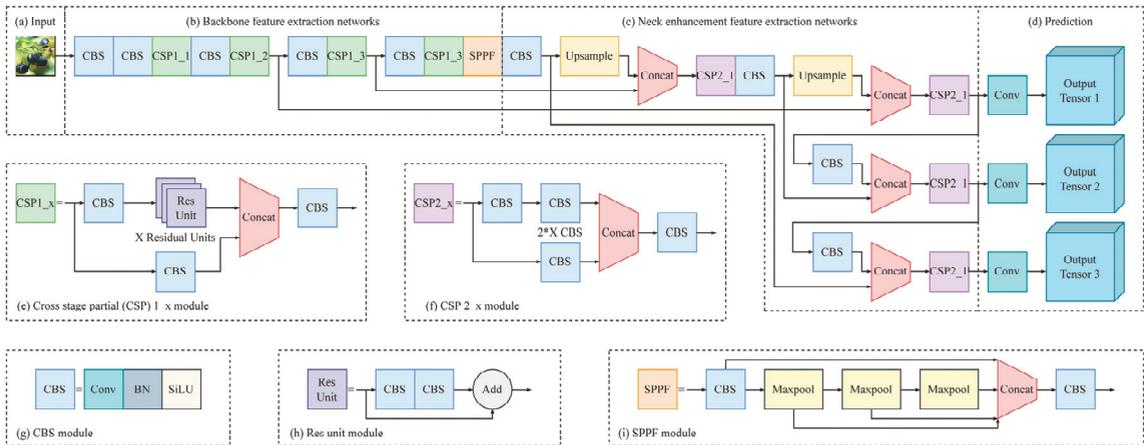


Figure 5. Structure of the YOLOv5 algorithm.

The input layers of the YOLOv5 model include Mosaic data enhancement, adaptive image scaling, and adaptive anchor box calculation [19]. The Mosaic data enhancement operation splices input images using random scaling, random cropping, and random arrangement [20]. It enhances the effectiveness of detecting and recognizing small targets. The adaptive image scaling operation adds minimal black borders to the original images while calculating the scaling ratio, scaled size, and padding value for the black borders. In training deep convolutional neural networks, the optimal anchor box is calculated using the adaptive anchor box calculation operation. Compared to the method of presetting the anchor box length and width, this calculation operation minimizes the difference between the anchor box and the ground truth box. Additionally, through reverse updating, the

parameters of deep convolutional neural networks are optimized, resulting in improved performance for target detection and recognition.

In the backbone feature extraction networks of the YOLOv5 model, the CBS module is responsible for extracting features from images and organizing feature maps. It involves enhancing dimensionality, reducing dimensionality, downsampling, and normalization for feature maps. As shown in Figure 5g, a CBS module consists of a Conv2d function, a BatchNorm2d function, and a SiLU activation function [21,22]. The padding of the Conv2d function in the CBS module is automatically calculated. The stride of the Conv2d function in the CBS module is set to 2, and the kernel size is set to 3. Therefore, during the process of downsampling the feature map to extract the target features, the CBS module reduces the width and height of the feature map by half each time. The BatchNorm2d function is a layer for batch normalization, which normalizes the input data in each batch. The expressions for the SiLU activation function and its derivative are as follows:

$$\text{SiLU}(x) = f(x) = x \cdot \text{sigmoid}(x) = x / (1 + e^{-x}) \quad (1)$$

$$\text{SiLU}'(x) = f'(x) = x \cdot f(x) \cdot (1 - f(x)) + f(x) \quad (2)$$

where x represents the variable, $\text{SiLU}(x)$ represents the SiLU activation function, and $\text{SiLU}'(x)$ represents the derivative of the SiLU activation function. The graph of the SiLU activation function and its derivative is shown in Figure 6. The SiLU activation function has the characteristics of being unbounded, smooth, and non-monotonic. It increases the nonlinearity of the input data. To be specific, the SiLU activation function remains continuously differentiable as x approaches 0, which helps to avoid the vanishing gradient problem of deep convolutional neural networks. The SiLU activation function has a weak negative response at $x < 0$, which promotes the generalization ability of deep convolutional neural networks.

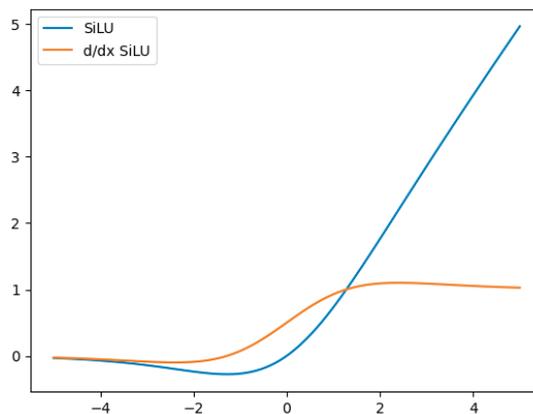


Figure 6. Graph of the SiLU activation function and its derivative.

As convolution operations progress, the deep convolutional neural networks extract increasingly complex feature information. In the shallow layers of the deep convolutional neural networks, the networks extract relatively simple feature information, such as color, shape, and texture. These features are visible in graphics, so they are referred to as graphical features. Next, the deeper convolutional neural networks fuse these graphical features and expand their dimensions to create new features. The new features are referred to as semantic features. Graphical features are simplistic and lack semantic depth, while semantic features provide more comprehensive information but may overlook basic visual elements.

In the neck enhancement feature extraction networks of the YOLOv5 model, the structure of the Feature Pyramid Network (FPN) and Path Aggregation Network (PANet) is

utilized to combine shallow graphical features with deep semantic features [23], as depicted in Figure 7. Specifically, the FPN fuses low-level features by upsampling them to the top-level features and makes predictions on each fused feature layer. This approach combines the advantages of low-level and high-level features, effectively improving the performance of detecting and recognizing small targets. However, while the FPN effectively transmits semantic features from top to bottom, it does not transmit positional information. Therefore, the PANet is introduced. The PANet includes a bottom-up path augmentation structure based on the FPN. This structure utilizes shallow features in deep convolutional neural networks. The top feature maps can benefit from the abundant positional information provided by the lower layers. This improves the performance of detecting medium- and large-sized targets. The structure of the FPN and PANet effectively enhances the detection and recognition capabilities of deep convolutional neural networks for fruits of various types and sizes.

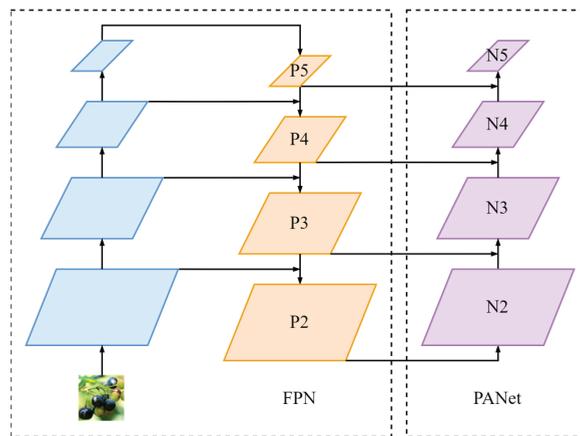


Figure 7. Structures of the FPN and PANet. (P2, P3, P4, and P5 indicate the feature layers generated by the FPN. N2, N3, N4, and N5 represent the feature layers generated by the PANet.)

The loss functions of the output layers of the YOLOv5 model consist of three components. The Generalized Intersection over Union (GIoU) loss function is used to calculate the loss for boundary regression [24]. The YOLOv5 model performs weighted Non-Maximum Suppression (NMS) on GIoU_Loss to achieve the efficient selection of the optimal bounding box. The Binary Cross Entropy with Logits (BCEWithLogits) loss function is used to calculate the loss for confidence prediction, while the Binary Cross Entropy loss (BCELoss) function is used to calculate the loss for class prediction.

2.3. The Improved YOLOv5 Algorithm

2.3.1. The ShuffleNet Module

In recent years, there have been some proposals for lightweight structures of deep convolutional neural networks, such as MobileNet and ShuffleNet. The ShuffleNetV1 is a lightweight deep-convolutional neural network proposed by MEGVII [25]. The modules of the ShuffleNetV1 are shown in Figure 8. In order to achieve a good balance between detection accuracy and speed, the ShuffleNetV1 utilizes the group convolution (GConv) operation and the channel shuffle operation.

Standard convolution is a method of densely connecting channels. This means that the feature information from each channel in the feature map of each layer is outputted to each channel of the feature map in the next layer through a convolution operation. The process is shown in Figure 9a. The parameters P for standard convolution are as follows:

$$P = D \times D \times M \times Z \quad (3)$$

where D represents the size of the convolution kernel, and M represents the number of input feature channels. Z represents the number of output feature channels, which is also equal to the number of convolution kernels.

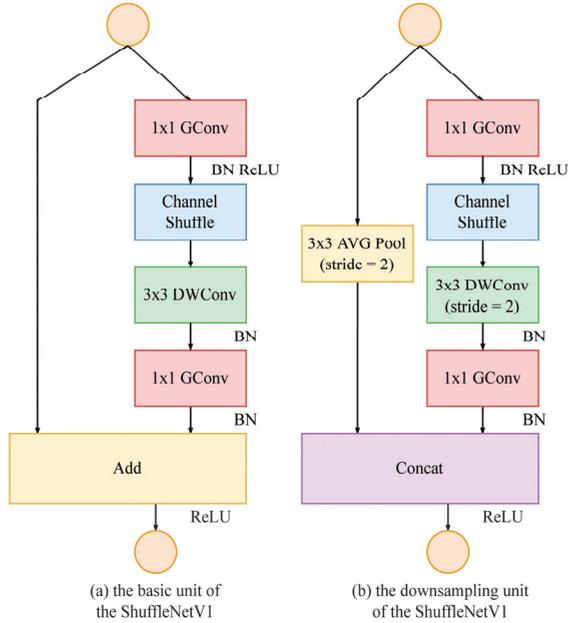


Figure 8. Modules of the ShuffleNetV1.

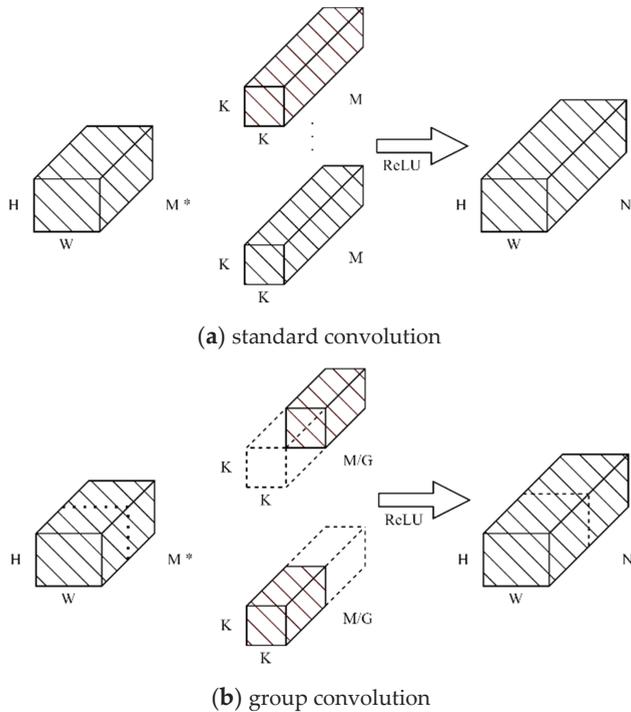


Figure 9. Standard convolution and group convolution.

Group convolution is a method used to connect sparse convolutions. In this method, the input feature channel number is divided into G groups. Each group has M/G feature channels for the convolution kernel. After the convolution operation, each group produces a feature map with an output feature channel number of N . The process is illustrated in Figure 9b. The parameters P_{GC} for group convolution are as follows:

$$P_{GC} = D \times D \times M/G \times Z \tag{4}$$

Comparing Equations (3) and (4), the parameters and calculations for standard convolution are G of the parameters and calculations for group convolution. The group convolution operation, however, has limitations in terms of its ability to exchange information among groups. As shown in Figure 10, the channel shuffle operation reorganizes the feature information of various groups in the output layers to improve communication throughout the feature information of each group. This method enhances the learning capacity of feature information across groups and reduces the computational load of deep convolutional neural networks.

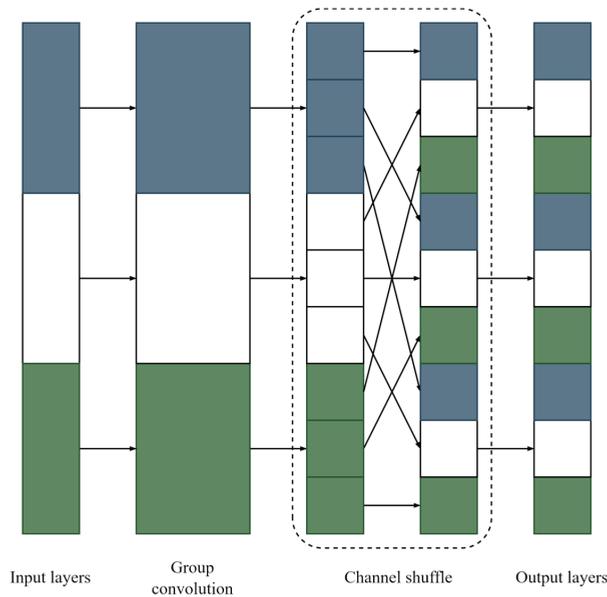


Figure 10. Channel shuffle schematic.

Ma et al. [26] designed the ShuffleNetV2 based on the ShuffleNetV1. As shown in Figure 11, the ShuffleNetV2 primarily consists of the basic unit and the downsampling unit. As shown in Figure 11a, the input feature channels in the basic unit of the ShuffleNetV2 are divided into two branches through the channel split operation. Each branch has an equal number of feature channels. The left branch performs identity mapping [27]. The right branch undergoes two 1×1 ordinary convolutions and one 3×3 depthwise convolution (DWConv) while maintaining an equal number of input and output channels. The left and right branches are merged through the channel concatenation operation, and the channel shuffle operation is performed to ensure the integration of feature information from both branches. The downsampling unit of the ShuffleNetV2, shown in Figure 11b, directly inputs the feature map into two branches. Each branch performs the 1×1 ordinary convolution and 3×3 DWConv with a stride of 2. After merging the two branches using the channel concatenation operation, the number of output channels is doubled. The merged feature map undergoes the channel shuffle operation. Unlike the basic unit, the downsampling

unit directly increases the number of feature channels in the deep convolutional neural networks and expands their width without using the channel split operation. This approach avoids increasing the computational burden of the deep convolutional neural networks while further enhancing their feature extraction capabilities.

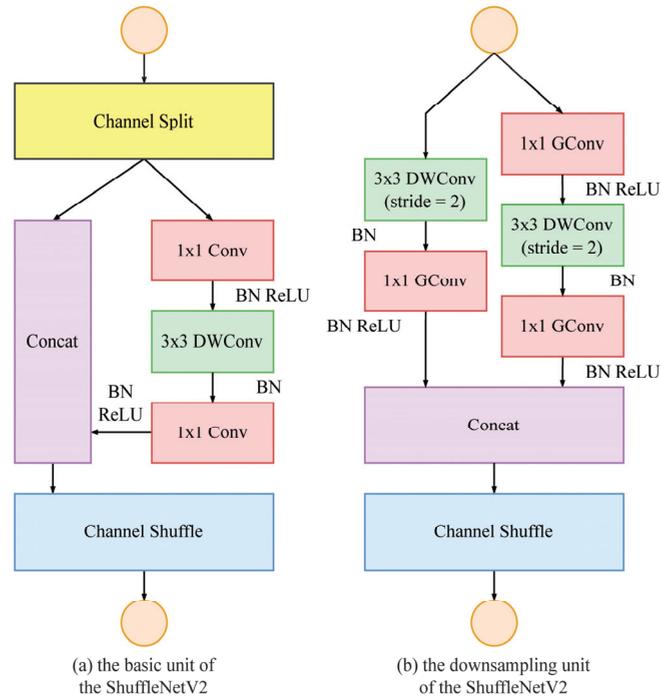


Figure 11. Modules of the ShuffleNetV2.

Under the same complexity, the ShuffleNetV2 outperforms the ShuffleNetV1, MobileNetV1, MobileNetV2, and other lightweight deep convolutional neural networks. The improved YOLOv5 algorithm was built upon the ShuffleNetV2, utilizing the SFB1_X and SFB2_X architectures. In the improved YOLOv5 algorithm, the backbone networks of the ShuffleNetV2 have replaced the 1024 convolution operation and the 5×5 pooling operation with the global average pooling operation. This change aims to improve the speed of detection and recognition by reducing the memory usage of the deep convolutional neural networks.

2.3.2. The CABM Module

Attention mechanisms originated from the study of human vision [28]. In computer vision (CV), attention mechanisms are used to process visual information. Traditional methods, such as the local feature extraction approach and the sliding window approach, can be considered as types of attention mechanisms. In DL, attention mechanisms are typically implemented as separate attention modules. Attention modules allow deep neural networks to assign different weights to different parts of the inputs, enabling them to focus on relevant units and suppress irrelevant units during the process of feature extraction. The Convolutional Block Attention Module (CBAM) [29], the Efficient Channel Attention Module (ECA) [30], and the Squeeze and Excitation Network (SENet) [31] are common attention mechanisms. The structure of the CBAM module is shown in Figure 12.

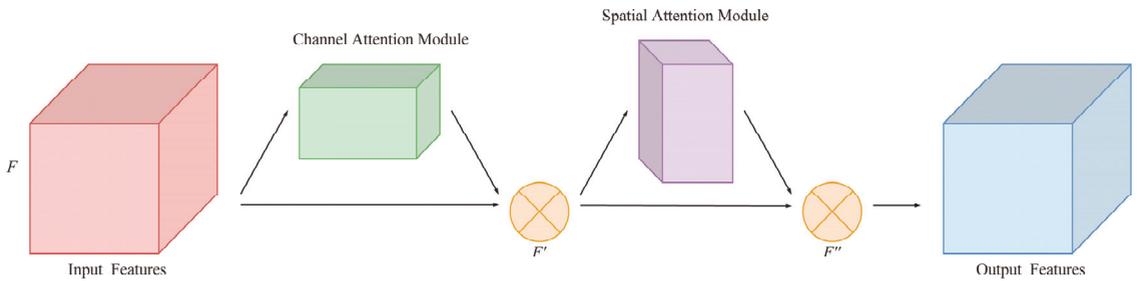


Figure 12. Structure of the CBAM module.

Compared to traditional attention mechanisms that solely focus on channels or spatial dimensions, the CBAM module consists of two parts: the Channel Attention Module (CAM), which focuses on channel information, and the Spatial Attention Module (SAM), which focuses on location information. The CBAM module combines channel information and spatial information to enable the deep convolutional neural networks to focus on important features and suppress interference from less significant ones. From our perspective, this will help address the issue of color similarity between immature blueberry fruits and their backgrounds, as well as occlusions caused by adjacent blueberry fruits and leaves. Therefore, this research used the CBAM module to enhance the performance of blueberry fruit detection and recognition. The overall attention process of the CBAM module used in this research is described by Equations (5) and (6):

$$F' = M_c(F) \otimes F \quad (5)$$

$$F'' = M_s(F') \otimes F' \quad (6)$$

where F represents the original input feature map, F' represents the adjusted feature map using the CAM, F'' represents the final feature map using the SAM, M_c represents the weight matrix after channel compression, M_s represents the weight matrix after spatial compression, and \otimes represents the element-wise multiplication of matrices.

2.3.3. The Improved YOLOv5 Algorithm

The overall network structure of the improved YOLOv5 algorithm is shown in Figure 13. It mainly consists of four parts: the input layers, the backbone feature extraction networks, the neck enhancement feature extraction networks, and the output layers. The input layers of the improved YOLOv5 algorithm accept blueberry fruit images with the size of 608×608 pixels. These images are then passed through the ShuffleNet modules in the backbone feature extraction networks of the improved YOLOv5 algorithm for blueberry fruit feature extraction. Subsequently, the blueberry fruit feature maps are sent to the neck enhanced feature extraction networks of the improved YOLOv5 algorithm for blueberry fruit feature fusion. Finally, the output layers of the improved YOLOv5 algorithm produce three prediction anchor boxes of different scales.

Compared to the YOLOv5 model:

- (1) First, because the SPPF module needs to perform pooling operations at multiple scales and splice the results, it takes up more memory space. This limits the application of network models to resource-constrained devices. In order to achieve lightweight deep convolutional neural networks, the improved YOLOv5 algorithm removes the SPPF module from the backbone feature extraction networks of the YOLOv5 algorithm.
- (2) Second, the CSP Bottleneck module utilizes the multi-channel separated convolution operation. Frequently using the CSP Bottleneck module can consume a significant amount of cache space and decrease the execution speed of deep convolutional neural networks. The ShuffleNet modules with Shuffle channels are used to replace the

CSPDarknet-53 modules in the backbone feature extraction networks of the YOLOv5 algorithm for blueberry fruit feature extraction.

- (3) Finally, the CBAM modules are integrated into the neck enhancement feature extraction networks of the YOLOv5 algorithm to enhance the feature fusion capability of deep convolutional neural networks. This enables the efficient extraction of important features and the suppression of irrelevant ones.

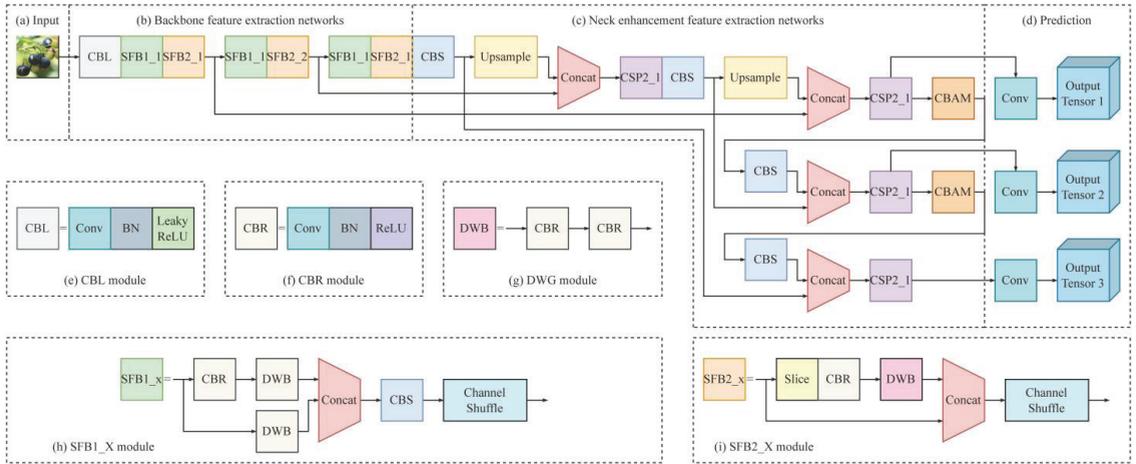


Figure 13. Structure of the improved YOLOv5 algorithm.

3. Results and Discussion

3.1. Experimental Platforms

The hardware platform used for the experiments is as follows: the CPU was a 13th Gen Intel® Core™ i5-13600K with 14 cores and 20 threads. It has a base frequency of 3.50 GHz and a maximum boost frequency of 5.10 GHz. The GPU was an NVIDIA GeForce RTX 3080 with 12 GB of memory and 8960 CUDA cores, which enable the accelerated training of deep convolutional neural networks. The memory consisted of two Hynix DDR5 5600 MHz 16 GB DIMMs (Dual In-Line Memory Modules). The hard disk was a Samsung SSD980 PRO with a capacity of 1 TB. The motherboard was an MSI PRO Z790-A WIFI DDR5.

The software configuration for the experiments is as follows: the operating system is Windows 11. The programming language is Python 3.8. The integrated development environment is PyCharm 2020.3.5. The DL framework is PyTorch 1.11.0. The parallel computer framework is CUDA 11.4.0, and the DL acceleration library is cuDNN 8.2.2.

In addition, this research sets the batch size to 32. The process of going through all the data in the blueberry fruit dataset once is referred to as one epoch, and the improved YOLOv5 algorithm runs for 1600 epochs. The values of the conf-thres parameter and iou-thres parameter are both set to 0.5.

3.2. Evaluation Metrics

To compare the performance of different fruit detection and recognition methods, the evaluation metrics of Precision (P), Recall (R), Average Precision (AP), and mean Average Precision (mAP) are used. The calculation formulas are shown in Equations (7)–(10). P represents the probability that the positive sample is accurately identified as the positive example by the classifier. R represents the classifier’s ability to correctly identify all positive samples. The area enclosed by the P - R curve is represented by AP . The P - R curve is formed with the R as the independent variable and the P as the dependent variable. mAP is the

average AP value across multiple categories [32]. It measures the classifier's ability to effectively detect and recognize all classes.

$$P = \frac{TP}{TP + FP} \quad (7)$$

$$R = \frac{TP}{TP + FN} \quad (8)$$

$$AP = \int_0^1 P(R) dR \quad (9)$$

$$mAP = \frac{1}{N} \cdot \sum_{i=1}^n (AP_i) \quad (10)$$

where TP represents the number of correctly detected blueberry fruits, FP represents the number of erroneously detected blueberry fruits, FN represents the number of missed blueberry fruits, and N represents the number of categories for object detection.

In addition, the complexity, computational efficiency, and real-time performance of the improved YOLOv5 algorithm are evaluated by considering factors such as model size, network parameters, FLOPs, and detection speed.

3.3. Experimental Results

Experiments were conducted to test the detection effect of blueberry fruit ripeness. The P curve, R curve, P - R curve, and $F1$ curve of the improved YOLOv5 algorithm are shown in Figure 14. For the category of fully mature blueberry fruits with abundant samples, the $mAP@0.5$ are the highest, and there are few false detections and missed detections. The category with the second-highest $mAP@0.5$ is the immature blueberry fruit category, while the semi-ripe blueberry fruit category has the lowest $mAP@0.5$.

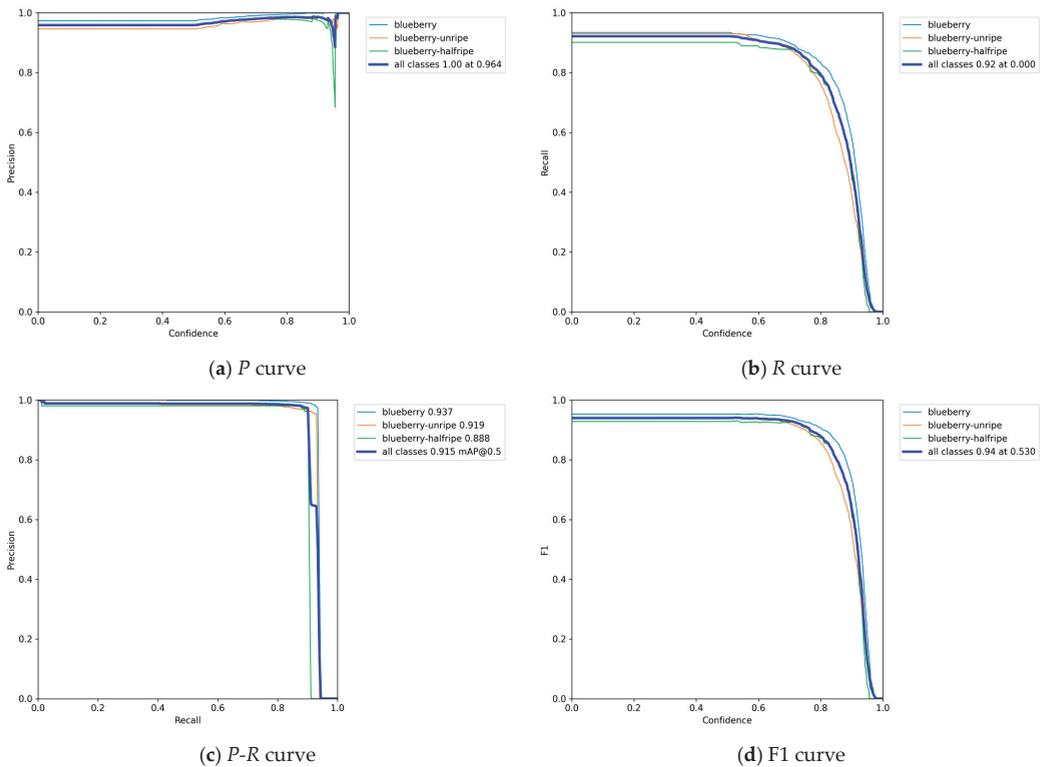


Figure 14. Experiment results of the improved YOLOv5 algorithm.

The examples of the detection effects of the improved YOLOv5 algorithm are shown in Figure 15. As shown in Figure 15a, the improved YOLOv5 algorithm demonstrates excellent detection results for blueberry fruits in close-range and medium-range images. It effectually detects blueberry fruits and recognizes their ripeness. As shown in Figure 15b, for blueberry fruits that are located at a greater distance, there is a higher likelihood of both false detection and missed detection. As shown in Figure 15c,d, the improved YOLOv5 algorithm can also effectually detect blueberry fruits, even in situations involving mild clustering, mild occlusion, backlighting, and blurred background. However, as shown in Figure 15b, the improved YOLOv5 algorithm may face challenges in effectually detecting blueberry fruits in scenarios with severe clustering and severe occlusion. This is because these instances do not provide enough distinctive feature information for classification.

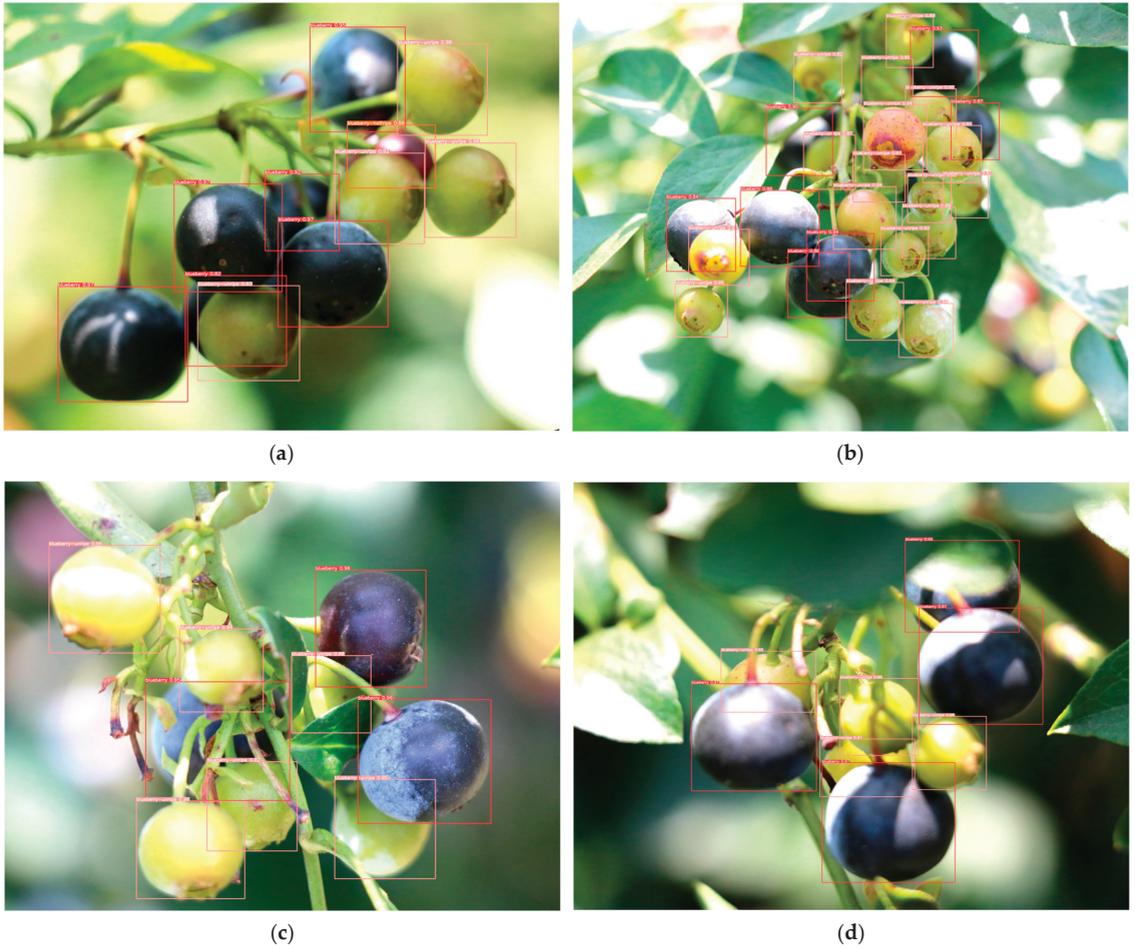


Figure 15. Examples of the detection effects of the improved YOLOv5 algorithm.

Generally speaking, the improved YOLOv5 algorithm achieves a P of 96.3%, a R of 92%, and a mAP of 91.5% at a threshold of 0.5. The average detection speed of the improved YOLOv5 algorithm is 67.1 fps with a batch size of 1 on the NVIDIA GeForce RTX 3080. The improved YOLOv5 algorithm has a 5.65 MB model size, 2.85 M network parameters, and 5.6 G FLOPs. It is suitable for migration and deployment on edge devices such as agricultural UAVs and agricultural UGVs.

3.4. Performance Comparison

This research also trains the YOLOv5, SSD, and Faster R-CNN algorithms using the blueberry fruit dataset. Their performance is then compared to the improved YOLOv5 algorithm. The performance of various blueberry fruit detection algorithms is presented in Table 2.

Table 2. Performance comparison of the various blueberry fruit detection algorithms.

Metrics/Models		YOLOv5	YOLOv5-Ours	SSD-vgg	Faster R-CNN-vgg
<i>P</i> (%)	mature	98.7	97.8	96.0	93.1
	semi-ripe	95.5	96.3	92.7	87.1
	immature	97.0	94.9	96.2	85.6
	mean value	97.1	96.3	95.0	88.6
<i>R</i> (%)	mature	93.5	92.9	96.0	95.8
	semi-ripe	91.3	90.1	89.0	90.1
	immature	93.4	93.0	93.9	93.0
	mean value	92.7	92.0	93.0	93.0
<i>mAP@0.5</i> (%)	mature	95.1	93.7	95.9	95.6
	semi-ripe	91.0	88.8	88.0	89.1
	immature	93.5	91.9	92.5	91.0
	mean value	93.2	91.5	92.1	91.9
Model size (MB)		13.6	5.65	91.6	521.0
Parameter (M)		7.02	2.85	23.6	136.7
FLOPs (G)		15.8	5.6	246.6	376.5
Speed (fps)		66.2	67.1	44.4	17.0

As shown in Table 2, when compared to the YOLOv5 algorithm, the improved YOLOv5 algorithm exhibits a 0.8% decrease in precision, a 0.7% decrease in recall, and a 1.7% decrease in *mAP@0.5*. However, the model size, network parameter, and FLOPs of the improved YOLOv5 algorithm decrease by 7.95 MB, 4.17 M, and 10.2 G, respectively. The average detection speed of the improved YOLOv5 algorithm increases 0.9 fps.

Compared to the SSD-vgg, the improved YOLOv5 algorithm exhibits a 1.0% decrease in recall and a 0.6% decrease in *mAP@0.5*. However, the model size, network parameter, and FLOPs of the improved YOLOv5 algorithm decrease by 85.95 MB, 20.75 M, and 241.0 G, respectively. The precision and average detection speed of the improved YOLOv5 algorithm increase by 1.3% and 22.7 fps, respectively.

Compared to the Faster R-CNN-vgg, the improved YOLOv5 algorithm exhibits a 1.0% decrease in recall and a 0.4% decrease in *mAP@0.5*. However, the model size, network parameter, and FLOPs of the improved YOLOv5 algorithm decrease by 515.35 MB, 133.85 M, and 370.9 G, respectively. The precision and average detection speed of the improved YOLOv5 algorithm increase by 7.7% and 50.1 fps, respectively.

Generally speaking, when compared to the YOLOv5, SSD, and Faster R-CNN, the improved YOLOv5 algorithm has a smaller model size, smaller network parameters, lower memory usage, lower computation usage, and faster detection speed while maintaining high detection performance. It is more suitable for migration and deployment on edge devices.

For the purpose of testing the impact of each module changed in the improved YOLOv5 algorithm, this research conducted a comprehensive ablation study. The YOLOv5-ShuffleNet algorithm only replaced the CSPDarknet-53 module in the backbone feature extraction networks of the YOLOv5 algorithm with the ShuffleNetv2 module. The YOLOv5-CBAM algorithm only integrated the CBAM module into the neck enhancement feature extraction networks of the YOLOv5 algorithm. Table 3 presents the performance comparison among the YOLOv5, YOLOv5-ShuffleNet, YOLOv5-CBAM, and YOLOv5-ShuffleNet-CBAM algorithms.

Table 3. Results of the ablation study.

Metrics/Models		YOLOv5	YOLOv5-ShuffleNet	YOLOv5-CBAM	YOLOv5-ShuffleNet-CBAM
P (%)	mature	98.7	97.8	98.8	97.8
	semi-ripe	95.5	94.5	97.5	96.3
	immature	97.0	95.9	97.1	94.9
	mean value	97.1	96.1	97.8	96.3
R (%)	mature	93.5	90.8	96.1	92.9
	semi-ripe	91.3	89.5	90.6	90.1
	immature	93.4	88.3	95.1	93.0
	mean value	92.7	89.5	93.9	92.0
mAP@0.5 (%)	mature	95.1	91.6	96.5	93.7
	semi-ripe	91.0	88.8	94.0	88.8
	immature	93.5	87.2	90.4	91.9
	mean value	93.2	89.2	93.6	91.5
Model size (MB)		13.6	2.8	13.6	5.65
Parameter (M)		7.02	2.84	7.02	2.85
FLOPs (G)		15.8	5.5	15.8	5.6
Speed (fps)		66.2	77.0	57.1	67.1

As shown in Table 3, when compared to the YOLOv5 algorithm, the YOLOv5-ShuffleNet algorithm exhibits a reduction in model size, network parameters, and FLOPs by 10.8 MB, 4.18 M, and 10.3 G, respectively. The detection speed increases by 11.2 fps. When compared to the YOLOv5-CBAM algorithm, the YOLOv5-ShuffleNet-CBAM algorithm exhibits a reduction in model size, network parameters, and FLOPs by 7.95 MB, 4.17 M, and 10.2 G, respectively. The detection speed increases by 10 fps. The experimental results demonstrate that the ShuffleNet module can effectively achieve lightweight deep convolutional neural networks.

When compared to the YOLOv5 algorithm, the YOLOv5-CBAM algorithm exhibits an increase of 0.7% in precision, a 1.2% increase in recall, and a 0.4% increase in $mAP@0.5$. When compared to the YOLOv5-ShuffleNet algorithm, the YOLOv5-ShuffleNet-CBAM algorithm demonstrates an improvement of 0.2% in precision, a 2.5% improvement in recall, and a 2.3% improvement in $mAP@0.5$. The experimental results demonstrate that the CBAM module can effectively enhance the feature extraction capability of deep convolutional neural networks.

4. Conclusions

Because most fruits mature in batches during their growth process, assessing fruit maturity is an important step in intelligent orchard management. Effective detection and statistics of fruit maturity are beneficial for planning fruit harvests and estimating fruit yields.

- (1) This research proposes a lightweight detection method based on an improved YOLOv5 algorithm. First, in order to achieve lightweight deep convolutional neural networks, the improved YOLOv5 algorithm removes the SPPF module from the backbone feature extraction networks of the YOLOv5 algorithm. The ShuffleNet modules with Shuffle channels are used to replace the CSPDarknet-53 modules in the backbone feature extraction networks of the YOLOv5 algorithm for blueberry fruit feature extraction. Second, the CBAM modules are integrated into the neck enhancement feature extraction networks of the YOLOv5 algorithm to enhance the feature fusion capability of lightweight deep convolutional neural networks.
- (2) The experimental results demonstrate that the improved YOLOv5 algorithm can effectively utilize RGB images to detect blueberry fruits and recognize their ripeness. The improved YOLOv5 algorithm achieves a P of 96.3%, an R of 92%, and a mAP of 91.5% at a threshold of 0.5. The average detection speed of the improved YOLOv5 algorithm is 67.1 fps with a batch size of 1 on the NVIDIA GeForce RTX 3080. The

improved YOLOv5 algorithm has a 5.65 MB model size, 2.85 M network parameters, and 5.6 G FLOPs. Compared to other detection algorithms such as YOLOv5, SSD, and Faster R-CNN, this method has a smaller model size, smaller network parameters, lower memory usage, lower computation usage, and faster detection speed while maintaining high detection performance.

Future research will explore more efficient and lightweight feature extraction modules for deep convolutional neural networks. This will enable the network model to better extract the intricate and variable characteristics of blueberry fruits.

Author Contributions: Conceptualization, F.X., H.W., Y.X. and Z.S.; data curation, F.X., H.W., Y.X. and Z.S.; formal analysis, F.X., H.W., Y.X. and Z.S.; funding acquisition, H.W.; investigation, F.X., H.W., Y.X. and Z.S.; methodology, F.X., H.W., Y.X. and Z.S.; project administration, F.X. and H.W.; resources, F.X., H.W., Y.X. and Z.S.; software, F.X. and Y.X.; supervision, H.W.; validation, Y.X. and Z.S.; visualization, F.X., Y.X. and Z.S.; writing—original draft, F.X.; writing—review and editing, F.X. and H.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Natural Science Foundation of Heilongjiang Province of China (LH2020C047) and the China Postdoctoral Science Foundation (2019T120248).

Institutional Review Board Statement: Not applicable.

Data Availability Statement: The data presented in this study can be requested from the corresponding author. The data is not currently available for public access because it is part of an ongoing research project.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Krishna, P.; Pandey, G.; Thomas, R.; Parks, S. Improving Blueberry Fruit Nutritional Quality through Physiological and Genetic Interventions: A Review of Current Research and Future Directions. *Antioxidants* **2023**, *12*, 810. [CrossRef] [PubMed]
2. Xiao, F.; Wang, H.; Li, Y.; Cao, Y.; Lv, X.; Xu, G. Object Detection and Recognition Techniques Based on Digital Image Processing and Traditional Machine Learning for Fruit and Vegetable Harvesting Robots: An Overview and Review. *Agronomy* **2023**, *13*, 639. [CrossRef]
3. Wang, H.; Lv, X.; Xiao, F.; Sun, L. Analysis and Testing of Rigid–Flexible Coupling Collision Harvesting Processes in Blueberry Plants. *Agriculture* **2022**, *12*, 1900. [CrossRef]
4. Obsie, E.Y.; Qu, H.; Zhang, Y.J.; Annis, S.; Drummond, F. Yolov5s-CA: An Improved Yolov5 Based on the Attention Mechanism for Mummy Berry Disease Detection. *Agriculture* **2023**, *13*, 78. [CrossRef]
5. Yang, W.; Ma, X.; Hu, W.; Tang, P. Lightweight Blueberry Fruit Recognition Based on Multi-Scale and Attention Fusion NCBAM. *Agronomy* **2022**, *12*, 2354. [CrossRef]
6. Yang, W.; Ma, X.; An, H. Blueberry Ripeness Detection Model Based on Enhanced Detail Feature and Content-Aware Reassembly. *Agronomy* **2023**, *13*, 1613. [CrossRef]
7. Wang, H.; Feng, J.; Yin, H. Improved Method for Apple Fruit Target Detection Based on YOLOv5s. *Agriculture* **2023**, *13*, 2167. [CrossRef]
8. Gu, B.; Wen, C.; Liu, X.; Hou, Y.; Hu, Y.; Su, H. Improved YOLOv7-Tiny Complex Environment Citrus Detection Based on Lightweighting. *Agronomy* **2023**, *13*, 2667. [CrossRef]
9. Ren, R.; Sun, H.; Zhang, S.; Wang, N.; Lu, X.; Jing, J.; Xin, M.; Cui, T. Intelligent Detection of Lightweight “Yuluxiang” Pear in Non-Structural Environment Based on YOLO-GEW. *Agronomy* **2023**, *13*, 2418. [CrossRef]
10. Xiong, J.; Liu, Z.; Chen, S.; Liu, B.; Zheng, Z.; Zhong, Z.; Yang, Z.; Peng, H. Visual Detection of Green Mangoes by an Unmanned Aerial Vehicle in Orchards Based on a Deep Learning Method. *Biosyst. Eng.* **2020**, *194*, 261–272. [CrossRef]
11. Zhang, W.; Wang, J.; Liu, Y.; Chen, K.; Li, H.; Duan, Y.; Wu, W.; Shi, Y.; Guo, W. Deep-Learning-Based in-Field Citrus Fruit Detection and Tracking. *Hortic. Res.* **2022**, *9*, uhac003. [CrossRef] [PubMed]
12. Gao, F.; Fang, W.; Sun, X.; Wu, Z.; Zhao, G.; Li, G.; Li, R.; Fu, L.; Zhang, Q. A Novel Apple Fruit Detection and Counting Methodology Based on Deep Learning and Trunk Tracking in Modern Orchard. *Comput. Electron. Agric.* **2022**, *197*, 107000. [CrossRef]
13. Miao, Z.; Yu, X.; Li, N.; Zhang, Z.; He, C.; Li, Z.; Deng, C.; Sun, T. Efficient Tomato Harvesting Robot Based on Image Processing and Deep Learning. *Precis. Agric.* **2023**, *24*, 254–287. [CrossRef]
14. Yu, Y.; Zhang, K.; Yang, L.; Zhang, D. Fruit Detection for Strawberry Harvesting Robot in Non-Structural Environment Based on Mask-RCNN. *Comput. Electron. Agric.* **2019**, *163*, 104846. [CrossRef]
15. Jia, W.; Tian, Y.; Luo, R.; Zhang, Z.; Lian, J.; Zheng, Y. Detection and Segmentation of Overlapped Fruits Based on Optimized Mask R-CNN Application in Apple Harvesting Robot. *Comput. Electron. Agric.* **2020**, *172*, 105380. [CrossRef]

16. Li, J.; Zhou, H.; Jayas, D.S.; Jia, Q. Construction of a Dataset of Stored-Grain Insects Images for Intelligent Monitoring. *Appl. Eng. Agric.* **2019**, *35*, 647–655. [CrossRef]
17. Xiong, Z.; Wang, L.; Zhao, Y.; Lan, Y. Precision Detection of Dense Litchi Fruit in UAV Images Based on Improved YOLOv5 Model. *Remote Sens.* **2023**, *15*, 4017. [CrossRef]
18. Cai, D.; Lu, Z.; Fan, X.; Ding, W.; Li, B. Improved YOLOv4-Tiny Target Detection Method Based on Adaptive Self-Order Piecewise Enhancement and Multiscale Feature Optimization. *Appl. Sci.* **2023**, *13*, 8177. [CrossRef]
19. Bie, M.; Liu, Y.; Li, G.; Hong, J.; Li, J. Real-Time Vehicle Detection Algorithm Based on a Lightweight You-Only-Look-Once (YOLOv5n-L) Approach. *Expert Syst. Appl.* **2023**, *213*, 119108. [CrossRef]
20. Zhou, Z.; Fang, Z.; Wang, J.; Chen, J.; Li, H.; Han, L.; Zhang, Z. Driver Vigilance Detection Based on Deep Learning with Fused Thermal Image Information for Public Transportation. *Eng. Appl. Artif. Intell.* **2023**, *124*, 106604. [CrossRef]
21. Li, Y.; Xue, J.; Zhang, M.; Yin, J.; Liu, Y.; Qiao, X.; Zheng, D.; Li, Z. YOLOv5-ASFF: A Multistage Strawberry Detection Algorithm Based on Improved YOLOv5. *Agronomy* **2023**, *13*, 1901. [CrossRef]
22. Yu, G.; Zhou, X. An Improved YOLOv5 Crack Detection Method Combined with a Bottleneck Transformer. *Mathematics* **2023**, *11*, 2377. [CrossRef]
23. Yang, W.; Liu, T.; Jiang, P.; Qi, A.; Deng, L.; Liu, Z.; He, Y. A Forest Wildlife Detection Algorithm Based on Improved YOLOv5s. *Animals* **2023**, *13*, 3134. [CrossRef] [PubMed]
24. Niu, S.; Zhou, X.; Zhou, D.; Yang, Z.; Liang, H.; Su, H. Fault Detection in Power Distribution Networks Based on Comprehensive-YOLOv5. *Sensors* **2023**, *23*, 6410. [CrossRef] [PubMed]
25. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2018), Salt Lake City, UT, USA, 18–23 June 2018. [CrossRef]
26. Ma, N.; Zhang, X.; Zheng, H.T.; Sun, J. ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design. In Proceedings of the 15th European Conference on Computer Vision (ECCV 2018), Munich, Germany, 8–14 September 2018. [CrossRef]
27. Zhang, T.; Sui, Y.; Wu, S.; Shao, F.; Sun, R. Table Structure Recognition Method Based on Lightweight Network and Channel Attention. *Electronics* **2023**, *12*, 673. [CrossRef]
28. Wei, B.; Chen, H.; Ding, Q.; Luo, H. SiamAGN: Siamese Attention-Guided Network for Visual Tracking. *Neurocomputing* **2022**, *512*, 69–82. [CrossRef]
29. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. CBAM: Convolutional Block Attention Module. In Proceedings of the 15th European Conference on Computer Vision (ECCV 2018), Munich, Germany, 8–14 September 2018. [CrossRef]
30. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2020), Seattle, WA, USA, 13–19 June 2020. [CrossRef]
31. Hu, J.; Shen, L.; Albanie, S.; Sun, G.; Wu, E. Squeeze-and-Excitation Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 2011–2023. [CrossRef]
32. Lu, A.; Ma, L.; Cui, H.; Liu, J.; Ma, Q. Instance Segmentation of Lotus Pods and Stalks in Unstructured Planting Environment Based on Improved YOLOv5. *Agriculture* **2023**, *13*, 1568. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.



Article

Prediction of Potato (*Solanum tuberosum* L.) Yield Based on Machine Learning Methods

Jarosław Kurek ^{1,*}, Gniewko Niedbała ², Tomasz Wojciechowski ², Bartosz Świderski ¹, Izabella Antoniuk ¹, Magdalena Piekutowska ³, Michał Kruk ¹ and Krzysztof Bobran ⁴

¹ Department of Artificial Intelligence, Institute of Information Technology, Warsaw University of Life Sciences, Nowoursynowska 159, 02-776 Warsaw, Poland; bartosz_swiderski@sggw.edu.pl (B.Ś.); izabella_antoniuk@sggw.edu.pl (I.A.); michal_kruk@sggw.edu.pl (M.K.)

² Department of Biosystems Engineering, Faculty of Environmental and Mechanical Engineering, Poznań University of Life Sciences, Wojska Polskiego 50, 60-627 Poznań, Poland; gniewko.niedbala@up.poznan.pl (G.N.); tomasz.wojciechowski@up.poznan.pl (T.W.)

³ Department of Botany and Nature Protection, Institute of Biology, Pomeranian University in Słupsk, 22b Arciszewskiego St., 76-200 Słupsk, Poland; magdalena.piekutowska@ups.edu.pl

⁴ Seth Software sp. z o.o., Strefowa 1, 36-060 Głogów Małopolski, Poland; kbobran@seth.software

* Correspondence: jaroslaw_kurek@sggw.edu.pl

Abstract: This research delves into the application of machine learning methods for predicting the yield of potato varieties used for French fries in Poland. By integrating a comprehensive dataset comprising agronomical, climatic, soil, and satellite-based vegetation data from 36 commercial potato fields over five growing seasons (2018–2022), we developed three distinct models: non-satellite, satellite, and hybrid. The non-satellite model, relying on 85 features, excludes vegetation indices, whereas the satellite model includes these indices within its 128 features. The hybrid model, combining all available features, encompasses a total of 165 features, presenting the most-comprehensive approach. Our findings revealed that the hybrid model, particularly when enhanced with SVM outlier detection, exhibited superior performance with the lowest Mean Absolute Percentage Error (MAPE) of 5.85%, underscoring the effectiveness of integrating diverse data sources into agricultural yield prediction. In contrast, the non-satellite and satellite models displayed higher MAPE values, indicating less accuracy compared to the hybrid model. Advanced data-processing techniques such as PCA and outlier detection methods (LOF and One-Class SVM) played a pivotal role in model performance, optimising feature selection and dataset refinement. The study concluded that machine learning methods, particularly when leveraging a multifaceted approach involving a wide array of data sources and advanced processing techniques, can significantly enhance the accuracy of agricultural yield predictions. These insights pave the way for more-efficient and -informed agricultural practices, emphasising the potential of machine learning in revolutionising yield prediction and crop management.

Keywords: machine learning; yield prediction; potato

Citation: Kurek, J.; Niedbała, G.; Wojciechowski, T.; Świderski, B.; Antoniuk, I.; Piekutowska, M.; Kruk, M.; Bobran, K. Prediction of Potato (*Solanum tuberosum* L.) Yield Based on Machine Learning Methods. *Agriculture* **2023**, *13*, 2259. <https://doi.org/10.3390/agriculture13122259>

Academic Editor: Francesco Marinello

Received: 10 November 2023

Revised: 3 December 2023

Accepted: 7 December 2023

Published: 11 December 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The potato (*Solanum tuberosum* L.) is one of the basic species of cultivated plants in the world. According to FAOSTAT data, the world potato production in 2021 reached 359 million t, and the pioneers in the cultivation of this species were China, India, and Ukraine [1]. In 2021, the highest tuber yield per hectare was recorded in the United States, approximately 51 t, New Zealand, 50.7 t, and Kuwait, 48.7 t. In Europe in 2021, the countries with the highest yield of potato tubers were: France, the Netherlands, Belgium, and Germany. In these countries, the discussed species yields were in the range of 40 to 45 t/ha [2].

In recent years, a change in culinary preferences of consumers towards potatoes has been observed, translating directly to trends in the cultivation of this plant. Currently, the dynamic growth of the market of “convenience” food, as well as fried products can be noted in developing and developed countries around the world [3]. The intensive development of potato processing contributed to the increase in the demand for fast food. Current trends in agricultural development strongly support precision agriculture. The idea behind this is centred on the keywords: observation, measurement, and constant response to inter- and intra-field crop variability [4]. In the case of industrial potato production, precision agriculture—constant monitoring of crop condition—is needed because of the desired good quality of the final raw material while maintaining a high level of commercial yield. Despite the many tools developed to monitor and analyse potato growth and yield over the last 20–30 years, with a peak in sensor development in the last 10 years, many fry potato farms are looking for superior solutions. Widely tested and refined predictive tools for estimating yield and quality prior to final harvest use advanced artificial intelligence methods, among others. It is believed that crop growth monitoring and yield mapping will become mainstream farm research and development in the coming years. Many input parameters specific to specialised crops have become more readily available. For example, remote sensing data on crop emergence date and aboveground biomass are being used to better set model parameters [5,6]. Hybrid forms of sensor systems and crop growth models will provide better information on crop growth during the season. This, combined with weighing systems and cameras on harvesters, will provide site-specific information on the yield and quality of harvested potatoes [7,8].

Research on the use of plant models in predicting potato yield at the field scale has been conducted for over half a century, but their intensity has increased since the beginning of the second decade of the 21st Century [9]. In these studies, it can be observed that classical plant growth models primarily utilise ground-based data, including commonly used factors such as nitrogen fertilisation levels, air temperature values, sunlight exposure, and precipitation levels. This applies to a wide range of models such as SUBSTOR Potato, CROPSYST-SIMPOTATO, and Potato Calculator [10–14]. It is noted that the limitations for the practical application of such models at the field production scale are data availability, the cost of data acquisition, and data quality issues. From another point of view, another source of input data for predictive models is satellite imagery [14]. With the development of satellite Earth observation systems, improved RS data availability for practical applications, and the increased quality of these data in terms of spatial and temporal resolution, satellite data have been increasingly incorporated into potato predictive models [15,16]. When it comes to potato yield forecasting at field scale, there are few publications describing the combined use of ground-based data (soil, agronomy, weather) and satellite data (vegetation indices) as the input parameters for models [12,17–21]. In the practical application of predictive models in agricultural decision support systems, the flexibility of data source selection for modelling becomes an important functional requirement, considering the aforementioned data availability and quality issues. Often, farmers do not have complete sets of ground-based or satellite data. Therefore, there is a need to evaluate and compare ground-based, satellite, and hybrid models that combine data from both types of sources.

Knowledge of yield determinants is important in the development of crop-management-improvement models [1] for both prediction and classification. In the correct construction of production models, it is useful to demonstrate good knowledge of the research object and to have knowledge of yield determinants and potential disturbances, changing the final modelling effect in an independent way [22,23]. The traits explaining varietal yield or supporting potato yield potential in forecasting models in the literature are classified according to the following categories [15,23,24]:

- Weather traits;
- Agricultural traits;
- Traits conditioned by genotype and phenological traits;
- Soil environment;

- Spectral data, including vegetation indices;
- Indicators related to plant productivity.

Data on weather conditions during the growing season can be treated as a disturbance over which both producers and predictive model developers have no direct influence. The most-advantageous solution from the point of view of preparing climatic model data is the selection of years in which the weather conditions represent the optimal case for the place of cultivation and observation. The dominant meteorological features in the prediction models are: total precipitation, average, minimum, and maximum daily air temperatures, insolation, relative air humidity, evapotranspiration, etc. [23,25–27].

Agrotechnical features are nothing more than variants of potato cultivation, i.e., the sum of mineral and organic fertilisation, the soil cultivation system, irrigation, plant protection, forecrops, etc. [23,28]. In the case of potato cultivation intended for processing, including French fry varieties, agrotechnical requirements and recommendations are usually prepared by companies purchasing the raw material. Such action guarantees an acceptable level of tuber yield and an even quality of the yield taken from different suppliers.

Features associated mainly with genotype and phenological features (meaning successive stages of development achieved by plants while growing in the field) are important for obtaining a satisfactory raw material in terms of quality [29]. Favourable soil conditions are very crucial in potato cultivation. It is known that cultivated plants are “more sensitive” to the abundance of available nutrients in the soil than to the ongoing fertilisation [30,31]. It is also important to maintain the recommended pH and looseness of the soil. Potato cultivation in the desired conditions reduces the occurrence of soil diseases [32], as well as prevents bruises [33]. Features related to potato productivity are usually various indicators, the interpretation of which allows for ongoing analysis of growth, yield, and photosynthetic activity.

In yield-forecasting models, the following are most-often used: Photosynthetically Active Radiation (PAR) and Leaf Area Index (LAI) [34,35]. Measuring these indicators is relatively easy, and the final data are not difficult to interpret. Information obtained using remote sensing and GIS methods is becoming increasingly important in the management of potato cultivation and, thus, in the creation of reliable predictive models. Vegetation indices calculated using these methods are a kind of quantitative measure that is correlated with the amount of biomass or the condition of the vegetation. They are usually formulated as a combination of two or three spectral channels (with red and near-infrared being the most-common). Their values are added, divided, or multiplied in order to obtain one value (index), which tells about the amount and condition of the vegetation [24,36]. A wide application in tuber yield-forecasting models has been confirmed for the Normalised Difference Vegetation Index (NDVI) [37], Normalised Difference Red Edge index (NDRE) [38], Potato Productivity Index (PPI) [16], SAVI, RDVI, and EVI.

In recent years, a departure from the use of classic models for predicting the yield of potato tubers, such as SUBSTOR, POTATO, Lintul-POTATO, etc., can be observed. Classical regression models also do not fully fulfil their role, because the forecast errors generated by such models are very high and, therefore, unacceptable in agriculture [23]. The trend of scientific development in yield modelling runs in two directions. One of them is the improvement by researchers of classic potato models—adapting them to specific climatic or cultivation conditions [10]. The second approach involves modern and reliable modelling techniques—Artificial Neural Networks (ANNs) [23,39], decision trees [37], and deep learning [15,40]. It is worth emphasising that the most-important feature of neural models is their ability to generalise the knowledge they acquire during a specific network learning process. Designing the proper structure of a neural network and determining its parameters requires the use of advanced optimisation algorithms. Solving a specific problem always involves the choice of the type of network. Forecasting issues are usually implemented using MLP models [23,24,41–45]. Neural modelling plays a significant role when solving practical problems requiring a quick response is expected [46]. In addition, analysis carried

out using nonlinear models, which include neural models, are characterised by a smaller forecast error compared to classical methods [47].

Predicting crop yields is an important task for agriculture, and predictive models can be useful in this process. However, there are some limitations and potential sources of error that are worth considering. Here are some of them [48–52]:

- Dependence on historical data.
- Disregarding nonlinear factors.
- Variability of environmental conditions.
- Errors in measurements and other inputs.
- No consideration of changes in agricultural practices.
- Complexity of the interaction between factors.

It is important to understand these limitations and potential sources of error in crop yield prediction. Models can be useful tools, but they should be used carefully, take into account a variety of factors, and evolve with advances in knowledge and data availability.

The aim of this article was to create three models predicting the yield of French fry potatoes grown in Poland using machine learning methods. The research focused on several important scientific aspects, including a thorough analysis of empirical data, which allowed the creation of predictive data aggregates. By subjecting the partial classification results to detailed interpretations, it was possible to reject data introducing distortions and noise in the prediction. Finally, based on the MAPE values, the most-accurate model for predicting tuber yield was indicated.

2. Materials and Methods

2.1. Dataset Description

The data used in this work came from 114 commercial potato fields located in northern Poland. Fields with potato cultivation varied in area from 6.5 to 156 ha. The cultivated potato varieties were Innovator, Ludmilla, Ivory Russet and Zorba. The data covered five growing seasons in the years 2018–2022, containing several types of information, i.e., agronomical data, climatic data, satellite-based vegetation, satellite data, and soil data. Source data were obtained from databases of different natures: public databases as open data, private databases of farmers, and ERP databases of agricultural producers. The field locations are presented in Figure 1. The structure of the potato dataset is shown in Table 1.

The data were divided into two sets, referred to as ground-based data (agronomic and weather data) and satellite-based data. These two sets constituted the sets for the non-integrated terrestrial and non-integrated satellite predictive models, respectively. Both sets were the basis for the creation of the hybrid models.

2.1.1. Data Augmentation

The data augmentation process plays a crucial role in enhancing the performance of machine learning models, especially when dealing with limited datasets. In the context of predicting potato yield, data augmentation involves creating synthetic but realistic data points based on the existing dataset. The augmentation procedure can be broken down into several key steps and has been described below.

Augmentation loop: For each record in the dataset, the algorithm performs the following steps multiple times (5 times), as determined by the number of copies specified:

1. A random change percentage (between 0.01 and 0.05) is chosen within a predefined range, which determines the degree of modification for the augmentation.
2. Noise is generated based on the random change percentage and is applied to both the features and the target variable. This noise addition simulates realistic variability within the data.
3. The new synthetic record, created by applying noise, is then denormalized to bring it back to the original data scale.
4. The synthetic record is appended to the augmented dataset along with its corresponding textual data.

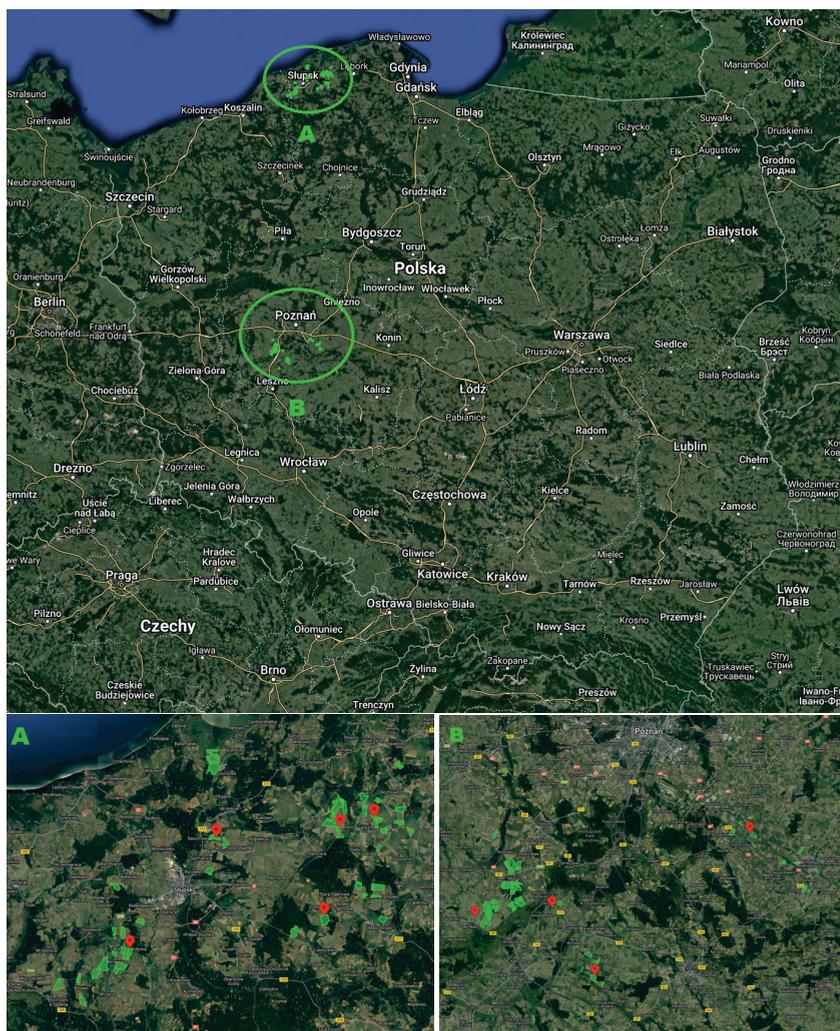


Figure 1. Field locations for the data-collection process: map of Poland, with marked general field placement areas (Top), and a close-up map for the area of Słupsk (A) and Poznań (B).

Table 1. Number of fields for the presented experiments, containing different potato varieties during consecutive years.

Variety	2018	2019	2020	2021	2022	Total
Innovator	60	65	40	55	20	240
Ludmilla	20	30	30	20	15	115
Ivory Russet	5	10	0	0	0	15
Zorba	5	20	15	10	0	50
Total	90	115	95	85	35	420

2.1.2. BBCH-Scale

Part of the data was allocated to the growth stages of the crops cultivated. A universally available BBCH-scale was used in that aspect. The abbreviation BBCH derives from the names of the original participating stakeholders: “Biologische Bundesanstalt,

Bundessortenamt und Chemische Industrie". The BBCH-scale is used to identify the phenological development stages of plants. It was developed for a range of crop species, where similar growth stages of each plant are given the same code. The phenological development stages of plants are used in a number of scientific disciplines (crop physiology, phytopathology, entomology, and plant breeding) and in the agriculture industry (risk assessment of pesticides, timing of pesticide application, fertilisation, and agricultural insurance).

The BBCH-scale uses a decimal code system, which is divided into principal and secondary growth stages and is based on the cereal code system (the Zadoks scale) developed by Jan Zadoks. The phenological development stages obtained from the producer were in following ranges [53,54]:

- (1) BBCH 0–10 (from planting to the beginning of emergence);
- (2) BBCH 11–50 (from the beginning of emergence to the beginning of tuber setting);
- (3) BBCH > 50 (from the beginning of tuber setting to harvest).

Based on the imported data, the BBCH phase limits in the ranges: (1), (2), and (3) were assigned and later used to calculate the aggregated data.

2.1.3. Agronomic Data

The agronomic data were obtained from the Plantator System [55] by Seth Software, as well as from the Plantator System operating during production in regard to: crop register, harvest registration, and registration of hourly work results. The acquired data have different formats, so in some cases, it was necessary to obtain a preprocessed dataset, e.g., from soil test results (pdf) or the locations of crops (jpg). The agronomic data also came from private grower databases. Information on the use of irrigation treatment in the irrigated/non-irrigated structure was also an explanatory feature.

2.1.4. Climate Data

Information on the weather data was gathered from agrometeorological stations situated near the cultivated areas. In cases where there were no local stations, relevant data were acquired from public databases, specifically from weather stations managed by the Institute of Meteorology and Water Management-National Research Institute (IMGW).

2.1.5. Soil Data

Depending on the season, the data regarding soil nutrient content came from different soil testing laboratories. Similarly, the data regarding liquid and solid mineral fertilisation were obtained, resulting in the variability of the analysed components and different units of measurement. All laboratories were nationally accredited for the soil parameters analysed. The soil parameters considered were: the pH and the phosphorus, potassium, and magnesium content. These are the range of parameters most often contracted for soil analyses by agricultural producers in Poland.

2.1.6. Satellite Data

The crop vegetation data were obtained through satellite remote sensing. The primary image database utilised in this study was the European Copernicus Sentinel 2 mission's image database. The Google Earth Engine (GEE) platform served as the direct data collection ("COPERNICUS S2 SR"). A Python script was developed by the authors to acquire, filter, and process images and calculate the Vegetation Indices (VIs). The script was executed on a local server, which communicated with the GEE service.

The secondary image database utilised in this study was the PlanetScope images (Planet Labs), geometrically and atmospherically corrected. The images, clipped to the analysed ROI, were downloaded via Planet's dedicated Data API.

In the initial step, the images for each of the potato fields were filtered based on cloud cover (threshold ranging from 7% to 13% depending on the year) using the QA60 band, for Sentinel and from the "cloud_percent" metadata for PlanetScope. The threshold depended on the availability of images for a given ROI. If there were not at least 3 images for

the ROI in the time period analysed, the threshold was automatically increased. The images and corresponding reflectance values were analysed for the period between April 1 and the end of September for each year under study. The VIs were then calculated for the acquisition date using the obtained reflectance values.

The following vegetation indices were applied in this study: Enhanced Vegetation Index (EVI), Normalised Difference Vegetation Index (NDVI), Renormalised Difference Vegetation Index (RDVI), and Soil-Adjusted Vegetation Index (SAVI) [56]. These VIs are widely used in the literature for predicting potato yield and were calculated according to the Index DataBase. Finally, a total of 16 vegetation features were calculated for the 4 VIs, including the minimum, mean, maximum, and standard deviation groups.

2.1.7. Selyaninov Hydrothermal Coefficient

The investigation of climate variation is a subject of keen interest among scientists in various fields, such as hydrology, meteorology, agriculture, and forestry. All of them aim to determine the most-accurate climatic conditions that will prevail in a specific region in the future. Despite having greater computing power, the analysis of increasingly complex models reveals that numerous environmental factors still need to be considered, rendering the issue unresolved.

Central Europe's different climate change scenarios suggest that an increase in temperature will be accompanied by a slight rise in annual precipitation, which will be redistributed throughout the year. Winter precipitation is projected to increase while summer rainfall to decrease. Given the limited retention capacity and a concomitant increase in evaporation, the amount of water available to plants will be reduced during the growing season, and there may be a depletion of reserves from the winter season. Moreover, the growing variance of precipitation and temperatures should not be overlooked as it indicates that unfavourable extreme situations for plant production are likely to occur more frequently.

One aspect that requires close monitoring is the evaluation of water availability in a particular area, particularly in extreme cases such as floods and droughts. Different indicators are used to measure the severity of water scarcity, one of which is the Selyaninov Hydrothermal Coefficient (HTC). This coefficient assesses drought based on the formula [57]:

$$HTC = 10 \sum n_i P_i \sum n_i t_i \quad (1)$$

where:

n —the length of the period considered in days;

P_i —the rainfall on the i -th day (mm);

t_i —the average daily temperature on the i -th day (°C).

Based on the above properties, three aggregated parameters for three vegetation stages (BBCH-based) were generated to be used as additional prediction features. The rainfall and temperature values used to calculate the HTC were taken from the IMGW net and our own agrometeorological stations, as described in Section 2.1.4.

2.1.8. GDD Features

When plants are not subjected to extreme conditions such as abnormal drought or disease, they usually grow incrementally, and the prevailing temperature heavily influences their growth rate. The Growing Degree Days (GDDs) [58] parameter considers various aspects of local weather, enabling farmers to anticipate and even regulate the pace at which their plants mature, particularly in greenhouse settings.

Provided the plants are not affected by other environmental factors such as soil moisture, their developmental rate from emergence to maturity hinges on the daily air temperature. Specific developmental phases of plants and insects depend on the accumulation of specific quantities of heat, allowing the prediction of when these events should occur during a growing season, regardless of temperature differences across years. The GDDs are defined as the number of degrees above the base temperature, which varies depending on the crop species. The base temperature is the temperature at which plant growth is zero.

To calculate the GDDs, each day's maximum and minimum temperatures are added and divided by two, and the base temperature is then subtracted. The GDDs are accumulated by adding each day's GDD contribution as the season progresses. GDDs can be used for various purposes, including:

- Assessing a region's suitability for cultivating specific crops;
- Estimating the growth stages of crops, weeds, or insects;
- Predicting the maturity and cutting dates of forage crops;
- Determining the optimal timing of fertiliser or pesticide application;
- Estimating heat stress on crops;
- Planning the spacing of planting dates to produce separate harvest dates.

These parameters can be calculated as shown in Equation (2):

$$\text{GDD} = \sum_{i=1}^n n_i \mathbf{T}_{avg} \quad (2)$$

where:

GDD—the Growing Degree Day (°C);

n —the length of the period considered in days;

\mathbf{T}_{avg} —the average daily air temperature ≥ 0 (°C).

Similar to the HTC, aggregated parameters for the three vegetation stages (BBCH-based) were generated based on this parameter.

2.1.9. Total Numerical features

After the initial analysis, a set of features was derived for the presented experiments. Apart from the basic crop data (season, variety, acreage, location of cultivation, age of cultivation, yield), we used the BBCH-scale (see Section 2.1.2).

A total of 250 potential explanatory features were derived for the target. The target variable is defined as the total yield of the harvested crop (harvest) (potato). All numerical data (both explanatory and dependent features) were aggregated to full years (2018, 2019, 2020, 2021, and 2022). The target variable was measured in tons (t). A summary of all the numerical feature groups used in the prediction of potato yield before data pruning is presented in Table 2.

Table 2. Summary of all the numerical feature groups used in the prediction of potato yield before data pruning.

Group of Features	No. of Features
Aggregated weather features	7
Weather features	92
Soil features	17
Agrotechnical treatment features	6
Vegetation indexes GE	64
Vegetation indexes PL	64
Total	250

2.1.10. Data Pruning: Addressing Missing Values

In this study, a common challenge of dealing with missing values in the dataset was encountered. Any data analysis, irrespective of the statistical methods applied, is only as robust as the quality and completeness of the data being analysed. In this case, the dataset initially comprised 250 features collected for predicting potato yield.

The initial step was to identify the extent of missing data in the dataset. This process of quantification was carried out by calculating the percentage of missing values in each variable. It is crucial to note that the quantity of missing values can considerably influence the performance and accuracy of AI models.

In order to maintain the integrity of our study, we established a cutoff threshold of 50%. Any variable with more than 50% missing values was deemed unreliable for our analysis due to the massive information gap. The rationale behind this decision was that imputing more than 50% of the data of a variable can introduce a substantial amount of bias and distortion in the prediction model. It also raises concerns about the reliability and validity of the subsequent findings, as more than half of the information would be synthetic or based on estimates. This decision was rooted in a balance between retaining valuable data and ensuring the reliability and robustness of our models. The rationale for choosing this specific threshold was multi-faceted:

- **Data integrity:** When more than half of the data for a variable are missing, the integrity and representativeness of that variable become questionable. With over 50% missing data, any form of imputation would largely be based on speculation, rather than trends or patterns inherent in the data.
- **Statistical significance:** Variables with significant missing data can potentially skew the results and lead to unreliable conclusions. By setting the threshold at 50%, we aimed to maintain variables that had a statistically significant amount of data, thereby ensuring that our models were built on solid and representative foundations.
- **Balance between data retention and quality:** The 50% threshold strikes a balance between retaining as much data as possible and ensuring the quality of the dataset. This threshold allowed us to keep a substantial portion of the dataset while avoiding the pitfalls of basing our analysis on largely imputed or speculative data.
- **Benchmarking against standard practices:** This threshold is in line with common practices in data science and statistical analysis, where a 50% cutoff is often used as a standard for determining the viability of a variable in a dataset.

By implementing this threshold, we aimed to enhance the robustness and reliability of our predictive models. This approach allowed us to use a dataset that was both comprehensive and credible, leading to more-accurate and -trustworthy outcomes in our study.

After a rigorous examination, it was confirmed that 85 out of 250 features had missing data exceeding the 50% threshold. Therefore, to ensure the reliability of the succeeding analysis, as well as to maintain the robustness of the model, it was decided to exclude those features from the dataset.

Thus, the pruned dataset contained only 86 features, ready for further analysis and AI model training. This data-reduction method helped to maintain the data quality while ensuring that the future predictive model would not suffer from the adverse impacts of missing values and imputation bias.

Moving forward, these 165 features will be used to develop our artificial-intelligence-based prediction models. The retained features were carefully selected from the dataset after excluding those with excessive missing data. The list of the remaining 165 features includes the following:

Taking this strategic approach to data management was meant to ensure the most-accurate and -meaningful results from the AI models in the prediction of potato yield. The final list of numerical feature used in potato yield prediction is presented in Table 3, with a summary of the number of final numerical feature groups provided in Table 4.

Table 3. List of the number of numerical feature groups used in the prediction of potato yield after data pruning.

Variable Type	List of Variables
Agrotechnical treatment features (4 items)	Liquid fertilisation, spraying, planting, broadcast fertilisation
Weather features (23 items)	Average temperature (°C), rainfall (mm), air temperature1 (°C), air temperature2 (°C), air temperature3 (°C), solar panel (mV), precipitation (mm), wind speed AVG (m/s), wind speed Min (m/s), wind speed Max (m/s), battery (mV), leaf wetness time (min), HC serial number, HC air temperature AVG (°C), HC air temperature Max (°C), HC air temperature Max (°C), HC relative humidity AVG (%), HC relative humidity AVG (%), HC relative humidity AVG (%), Dev point temperature AVG (°C), Dev point temperature Max (°C), vapour pressure deficit AVG (mBar), vapour pressure deficit Min (mBar)
Aggregated weather features (6 items)	HTC 0–10, HTC 11–50, HTC > 50, GDD 0–10, GDD > 50, GDD 11–50
Soil features (4 items)	Soil pH H ₂ O, phosphorus (mg/100 g), potassium (mg/100 g), magnesium (mg/100 g)
Vegetation indices GE (calculated based on Sentinel via Google Earth) (64 items)	EVI_GE_0_10_Max, EVI_GE_11_50_Max, EVI_GE_50_Max, EVI_GE_daily_Max, NDVI_GE_0_10_Max, NDVI_GE_11_50_Max, NDVI_GE_50_Max, NDVI_GE_daily_Max, RDVI_GE_0_10_Max, RDVI_GE_11_50_Max, RDVI_GE_50_Max, RDVI_GE_daily_Max, SAVI_GE_0_10_Max, SAVI_GE_11_50_Max, SAVI_GE_50_Max, SAVI_GE_daily_Max, and so on, for mean, Min, StdDev variants
Vegetation indices PL (calculated based on PlanetScope via Planet Labs) (64 items)	EVI_PL_0_10_Max, EVI_PL_11_50_Max, EVI_PL_50_Max, EVI_PL_daily_Max, NDVI_PL_0_10_Max, NDVI_PL_11_50_Max, NDVI_PL_50_Max, NDVI_PL_daily_Max, RDVI_PL_0_10_Max, RDVI_PL_11_50_Max, RDVI_PL_50_Max, RDVI_PL_daily_Max, SAVI_PL_0_10_Max, SAVI_PL_11_50_Max, SAVI_PL_50_Max, SAVI_PL_daily_Max, and so on, for mean, Min, StdDev variants

Table 4. Summary of the number of final numerical feature groups used in prediction of potato yield after data pruning.

Group of Features	No. of Features
Aggregated weather features	4
Weather features	23
Soil features	4
Agrotechnical treatment features	6
Vegetation indexes GE	64
Vegetation indexes PL	64
Total	165

2.2. Data Imputation

Data imputation, or the process of filling in missing data points in datasets, is a critical aspect of predictive modelling [59], particularly in the field of agricultural yield predictions. The robustness and accuracy of Artificial Intelligence (AI) models depend highly on the quality and completeness of the underlying datasets. In the case of predicting potato yield, incomplete datasets can lead to inaccurate models and predictions, thus impeding the optimisation of crop production.

In the context of AI, missing data could induce significant bias, reduce the statistical power, and ultimately distort the representation of the real-world scenario that the AI model is attempting to capture. This issue is particularly pertinent in agricultural datasets, where factors such as weather conditions, soil properties, and crop health measures can be highly variable and sometimes difficult to measure consistently. Without adequate data in these areas, AI models may not accurately reflect the complex interactions and dependencies among these factors, leading to erroneous predictions of potato yield.

Methods of Data Imputation

Data imputation is a critical step in the preprocessing phase of predictive modelling, especially when dealing with incomplete datasets. In the context of this research, we implemented a hybrid approach that combines regression and mean/median imputation strategies. This method intends to balance the bias introduced by mean/median imputation with the variance captured through regression techniques.

The proposed hybrid procedure, outlined in Algorithm 1, iteratively applies polynomial interpolation to create multiple imputations of the missing data, followed by median aggregation to ensure robustness. This method is particularly suitable for datasets with nonlinear relationships among the variables, such as the one used for predicting potato yield in this study. By applying a polynomial approach, we aimed to capture the intricate patterns inherent in the data, thereby enhancing the accuracy of our imputations. The decision to use this technique was based on preliminary analysis indicating significant nonlinear interactions among the predictive features.

Algorithm 1 Hybrid imputation procedure.

```

1: procedure HYBRIDIMPUTATION(DataFrame, ColumnName)
2:   ProcessedColumn  $\leftarrow$  DeepCopy(DataFrame[ColumnName])
3:   ProcessedColumn  $\leftarrow$  AddIndexColumn(ProcessedColumn)
4:   ImputationTargets  $\leftarrow$  [ColumnName]
5:   ThresholdValidValues  $\leftarrow$  86
6:   IterationCount  $\leftarrow$  0
7:   while CountNonMissing(ProcessedColumn[ImputationTargets]) <
   ThresholdValidValues do
8:     TempColumn  $\leftarrow$  InterpolateColumn(ProcessedColumn, IterationCount)
9:     ProcessedColumn  $\leftarrow$  MergeColumns(ProcessedColumn, TempColumn)
10:    IterationCount  $\leftarrow$  Increment(IterationCount)
11:  end while
12:  DataFrame[ColumnName]  $\leftarrow$  ComputeMedian(ProcessedColumn[ImputationTargets])
13:  return DataFrame
14: end procedure

```

2.3. Data Normalisation

Data normalisation is an essential preprocessing step while dealing with machine learning or artificial intelligence algorithms. It is performed to bring all features into the range of 0 to 1, maintaining the distribution and relationships of the original raw data. This normalisation process helps to scale down the values of different scale attributes to a standard scale, which, in turn, enhances the performance of the model by allowing it

to converge faster during training. Additionally, it mitigates the risk of the model being influenced disproportionately by different features.

The particular method of normalisation used in this study was the Min–Max normalisation. This method re-scales features to a fixed range, typically 0 to 1, or alternatively –1 to 1 if there are negative values. This transformation preserves the original distribution of the data while ensuring that the impact of outliers is minimised.

The Min–Max normalisation is defined by the following formula [60,61]:

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (3)$$

where:

- X_{norm} is the normalised value;
- X is the original value;
- X_{min} is the minimum value in the feature column;
- X_{max} is the maximum value in the feature column.

Each data value in the dataset is replaced by its corresponding normalised value, leading to a new dataset where all feature columns are within the same range. By implementing the Min–Max normalisation method, it is ensured that the model is less biased and, hence, more accurate in predicting potato yield based on the given features.

2.4. Prototyping the 3 AI Models: Non-Satellite, Satellite, and Hybrid

The accurate prediction of agricultural yields such as potato can greatly benefit both the farmers and the supply chain stakeholders. Utilising Artificial Intelligence (AI) methods for these predictions can potentially provide robust and reliable estimations. In this research, we investigated the effectiveness of different AI regression algorithms in constructing the predictive models. The objective was to gain insights into the strengths and limitations of various methods and subsequently guide future applications of AI in agriculture.

Three different models were considered in this study, namely non-satellite, satellite, and hybrid models. These models differed in their variable selections, which influenced their representation of real-world conditions.

The non-satellite model uses 37 features, which do not include vegetation indices. In contrast, the satellite model uses 128 features consisting exclusively the vegetation indices. These two models served as a basis for comparison to evaluate the contribution of vegetation indices in improving prediction accuracy. The hybrid model takes a comprehensive approach by including all 86 features available, thus merging the characteristics of both non-satellite and satellite models.

In constructing these models, a range of regression algorithms was applied, including Linear Regression [62], Ridge [63], Lasso [64], Elastic Net [65], the XGBoost Regressor [66], the Random Forest Regressor [67], the MLPRegressor [68] with different hidden layer sizes, the SGDRegressor [69], and Support Vector Regression (SVR) [70] with different parameters. These algorithms were chosen due to their diverse underlying principles, which provides a broad perspective on the prediction problem.

In the following sections, we detail the construction of these models and discuss the potential implications of our findings. The models were constructed using the Python programming language with the aid of powerful libraries such as scikit-learn and XGBoost.

2.5. Data Partitioning: Training, Validation, and Testing

An essential aspect of building robust and generalizable AI models is data partitioning. This process involves dividing the available dataset into distinct subsets: the training, validation, and test sets.

The training set is utilised to train the model, which essentially involves the adjustment of the model's parameters based on the input–output pairs in the data. The validation set is used during model training to provide an unbiased evaluation of the model's performance.

It allows for the tuning of hyperparameters and helps in model selection. Importantly, the validation set serves as a checkpoint to prevent overfitting, which occurs when the model learns the training data too well and performs poorly on unseen data. Finally, the test set is a separate data subset that is only used once the model has been trained and validated. It offers an objective evaluation of the final model's performance, representing how well the model is likely to perform on unseen, real-world data.

In this study, due to the forecasting nature of the task for upcoming years, we split the data based on the years:

- `df_train`—data from the years 2018 and 2019;
- `df_val`—data from the year 2020;
- `df_test`—data from the year 2021.

The model was trained using “`df_train`”, while its performance was monitored on “`df_val`”. Lastly, “`df_test`” was set aside for the final evaluation of the model, providing a benchmark of its performance on unseen data that did not participate in training. As such, our main focus was on the results obtained on “`df_test`”.

2.6. Feature Selection

In this study, feature selection was executed using multiple methods including stepwise regression (`stepwisefit`), the Pearson correlation, the Chi-squared (χ^2) test, and Principal Component Analysis (PCA) [71–74]. These techniques were designed to select the most-relevant features for the task of predicting future years, thus potentially improving model accuracy, computational efficiency, and model interpretability.

2.6.1. Stepwise Regression

The stepwise regression [75] method was applied first for feature selection. Stepwise regression is an iterative process of adding and removing predictor features based on their statistical significance in a regression model. The technique starts from an initial model and takes steps to modify it by adding or removing predictors. The statistical significance of a predictor is typically measured by the p -value of the F-statistic when testing the models with and without the predictor.

In general, the stepwise regression process can be described as follows:

1. Fit the initial model.
2. If any predictors not in the model have p -values less than the entry tolerance (e.g., 0.05), add the one with the smallest p -value and repeat this step. If not, proceed to the next step.
3. If any predictors in the model have p -values greater than the exit tolerance (e.g., 0.10), remove the one with the largest p -value, and go back to the previous step. If not, stop.

It should be noted that the stepwise regression method is heuristic and does not guarantee that the final model is globally optimal, meaning that it has the best possible fit to the data. A different initial model or a different sequence of steps could lead to a better fit. In this sense, stepwise models are locally optimal, but not necessarily globally.

In this study, the stepwise regression function, “`stepwisefit`”, was tested using a range of `penter` and `premove` values. Specifically, 90 different pairs of `penter` and `premove` were used, from (0.01, 0.06) to (0.9, 0.95). The goal of this testing was to explore how different thresholds for adding and removing features would impact the feature sets that the stepwise regression selected. However, only unique feature sets were extracted, which means that there may not necessarily be 90 distinct feature sets as a result of this procedure. The exact thresholds tested in this study are listed in Table 5.

It should be noted that different `penter` and `premove` values can have significant impacts on the stepwise regression outcomes. Lower `penter` values mean that the bar for adding a feature to the model is set higher, as it needs to have a higher level of statistical significance to be included. Similarly, higher `premove` values mean that the bar for removing a feature from the model is set lower, as it can be excluded even if its

significance is still relatively high. Therefore, different combinations of the penter and premove values can lead to diverse sets of selected features, providing a broad exploration of possible models.

Table 5. penter and premove values for stepwise regression.

No.	Penter	Premove
1	0.01	0.06
2	0.02	0.07
3	0.03	0.08
4	0.04	0.09
...
87	0.87	0.92
88	0.88	0.93
89	0.89	0.94
90	0.9	0.95

2.6.2. Pearson Correlation

In addition to stepwise regression, the Pearson correlation method was also used for feature selection [71,76]. It measures the linear relationship between two features, ranging from -1 to 1 , where 1 means a perfect positive linear relationship, -1 means a perfect negative linear relationship, and 0 means no linear relationship.

In this study, if the absolute value of the Pearson correlation between two features exceeded 0.95 , one of the two correlated features was removed from the set of predictive features. This was performed to mitigate the issue of multicollinearity, which can affect the performance and interpretability of the model.

2.6.3. Chi-Squared Test

The Chi-squared test was also applied as a feature-selection method [71]. This statistical test measures the independence between categorical features. In the context of feature selection, the Chi-squared test can be used to select those features that are most likely to be independent of each other and dependent on the target variable.

In this study, if the p -value of the Chi-squared test was greater than 0.05 , the corresponding feature was added to the set of predictive features. Otherwise, the feature was blocked and not included in the set of predictive features.

2.6.4. Principal Component Analysis

Lastly, Principal Component Analysis (PCA) was utilised as a feature-selection and dimensionality-reduction method. PCA transforms the original features into a new set of features, which are linear combinations of the original ones [72,77]. These new features (or principal components) are uncorrelated with each other.

In this study, PCA was performed for different numbers of principal components (3, 4, 5, 6, 7, 8, 9, 10). The goal was to assess whether generating artificial features through PCA would enhance the performance of the model. This was performed both for the full set of features and the features selected by stepwise regression.

The advantage of PCA lies in its ability to transform a high-dimensional dataset into a lower-dimensional one while retaining most of the important information. However, the interpretability of the model can be compromised because the new features (principal components) are artificial and are not directly interpretable in terms of the original features.

3. Outlier Detection

Outlier detection is an important step in data preprocessing. Outliers are unusual data points that deviate significantly from the rest of the data. While some outliers may be errors and, hence, require correction, others may carry important information about the

data. In the presented study, two methods for outlier detection were used: Local Outlier Factor (LOF) and One-Class SVM [78].

Both methods have their own strengths and are appropriate for different types of datasets. In general, the LOF is good at detecting outliers that are in low-density regions, while One-Class SVM is effective at identifying outliers that are far away from the majority of the data.

3.1. Local Outlier Factor

The LOF method measures the local density deviation of a given data point with respect to its neighbours [79]. It considers as outliers the samples that have a substantially lower density than their neighbours. The number of neighbours considered (parameter “*n_neighbours*”) is typically set to be 20% of the total number of samples. The outline of the methods is presented in Algorithm 2.

Algorithm 2 Pseudocode for local outlier factor.

```

1: procedure LOF( $X, n\_neighbours$ )
2:   for  $x \in X$  do
3:     Calculate the distance to the  $n\_neighbours$  nearest neighbours of  $x$ 
4:     Compute the reachability distance of  $x$ 
5:     Compute the local reachability density of  $x$ 
6:   end for
7:   for  $x \in X$  do
8:     Compute the LOF of  $x$  as the average ratio of the local reachability densities of
       the neighbours of  $x$  to the local reachability density of  $x$ 
9:   end for
10:  Return the LOF of each sample
11: end procedure

```

3.2. One-Class SVM

One-Class SVM [77] is a method associated with the SVM family, but it is suited for the problem of outlier detection. The class of interest is modelled with a tight sphere in the feature space characterising the normal behaviour, and those instances that fall outside this sphere are considered outliers. The parameters used in the experiments are kernel = “rbf”, gamma = “0.1”, and nu = 0.5. Algorithm 3 presents the general overview of this procedure.

Algorithm 3 Pseudocode for One-Class SVM.

```

1: procedure ONECLASSSVM( $X, nu, kernel, gamma$ )
2:  Initialise One-Class SVM with parameters  $nu, kernel,$  and  $gamma$ 
3:  Fit SVM to the data  $X$ 
4:  Predict the labels (1 for inliers,  $-1$  for outliers) for  $X$ 
5:  Return the predicted labels
6: end procedure

```

4. Results and Discussion

In the presented experiments, a total of three models were prepared: the non-satellite, satellite, and hybrid one, where the first two take into account only subsets of feature, either excluding or including vegetation data, while the final model incorporates all potential features. Table 6 outlines the dataset organisation for different models, while Table 7 shows the parameter configurations used in each case.

Table 6. Dataset organisation for all prepared models.

Model (Number)	Training Set (Samples/Features)	Validation Set (Samples/Features)	Test Set (Samples/Features)
NSM Without Outlier Detection (1)	205/37	95/37	120/37
NSM With Outlier Detection Using Local Outlier Factor (2)	200/37	95/37	120/37
NSM With Outlier Detection Using One-Class SVM (3)	103/37	95/37	120/37
SM Without Outlier Detection (4)	205/128	95/128	120/128
SM With Outlier Detection Using Local Outlier Factor (5)	201/128	95/128	120/128
SM With Outlier Detection Using One-Class SVM (6)	104/128	95/128	120/128
HM Without Outlier Detection (7)	205/165	95/165	120/165
HM With Outlier Detection Using Local Outlier Factor (8)	200/165	95/165	120/165
HM With Outlier Detection Using One-Class SVM (9)	120/165	95/165	101/165

Table 7. Setup of parameters used for the prepared models. Model numbers refer directly to the method organisation presented in Table 6.

Model Number	Is_Stepwise Fit_Used	Penter Remove	Is_Pearson_Used	Is_Chi2_Used	Is_PCA	n_PCA_Components
(1)	True	0.8 0.85	False	False	True	5
(2)	False	N/A N/A	False	False	True	5
(3)	True	0.3 0.35	False	False	True	5
(4)	True	0.5 0.44	False	False	False	0
(5)	True	0.4 0.45	False	False	False	0
(6)	True	0.4 0.45	False	False	False	0
(7)	True	0.6 0.65	True	False	False	0
(8)	True	0.2 0.25	False	False	False	0
(9)	True	0.1 0.15	False	False	True	5

4.1. Non-Satellite Model

The Non-Satellite Model (NSM) leverages 37 features excluding the vegetation indices data. The list of features is outlined in Table 8.

Table 8. Summary of the number of numerical feature groups used in non-satellite model.

Group of Features	No.
Aggregated weather features	4
Weather features	23
Soil features	4
Agrotechnical treatment features	6
Total	37

The modelling for the non-satellite data considered three different scenarios: (a) without outlier detection, (b) with outlier detection using the Local Outlier Factor method, and (c) with outlier detection using the One-Class SVM method.

In the case of modelling without outlier detection, the Mean Absolute Percentage Error (MAPE) was found to be 17.31% using SVR. It is important to note that 32 (5 PCs) features were identified as significant in this scenario.

In the case of modelling with outlier detection using the Local Outlier Factor, the Mean Absolute Percentage Error (MAPE) was found to be 16.99% using SVR. It is important to note that only five (PCs) features were identified as significant in this scenario.

In the third scenario, when the One-Class SVM method was incorporated, the Mean Absolute Percentage Error (MAPE) was found to be 18.47% using XGB. In addition, 18 features were identified as significant for the model built using the modified datasets.

4.2. Satellite Model

The Satellite Model (SM) takes into account the vegetation indices, containing a total of 128 features. The model creation for the satellite data considered the same three scenarios as for the non-satellite model.

In the case of modelling without outlier detection, the Mean Absolute Percentage Error (MAPE) equalled 14.87% using Ridge, and 92 features were identified as significant for the model built using the modified datasets.

In the case of modelling with outlier detection using the Local Outlier Factor, the Mean Absolute Percentage Error (MAPE) equalled 15.43% using Ridge, and 83 features were identified as significant for the model built using the modified datasets.

In the final scenario with the One-Class SVM method, the Mean Absolute Percentage Error (MAPE) equalled 16.38% using Ridge, and 102 features were identified as significant for the model built using the modified datasets.

4.3. Hybrid Model

The Hybrid Model (HM) takes into account all 165 features. This includes both vegetation indices and the features used in the non-satellite model. The modelling process for the whole set of data (hybrid model) considered the same three scenarios, including the approach without outlier detection, as well as two additional ones, using the LOF and One-Class SVM for this problem.

In the case of modelling without outlier detection, the Mean Absolute Percentage Error (MAPE) was found to be 6.10% using XGB. It is important to note that 79 features were identified as significant in this scenario.

Before applying the Local Outlier Factor method for outlier detection, the dimensions of the training, validation, and test datasets were as presented in Table 6. After applying the Local Outlier Factor method, the Mean Absolute Percentage Error (MAPE) was found to be 6.94% using Random Forest. It is important to note that 80 features were identified as significant in this scenario.

In the final scenario, with the initial dataset dimensions as with the LOF method, One-Class SVM was applied. In that case, the training dataset was reduced, indicating that the method identified and removed 11 records as outliers. In this case, the Mean Absolute

Percentage Error (MAPE) was found to be 5.85% using XGB. In addition, 57 features were identified as significant for the model built using the modified datasets.

4.4. Models Comparison

Although it is believed in agricultural practice that the potato is one of the plants with low production requirements, potato varieties for frying purposes need cultivation management at a very high level [80]. Choosing a good variety is a key element in determining the plant's behaviour under field conditions. The information related to the traits responsible for the quality of the product—ready to eat—is “written” in the genotype: high nutritional value and good sensory properties. Most quality traits of tubers are strongly influenced by a number of factors acting on the potato during the growing season [81]. It is known that yield plays a key role in the cultivation of potatoes for frying purposes, as it generates farm profitability [82].

Yield, or the product extracted from the crop, can be considered in various aspects. Potential (theoretical) yield is achieved when the main abiotic factors: CO₂ concentration, solar radiation, and air temperature, are used by the plants with the greatest efficiency [81]. To estimate the potential yield, additional aspects must be taken into account. It should be assumed that a particular variety is grown in an environment that is optimal for it, with sufficient water and nutrients, as well as effective control of all biotic stresses. Potential yield is important for crops and environments where irrigation, the amount and distribution of rainfall, or a combination of irrigation and rainfall ensure that water deficits do not reduce yields [83]. Determining the level of potential yield is difficult, but feasible. Simulation modelling, the results of detailed agronomic experiments, yield tests, and knowledge of the maximum yields achieved by farmers are used to achieve this goal [84,85].

Actual yield is the real harvest achieved by most producers under actual production conditions. Real yield is determined relatively easily, but accurate analytical results can only be obtained by ongoing monitoring of yield potential during the growing season. The integration of several methods then comes to the rescue: remote sensing, geospatial analysis, and modelling combined with method validation through field experiments [81,84].

Maintaining high yield potential in the era of climate change is a very difficult task. The relationship between potential, achievable, and real potato yield is well explained by, i.e., yield gap analysis [86]. The yield gap shows the relationship between quantitative differences in potential, attainable, and actual yield at a specific spatial and temporal scale [85]. This analysis makes it possible to reliably identify unused food production capacity [81,87].

The above considerations show that yield prediction, regardless of the purpose of the forecast, is necessary and important [88]. Most valuable, from the point of view of agricultural practice, are models that allow the prediction of pre-harvest yields, in the current agronomic season [23,41,71]. In the case of potato production for French fries, the prediction of the actual yield of tubers before harvest provides the producer with a range of valuable information. They can be the basis for considering the amount of potential profit, the degree of fulfilment of the contract agreement, and the security of storage space [23,37]. The prediction of potential tuber yield, made before harvest, is also crucial for breeders of new varieties and seed companies [23,89]. The results of the analyses will indicate the “fit” of the tested genotypes to local growing conditions while maintaining a high level of controllable factors. The yield gap forecast provides valuable knowledge to institutions that track national and global food resources. It allows estimating food shortages, especially in poor countries with malnourished populations. Currently, it is believed that actual potato yields only reach 2/3 of their potential. Breeders of new varieties are far less likely to fill the gaps with improved, high-yielding genotypes than they could [33,81]. Effective planning and management of potato production now require the use of effective forecasting tools [90]. Tuber-yield-forecasting products must be carefully prepared and well thought out. The greatest difficulties in working with forecasting models are the selection of an appropriate prediction method and the selection of independent

variables that realistically affect tuber yield. It is important that all of the variables tested are readily available to the average user of such models and describe the relationships between phenomena in potato cultivation in a way that is understandable to the producer [16,23,24].

An important measure of prediction quality is the MAPE. The MAPE is defined as the average variance between the significant values in the dataset and the projected values in the same dataset [91]. The interpretation of the magnitude of this error is as follows: a MAPE of less than 10% indicates a very good model fit; when the MAPE is in the range of 10–20%, the degree of model fitness is good. A forecasting model that achieves a MAPE error of more than 30% should be rejected due to the poor mapping of predicted values to the actual ones [45,92]. In agricultural research, an acceptable upper limit for the MAPE's magnitude is around 15% [23,41,42].

Current trends in potato yield forecasting are mainly directed toward the use of various spectral indices and GIS data as independent variables for model construction [16,37,38]. Al Gaadi et al. [93] assessed crop condition and predicted potato tuber yield in Saudi Arabia. Two vegetation indices, NDVI and SAVI, were generated from Landsat-8 and Sentinel 2 satellite images acquired from different stages of potato growth. Yield samples were collected 2–3 days before harvest and correlated with the final yield. Based on this, yield-prediction models and yield maps were developed. The results showed that the difference between predicted yield values and actual yield values (prediction error) ranged from 7.9 to 13.5% for Landsat-8 images and from 3.8 to 10.2% for Sentinel-2 images. Since the prediction errors in the above cases did not exceed 15%, the models created by the authors can be used in practical applications. Li et al. [94] attempted to improve potato yield predictions using Unmanned Aerial Vehicle (UAV) remote sensing by incorporating variety information into machine learning methods. The research was conducted in the state of Minnesota—the northern part of the United States. Although the authors failed to generate accurate predictive models, very interesting research conclusions were drawn. Firstly, it was discovered that UAV-based spectral data from early in the growing season at the tuber initiation stage (late June) were more correlated with the commercial yield of potatoes than spectral data from later in the growing season at the tuber maturation stage. Secondly, it was established that combining high-spatial-resolution UAV images and variety information using machine learning algorithms can significantly improve potato yield prediction, when compared with methods excluding the variety information. The work on yield prediction in potato cultivation is difficult, but research shows that the most-accurate models can be achieved with the compilation of multiple variables: agrotechnical, soil, spectral, and meteorological.

In this study, three distinct models were used—non-satellite, satellite, and hybrid. Each of these models was evaluated in three different scenarios: (a) without outlier detection, (b) with outlier detection using the Local Outlier Factor method, and (c) with outlier detection using the One-Class SVM method. The comparative summary of the non-satellite, satellite, and hybrid models is presented in Table 9.

The comparative analysis of the non-satellite, satellite, and hybrid models in potato yield prediction revealed distinct trends in model performance across various scenarios. The hybrid models consistently showed superior predictive accuracy, evidenced by their significantly lower Mean Absolute Percentage Error (MAPE) values in all scenarios. This enhanced performance is likely attributed to the comprehensive integration of both satellite and non-satellite data features, suggesting the critical role of a diverse feature set in predictive modelling.

In scenarios where Principal Component Analysis (PCA) was applied, particularly in the non-satellite and hybrid models with Support Vector Machine (SVM) for outlier detection, there was a notable reduction in the number of features used. This indicates that PCA is effective at refining feature sets, thereby potentially improving model performance. Specifically, the hybrid model with SVM outlier detection not only achieved the lowest MAPE, but also demonstrated the impactful role of PCA in optimising the feature set for enhanced predictive accuracy.

Table 9. Comparative summary of non-satellite, satellite, and hybrid models.

Type	Model	Outlier Detection	No. of Features	MAPE	PCA Used	PCA No. of Features
Non-satellite	SVR	N/A	32	17.31%	True	5
Non-satellite	SVR	LOF	37	16.99%	True	5
Non-satellite	XGB	SVM	18	8.47%	True	5
Satellite	Ridge	N/A	92	14.87%	False	0
Satellite	Ridge	LOF	83	15.43%	False	0
Satellite	Ridge	SVM	102	16.38%	False	0
Hybrid	XGB	N/A	79	6.10%	False	0
Hybrid	Random Forest	LOF	80	6.94%	False	0
Hybrid	XGB	SVM	57	5.85%	True	5

Conversely, the non-satellite models, which lacked satellite-derived vegetation indices, exhibited higher MAPE values. This observation underscores the importance of vegetation indices in yield prediction, highlighting their contribution to model accuracy.

The satellite models presented an interesting trend, where an increase in the number of features, as seen in the SVM scenario, did not correspond to a decrease in the MAPE. This contrasts with the hybrid models, where a more judicious feature selection yielded better results. This suggests that increasing the number of features does not inherently enhance model performance; rather, the relevance and effective integration of these features are crucial.

The influence of outlier detection methods, namely the Local Outlier Factor (LOF) and SVM, varied across the models. While the hybrid models benefited significantly from SVM outlier detection, the impact on the non-satellite and satellite models was less pronounced. This difference in impact reiterates the necessity of context-specific approaches in outlier management for predictive modelling.

In summary, the hybrid models, especially with SVM for outlier detection, emerged as the most-effective strategy, achieving the lowest MAPE (5.85%) and, thereby, indicating the highest prediction accuracy among the evaluated models. This analysis reinforces the need for the careful selection and integration of features, coupled with appropriate data preprocessing techniques, to enhance the performance of machine learning models in agricultural yield prediction.

5. Conclusions

The comprehensive study on predicting potato yield using machine learning methods, specifically in the context of Polish potato varieties used for French fry production, yielded significant insights. The research highlighted the effectiveness of integrating diverse datasets, including both satellite and non-satellite data, in enhancing the accuracy of yield predictions. The hybrid model, which combined these datasets, demonstrated superior performance over models that utilised either non-satellite or satellite data alone. This superiority was evident in its lower Mean Absolute Percentage Error (MAPE) (5.85%), suggesting a higher prediction accuracy. The results clearly indicated that a multifaceted approach, utilising a broad spectrum of data sources, significantly improved the model's ability to predict yield accurately.

Advanced data processing techniques, such as feature selection and outlier detection, were found to play a pivotal role in the performance of the predictive models. The application of Principal Component Analysis (PCA) and outlier detection methods, including the Local Outlier Factor (LOF) and One-Class SVM, contributed to improvements in model accuracy. This underscores the importance of sophisticated data processing in machine learning applications for agricultural yield prediction.

The comparative analysis of the non-satellite, satellite, and hybrid models, as presented in the table “Comparative summary of non-satellite, satellite, and hybrid models”, provided critical insights. The analysis revealed that the hybrid model, especially when coupled with SVM for outlier detection, emerged as the most-effective in predicting potato yield. This model achieved the lowest MAPE, indicating its high accuracy and reliability. In contrast, the non-satellite and satellite models, while beneficial in certain scenarios, did not match the comprehensive accuracy of the hybrid model. The findings from this comparative analysis reinforce the conclusion that a combined approach, utilising an extensive array of features and data sources, is essential for developing robust and accurate agricultural-yield-prediction models.

In conclusion, this study illustrated the potential of machine learning methods in revolutionising agricultural yield predictions. The integration of varied data sources, coupled with advanced data-processing techniques, offers a pathway towards more-efficient, -informed, and -sustainable agricultural practices. As the field of agricultural technology continues to evolve, these findings provide a foundation for further research and development in yield prediction and crop management.

Author Contributions: K.B., G.N. and T.W. conceived of the study design, managed the data collection, built the database, and performed the first data analysis. J.K. and B.Ś. carried out all deep data analyses and built the models until final results were attained. I.A., G.N., T.W., J.K., M.P., M.K. and B.Ś. wrote the manuscript with substantial input from K.B. All authors have read and agreed to the published version of the manuscript.

Funding: The project is co-financed by the European Union from the European Regional Development Fund under the Smart Growth Operational Programme. The project is being conducted under the competition of the National Centre for Research and Development, within the 1.1.1 programme for R&D projects of enterprises “Fast track–Agrotech” Number POIR.01.01.01-00-2298/20.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy issues.

Acknowledgments: We gratefully acknowledge Łukasz Cypcar and Szymon Margański for facilitating the sample collection and organising the data in storage and all the members from Seth Software teams for their assistance and helpful discussions. We thank Joanna Kogut for administrative services, without which we could not provide such fruitful results.

Conflicts of Interest: The authors declare no conflict of interest. Author Krzysztof Bobran was employed by the company Seth Software. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. FAO. FAOSTAT Statistical Database. 2023. Available online: <https://ourworldindata.org/grapher/potato-yields> (accessed on 7 June 2023).
2. Potatonewstoday. FAO Updates Global Potato Statistics. Available online: <https://www.potatonewstoday.com/2022/03/28/fao-updates-global-potato-statistics/> (accessed on 7 June 2023).
3. Popkin, B.M.; Reardon, T. Obesity and the food system transformation in Latin America. *Obes. Rev.* **2018**, *19*, 1028–1064. [CrossRef] [PubMed]
4. Shafi, U.; Mumtaz, R.; García-Nieto, J.; Hassan, S.A.; Zaidi, S.A.R.; Iqbal, N. Precision agriculture techniques and practices: From considerations to applications. *Sensors* **2019**, *19*, 3796. [CrossRef] [PubMed]
5. Vannoppen, A.; Gobin, A. Estimating yield from NDVI, weather data, and soil water depletion for sugar beet and potato in Northern Belgium. *Water* **2022**, *14*, 1188. [CrossRef]
6. Newton, I.H.; Tariqul Islam, A.; Saiful Islam, A.; Tarekul Islam, G.; Tahsin, A.; Razzaque, S. Yield prediction model for potato using landsat time series images driven vegetation indices. *Remote Sens. Earth Syst. Sci.* **2018**, *1*, 29–38. [CrossRef]
7. Cambouris, A.N.; Zebbarth, B.J.; Ziadi, N.; Perron, I. Precision agriculture in potato production. *Potato Res.* **2014**, *57*, 249–262. [CrossRef]

8. Hwang, E.; Park, Y.S.; Kim, J.Y.; Park, S.H.; Kim, J.; Kim, S.H. Intraoperative Hypotension Prediction Based on Features Automatically Generated Within an Interpretable Deep Learning Model. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, 1–15. [CrossRef]
9. Renju, R.S.; Deepthi, P.S.; Chitra, M.T. A Review of Crop Yield Prediction Strategies based on Machine Learning and Deep Learning. In Proceedings of the 2022 International Conference on Computing, Communication, Security and Intelligent Systems (IC3SIS), Kochi, India, 23–25 June 2022; pp. 1–6. [CrossRef]
10. Šťastná, M.; Toman, F.; Dufková, J. Usage of SUBSTOR model in potato yield prediction. *Agric. Water Manag.* **2010**, *97*, 286–290. [CrossRef]
11. Ahmed, M.; Fatima, Z.; Iqbal, P.; Kalsoom, T.; Abbasi, K.S.; Shaheen, F.A.; Ahmad, S. Potato Modelling. In *Systems Modelling*; Ahmed, M., Ed.; Springer: Singapore, 2020; pp. 383–401. [CrossRef]
12. Divya, K.L.; Mhatre, P.H.; Venkatasalam, E.P.; Sudha, R. Crop Simulation Models as Decision-Supporting Tools for Sustainable Potato Production: A Review. *Potato Res.* **2021**, *64*, 387–419. [CrossRef]
13. Travasso, M.I.; Caldiz, D.O.; Saluzzo, J.A. Yield prediction using the SUBSTOR-potato model under Argentinian conditions. *Potato Res.* **1996**, *39*, 305–312. [CrossRef]
14. Bala, S.K.; Islam, A.S. Correlation between potato yield and MODIS-derived vegetation indices. *Int. J. Remote Sens.* **2009**, *30*, 2491–2507. [CrossRef]
15. Gómez, D.; Salvador, P.; Sanz, J.; Casanova, J.L. Potato Yield Prediction Using Machine Learning Techniques and Sentinel 2 Data. *Remote Sens.* **2019**, *11*, 1745. [CrossRef]
16. Gómez, D.; Salvador, P.; Sanz, J.; Casanova, J.L. New spectral indicator Potato Productivity Index based on Sentinel-2 data to improve potato yield prediction: A machine learning approach. *Int. J. Remote Sens.* **2021**, *42*, 3426–3444. [CrossRef]
17. Sun, J.; Di, L.; Sun, Z.; Shen, Y.; Lai, Z. County-Level Soybean Yield Prediction Using Deep CNN-LSTM Model. *Sensors* **2019**, *19*, 4363 [CrossRef] [PubMed]
18. Gobin, A.; Sallah, A.H.M.; Curnel, Y.; Delvoye, C.; Weiss, M.; Wellens, J.; Piccard, I.; Planchon, V.; Tychon, B.; Goffart, J.P.; et al. Crop Phenology Modelling Using Proximal and Satellite Sensor Data. *Remote Sens.* **2023**, *15*, 2090. [CrossRef]
19. Lin, Y.; Li, S.; Ye, Y.; Li, B.; Li, G.; Lyv, D.; Jin, L.; Bian, C.; Liu, J. Methodological evolution of potato yield prediction: A comprehensive review. *Front. Plant Sci.* **2023**, *14*, 1214006. [CrossRef]
20. Akhand, K.; Nizamuddin, M.; Roytman, L.; Kogan, F. Using remote sensing satellite data and artificial neural network for prediction of potato yield in Bangladesh. In *Remote Sensing and Modelling of Ecosystems for Sustainability XIII*; SPIE: Bellingham, WA, USA, 2016; Volume 9975, pp. 52–66.
21. Al-Gaadi, K.A.; Hassaballa, A.A.; Madugundu, R.; Tola, E.; Fulleros, R.B. Prediction of potato high-yield zones of a field: Bivariate frequency ratio technique. *Curr. Sci.* **2020**, *119*, 992. [CrossRef]
22. Noman, A.M.; Haidar, Z.A.; Aljumah, A.S.; Almutairi, S.Z.; Alqahtani, M.H. Forecasting the Distortion in Solar Radiation during Midday Hours by Analyzing Solar Radiation during Early Morning Hours. *Appl. Sci.* **2023**, *13*, 6049. [CrossRef]
23. Piekutowska, M.; Niedbała, G.; Piskier, T.; Lenartowicz, T.; Pilarski, K.; Wojciechowski, T.; Pilarska, A.A.; Czechowska-Kosacka, A. The application of multiple linear regression and artificial neural network models for yield prediction of very early potato cultivars before harvest. *Agronomy* **2021**, *11*, 885. [CrossRef]
24. Hara, P.; Piekutowska, M.; Niedbała, G. Selection of independent variables for crop yield prediction using artificial neural network models with remote sensing data. *Land* **2021**, *10*, 609. [CrossRef]
25. Li, Q.; Zhang, S. Impacts of recent climate change on potato yields at a provincial scale in Northwest China. *Agronomy* **2020**, *10*, 426. [CrossRef]
26. Rymuza, K.; Radzka, E.; Lenartowicz, T. Effect of weather conditions on early potato yields in east-central Poland. *Commun. Biometry Crop Sci.* **2015**, *10*, 65–72.
27. Nyawade, S.O.; Gitari, H.I.; Karanja, N.N.; Gachene, C.K.; Schulte-Geldermann, E.; Parker, M.L. Yield and evapotranspiration characteristics of potato-legume intercropping simulated using a dual coefficient approach in a tropical highland. *Field Crop. Res.* **2021**, *274*, 108327. [CrossRef]
28. Blecharczyk, A.; Kowalczewski, P.L.; Sawinska, Z.; Rybacki, P.; Radzikowska-Kujawska, D. Impact of Crop Sequence and Fertilization on Potato Yield in a Long-Term Study. *Plants* **2023**, *12*, 495. [CrossRef] [PubMed]
29. Pandey, J.; Scheuring, D.C.; Koym, J.W.; Vales, M.I. Genomic regions associated with tuber traits in tetraploid potatoes and identification of superior clones for breeding purposes. *Front. Plant Sci.* **2022**, *13*, 952263. [CrossRef] [PubMed]
30. Singh, B. Are nitrogen fertilizers deleterious to soil health? *Agronomy* **2018**, *8*, 48. [CrossRef]
31. Hasnain, M.; Chen, J.; Ahmed, N.; Memon, S.; Wang, L.; Wang, Y.; Wang, P. The effects of fertilizer type and application time on soil properties, plant traits, yield and quality of tomato. *Sustainability* **2020**, *12*, 9065. [CrossRef]
32. Fiers, M.; Edel-Hermann, V.; Chatot, C.; Le Hingrat, Y.; Alabouvette, C.; Steinberg, C. Potato soil-borne diseases. A review. *Agron. Sustain. Dev.* **2012**, *32*, 93–132. [CrossRef]
33. Vreugdenhil, D.; Bradshaw, J.; Gebhardt, C.; Govers, F.; Taylor, M.A.; MacKerron, D.K.; Ross, H.A. *Potato Biology and Biotechnology: Advances and Perspectives*; Elsevier: Amsterdam, The Netherlands, 2011.
34. Boyd, N.; Gordon, R.; Martin, R. Relationship between leaf area index and ground cover in potato under different management conditions. *Potato Res.* **2002**, *45*, 117–129. [CrossRef]

35. Quiroz, R.; Loayza, H.; Barreda, C.; Gavilán, C.; Posadas, A.; Ramírez, D. Linking process-based potato models with light reflectance data: Does model complexity enhance yield prediction accuracy? *Eur. J. Agron.* **2017**, *82*, 104–112. [CrossRef]
36. Rokhafrouz, M.; Latifi, H.; Abkar, A.; Wojciechowski, T.; Czechowski, M.; Naieni, A.; Maghsoudi, Y.; Niedbała, G. Simplified and Hybrid Remote Sensing-Based Delineation of Management Zones for Nitrogen Variable Rate Application in Wheat. *Agriculture* **2021**, *11*, 1104. [CrossRef]
37. Salvador, P.; Gómez, D.; Sanz, J.; Casanova, J.L. Estimation of potato yield using satellite data at a municipal level: A machine learning approach. *ISPRS Int. J. Geo-Inf.* **2020**, *9*, 343. [CrossRef]
38. Samborski, S.; Leszczyńska, R.; Gozdowski, D. Detecting spatial variability of potato canopy using various remote sensing data. In Proceedings of the Precision Agriculture'21, Budapest, Hungary, 19–22 July 2021; Wageningen Academic Publishers: Wageningen, The Netherlands, 2021; pp. 1786–1798.
39. Prasad Patnaik, P.; Padhy, N. An Approach for Potato Yield Prediction Using Machine Learning Regression Algorithms. In Proceedings of the Next Generation of Internet of Things, Gunupur, India, 5–6 February 2021; Kumar, R., Mishra, B.K., Patnaik, P.K., Eds.; Springer Nature: Singapore, 2023; pp. 327–336.
40. Sharma, A.K.; Rajawat, A.S. Crop Yield Prediction using Hybrid Deep Learning Algorithm for Smart Agriculture. In Proceedings of the 2022 Second International Approach for Potato Yield Prediction Using Machine Learning Regression Algorithms Conference on Artificial Intelligence and Smart Energy (ICAIS), Coimbatore, India, 23–25 February 2022; pp. 330–335. [CrossRef]
41. Niedbała, G. Simple model based on artificial neural network for early prediction and simulation winter rapeseed yield. *J. Integr. Agric.* **2019**, *18*, 54–61. [CrossRef]
42. Niedbała, G.; Kurasiak-Popowska, D.; Piekutowska, M.; Wojciechowski, T.; Kwiatek, M.; Nawracała, J. Application of Artificial Neural Network Sensitivity Analysis to Identify Key Determinants of Harvesting Date and Yield of Soybean (*Glycine max* [L.] Merrill) Cultivar Augusta. *Agriculture* **2022**, *12*, 754. [CrossRef]
43. Niedbała, G.; Wróbel, B.; Piekutowska, M.; Zielewicz, W.; Paszkiewicz-Jasińska, A.; Wojciechowski, T.; Niazian, M. Application of Artificial Neural Networks Sensitivity Analysis for the Pre-Identification of Highly Significant Factors Influencing the Yield and Digestibility of Grassland Sward in the Climatic Conditions of Central Poland. *Agronomy* **2022**, *12*, 1133. [CrossRef]
44. Niedbała, G.; Kurasiak-Popowska, D.; Stuper-Szablewska, K.; Nawracała, J. Application of Artificial Neural Networks to Analyze the Concentration of Ferulic Acid, Deoxynivalenol, and Nivalenol in Winter Wheat Grain. *Agriculture* **2020**, *10*, 127. [CrossRef]
45. Hara, P.; Piekutowska, M.; Niedbała, G. Prediction of Protein Content in Pea (*Pisum sativum* L.) Seeds Using Artificial Neural Networks. *Agriculture* **2022**, *13*, 29. [CrossRef]
46. Boniecki, P.; Sujak, A.; Niedbała, G.; Piekarska-Boniecka, H.; Wawrzyniak, A.; Przybylak, A. Neural Modelling from the Perspective of Selected Statistical Methods on Examples of Agricultural Applications. *Agriculture* **2023**, *13*, 762. [CrossRef]
47. Gonzalez-Sanchez, A.; Frausto-Solis, J.; Ojeda-Bustamante, W. Attribute selection impact on linear and nonlinear regression models for crop yield prediction. *Sci. World J.* **2014**, *2014*, 509429. [CrossRef]
48. Maestrini, B.; Mimić, G.; van Oort, P.A.; Jindo, K.; Brdar, S.; Athanasiadis, I.N.; van Evert, F.K. Mixing process-based and data-driven approaches in yield prediction. *Eur. J. Agron.* **2022**, *139*, 126569. [CrossRef]
49. Morales, A.; Villalobos, F.J. Using machine learning for crop yield prediction in the past or the future. *Front. Plant Sci.* **2023**, *14*, 1128388. [CrossRef]
50. Ansarifar, J.; Wang, L.; Archontoulis, S.V. An interaction regression model for crop yield prediction. *Sci. Rep.* **2021**, *11*, 17754. [CrossRef]
51. Kuradusenge, M.; Hitimana, E.; Hanyurwimfura, D.; Rukundo, P.; Mtonga, K.; Mukasine, A.; Uwitonze, C.; Ngabonziza, J.; Uwamahoro, A. Crop Yield Prediction Using Machine Learning Models: Case of Irish Potato and Maize. *Agriculture* **2023**, *13*, 225. [CrossRef]
52. Yun, S.D.; Gramig, B.M. Spatial Panel Models of Crop Yield Response to Weather: Econometric Specification Strategies and Prediction Performance. *J. Agric. Appl. Econ.* **2022**, *54*, 53–71. [CrossRef]
53. Fadón, E.; Herrero, M.; Rodrigo, J. Flower development in sweet cherry framed in the BBCH scale. *Sci. Hortic.* **2015**, *192*, 141–147. [CrossRef]
54. Alcaraz, M.; Thorp, T.; Hormaza, J. Phenological growth stages of avocado (*Persea americana*) according to the BBCH scale. *Sci. Hortic.* **2013**, *164*, 434–439. [CrossRef]
55. Seth Software Sp. z o.o. Plantator System. 2023. Available online: <https://plantator.com> (accessed on 1 November 2023).
56. Matsushita, B.; Yang, W.; Chen, J.; Onda, Y.; Qiu, G. Sensitivity of the Enhanced Vegetation Index (EVI) and Normalized Difference Vegetation Index (NDVI) to Topographic Effects: A Case Study in High-density Cypress Forest. *Sensors* **2007**, *7*, 2636–2651. [CrossRef]
57. Chmiś-Sikorska, J.; Kepinska-Kasprzak, M.; Struzik, P. Agricultural drought assessment on the base of Hydro-thermal Coefficient of Selyaninov in Poland. *Ital. J. Agrometeorol.* **2022**, *1*, 3–12. [CrossRef]
58. McMaster, G.S.; Wilhelm, W. Growing degree-days: One equation, two interpretations. *Agric. For. Meteorol.* **1997**, *87*, 291–300. [CrossRef]
59. Zhang, Z. Missing data imputation: Focusing on single imputation. *Ann. Transl. Med.* **2016**, *4*, 9.
60. Henderi, H.; Wahyuningsih, T.; Rahwanto, E. Comparison of Min-Max normalization and Z-Score Normalization in the K-nearest neighbor (kNN) Algorithm to Test the Accuracy of Types of Breast Cancer. *Int. J. Inform. Inf. Syst.* **2021**, *4*, 13–20. [CrossRef]

61. Jegorowa, A.; Górski, J.; Kurek, J.; Kruk, M. Use of nearest neighbors (k-NN) algorithm in tool condition identification in the case of drilling in melamine faced particleboard. *Maderas. Ciencia Y Tecnología*. **2020**, *22*, 189–196. Available online: http://www.scielo.cl/scielo.php?script=sci_arttext&pid=50718-221X2020000200189&nrm=iso (accessed on 1 November 2023). [CrossRef]
62. sklearn.linear_model.LinearRegression—Scikit-Learn 1.0.2 Documentation. 2023. Available online: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html (accessed on 1 November 2023).
63. Ridge Regression in Scikit-Learn. 2023. Available online: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Ridge.html (accessed on 1 November 2023).
64. Lasso in Scikit-Learn. 2023. Available online: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Lasso.html (accessed on 1 November 2023).
65. ElasticNet in Scikit-Learn. 2023. Available online: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.ElasticNet.html (accessed on 1 November 2023).
66. XGBoost Python Package. 2023. Available online: https://xgboost.readthedocs.io/en/stable/Python/python_api.html (accessed on 1 November 2023).
67. Random Forest Regressor in Scikit-Learn. 2023. Available online: <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html> (accessed on 1 November 2023).
68. MLP Regressor in Scikit-Learn. 2023. Available online: https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPRegressor.html (accessed on 1 November 2023).
69. SGD Regressor in Scikit-Learn. 2023. Available online: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDRegressor.html (accessed on 1 November 2023).
70. SVR in Scikit-Learn. 2023. Available online: <https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html> (accessed on 1 November 2023).
71. Niedbała, G.; Kurek, J.; Świdorski, B.; Wojciechowski, T.; Antoniuk, I.; Bobran, K. Prediction of Blueberry (*Vaccinium corymbosum* L.) Yield Based on Artificial Intelligence Methods. *Agriculture* **2022**, *12*, 2089. [CrossRef]
72. Kurek, J.; Świdorski, B.; Osowski, S.; Kruk, M.; Barhoumi, W. Deep learning versus classical neural approach to mammogram recognition. *Bull. Pol. Acad. Sci. Tech. Sci.* **2018**, *66*, 831–840. [CrossRef]
73. Swiderski, B.; Kurek, J.; Osowski, S. Multistage classification by using logistic regression and neural networks for assessment of financial condition of company. *Decis. Support Syst.* **2012**, *52*, 539–547. [CrossRef]
74. Osowski, S.; Les, T. Deep learning ensemble for melanoma recognition. In Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 19–24 July 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–7.
75. Gil, F.; Osowski, S.; Slowinska, M. Melanoma recognition using deep learning and ensemble of classifiers. In Proceedings of the 2022 23rd International Conference on Computational Problems of Electrical Engineering (CPEE), Zuberec, Slovak Republic, 11–14 September 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 1–4.
76. Kruk, M.; Kurek, J.; Osowski, S.; Koktysz, R.; Swiderski, B.; Markiewicz, T. Ensemble of classifiers and wavelet transformation for improved recognition of Fuhrman grading in clear-cell renal carcinoma. *Biocybern. Biomed. Eng.* **2017**, *37*, 357–364. [CrossRef]
77. Siwek, K.; Osowski, S.; Kurek, J. Ensemble Neural Network Approach to the Load Forecasting in the Power System. In Proceedings of the International IEEE Conference on ISETET’05, Lviv, Ukraine, 4–7 July 2005; pp. 380–383.
78. Kurek, J.; Osowski, S. Support Vector Machine for diagnosis of the bars of cage inductance motor. In Proceedings of the 2008 15th IEEE International Conference on Electronics, Circuits and Systems, Saint Julian’s, Malta, 31 August–3 September 2008; IEEE: Piscataway, NJ, USA, 2008; pp. 1022–1025.
79. Cheng, Z.; Zou, C.; Dong, J. Outlier Detection Using Isolation Forest and Local Outlier Factor. In Proceedings of the Conference on Research in Adaptive and Convergent Systems, RACS’19, Chongqing, China, 24–27 September 2019; Association for Computing Machinery: New York, NY, USA, 2019; pp. 161–168. [CrossRef]
80. Sawicka, B.; Pyszczółkowski, P.; Kiełtyka-Dadasiewicz, A.; Barbaś, P.; Ćwintal, M.; Krochmal-Marczak, B. The Effect of Effective Microorganisms on the Quality of Potato Chips and French Fries. *Appl. Sci.* **2021**, *11*, 1415. [CrossRef]
81. Haverkort, A.; Struik, P. Yield Levels of Potato Crops: Recent Achievements and Future Prospects. *Field Crops Res.* **2021**, *182*, 76–85. [CrossRef]
82. Cirocki, R.; Gołębiewska, B. Changes in the profitability of production of industrial potatoes in Poland—A case study. *Ann. Polish Assoc. Agric. Agribus. Econ.* **2019**, *21*, 19–28. [CrossRef]
83. Licker, R.; Johnston, M.; Foley, J.; Barford, C.; Kucharik, C.; Monfreda, C.; Ramankutty, N. Mind the Gap: How Do Climate and Agricultural Management Explain the ‘Yield Gap’ of Croplands around the World? *Glob. Ecol. Biogeogr.* **2010**, *19*, 769–782. [CrossRef]
84. Hochman, Z.; Gobbett, D.; Holzworth, D.; McClelland, T.; van Rees, H.; Marinoni, O.; Garcia, J.; Horan, H. Reprint of “Quantifying Yield Gaps in Rainfed Cropping Systems: A Case Study of Wheat in Australia”. *Field Crops Res.* **2013**, *143*, 65–75. [CrossRef]
85. Harahagazwe, D.; Condori, B.; Barreda, C.; Bararyenya, A.; Byarugaba, A.; Kude, D.; Lung’aho, C.; Martinho, C.; Mbiri, D.; Nasona, B.; et al. How Big Is the Potato (*Solanum tuberosum* L.) Yield Gap in Sub-Saharan Africa and Why? A Participatory Approach. *Open Agric.* **2018**, *3*, 180–189. [CrossRef]
86. Campos, H.; Ortiz, O. The Potato Crop. In *Its Agricultural, Nutritional and Social Contribution to Humankind*; Springer: Berlin/Heidelberg, Germany, 2020; ISBN 978-3-030-28682-8.

87. Grassini, P.; van Bussel, L.; Van Wart, J.; Wolf, J.; Claessens, L.; Yang, H.; Boogaard, H.; de Groot, H.; van Ittersum, M.; Cassman, K. How Good Is Good Enough? Data Requirements for Reliable Crop Yield Simulations and Yield-Gap Analysis. *Field Crops Res.* **2015**, *177*, 49–63. [CrossRef]
88. Meroni, M.; Waldner, F.; Seguini, L.; Kerdiles, H.; Rembold, F. Yield Forecasting with Machine Learning and Small Data: What Gains for Grains? *Agric. For. Meteorol.* **2021**, *108555*, 308–309.
89. Dwivedi, S.; Goldman, I.; Ortiz, R. Pursuing the Potential of Heirloom Cultivars to Improve Adaptation, Nutritional, and Culinary Features of Food Crops. *Agronomy* **2019**, *9*, 441. [CrossRef]
90. Ahmad, U.; Sharma, L.A. Review of Best Management Practices for Potato Crop Using Precision Agricultural Technologies. *Smart Agric. Technol.* **2023**, *4*, 100220. [CrossRef]
91. Vetrovsky, T.; Siranec, M.; Marecnakova, J.; Tufano, J.; Capek, V.; Bunc, V.; Belohlavek, J. Validity of Six Consumer-Level Activity Monitors for Measuring Steps in Patients with Chronic Heart Failure. *PLoS ONE* **2019**, *14*, e0222569. [CrossRef]
92. Hara, P.; Piekutowska, M.; Niedbała, G. Prediction of Pea (*Pisum sativum* L.) Seeds Yield Using Artificial Neural Networks. *Agriculture* **2023**, *13*, 661. [CrossRef]
93. Al-Gaadi, K.; Hassaballa, A.; Tola, E.; Kayad, A.; Madugundu, R.; Alblewi, B.; Assiri, F. Prediction of Potato Crop Yield Using Precision Agriculture Techniques. *PLoS ONE* **2016**, *11*, e0162219. [CrossRef]
94. Li, D.; Miao, Y.; Gupta, S.; Rosen, C.; Yuan, F.; Wang, C.; Wang, L.; Huang, Y. Improving Potato Yield Prediction by Combining Cultivar Information and UAV Remote Sensing Data Using Machine Learning. *Remote Sens.* **2021**, *13*, 3322. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.



Article

The Development of a Weight Prediction System for Pigs Using Raspberry Pi

Myung Hwan Na ¹, Wan Hyun Cho ¹, Sang Kyoon Kim ¹ and In Seop Na ^{2,*}

¹ Department of Statistics, Chonnam National University, Gwangju 61186, Republic of Korea; nmh@jnu.ac.kr (M.H.N.); whcho@jnu.ac.kr (W.H.C.); narciss76@jnu.ac.kr (S.K.K.)

² Division of Culture Contents, Chonnam National University, Yeosu 59626, Republic of Korea

* Correspondence: ypencil@hanmail.net

Abstract: Generally, measuring the weight of livestock is difficult; it is time consuming, inconvenient, and stressful for both livestock farms and livestock to be measured. Therefore, these problems must be resolved to boost convenience and reduce economic costs. In this study, we develop a portable prediction system that can automatically predict the weights of pigs, which are commonly used for consumption among livestock, using Raspberry Pi. The proposed system consists of three parts: pig image data capture, pig weight prediction, and the visualization of the predicted results. First, the pig image data are captured using a three-dimensional depth camera. Second, the pig weight is predicted by segmenting the livestock from the input image using the Raspberry Pi module and extracting features from the segmented image. Third, a 10.1-inch monitor is used to visually show the predicted results. To evaluate the performance of the constructed prediction device, the device is learned using the 3D sensor dataset collected from specific breeding farms, and the efficiency of the system is evaluated using separate verification data. The evaluation results show that the proposed device achieves approximately 10.702 for RMSE, 8.348 for MAPE, and 0.146 for MASE predictive power.

Keywords: livestock measurement device; computer vision techniques; RGB-D sensor data; pig segmentation; body and shape feature extraction; prediction of pig weight

Citation: Na, M.H.; Cho, W.H.; Kim, S.K.; Na, I.S. The Development of a Weight Prediction System for Pigs Using Raspberry Pi. *Agriculture* **2023**, *13*, 2027. <https://doi.org/10.3390/agriculture13102027>

Academic Editors: Gniewko Niedbala, Sebastian Kujawa, Tomasz Wojciechowski and Magdalena Piekutowska

Received: 30 September 2023
Revised: 14 October 2023
Accepted: 17 October 2023
Published: 19 October 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Livestock rearing is an industry that significantly incurs costs compared to other industries due to the need for optimal barn management and ample feed grain provision. Furthermore, the demand for processed products derived from livestock is increasing as a result of population growth, improved income levels, and urbanization. Consequently, the global scale of livestock farming, including cattle, pigs, and poultry, is expanding to meet such demand. However, this extensive livestock rearing gives rise to various animal management issues such as diseases and environmental concerns. Additionally, the current practices of raising and managing animals rely heavily on traditional methods based on farmers' experiences and practices, which require substantial labor and time. Therefore, in order to address these challenges, there is a growing need for various advanced digital technologies based on artificial intelligence, which have been receiving significant attention in modernized livestock farms, to efficiently support livestock rearing.

Additionally, the real-time detection of livestock health status and abnormal behavior can contribute to enhancing job satisfaction among livestock farmers while simultaneously minimizing livestock production costs and effectively managing economic losses caused by diseases and mortality. Hence, there is a pressing need for livestock producers to invest in the development of cutting-edge technologies for real-time livestock monitoring, the implementation of advanced sensor systems, and the creation of streamlined processing systems to ensure the production of top-quality livestock products with reduced processing time. However, in most livestock farms, monitoring the condition, measuring the weight,

and observing the behavior or feeding behavior of livestock is usually performed by farmers raising livestock, either visually or manually. Such passive livestock management is laborious, expensive, and imposes stress on the livestock. Therefore, the applicability of computer vision technology and deep-learning algorithms in livestock operations is emerging as an important issue [1,2].

Here, we briefly review previous studies published on problems associated with breeding livestock. First, we consider several review papers that introduce comprehensive information on raising livestock. Femandes et al. [1] presented significant advancements and challenging research areas in computer vision systems that can be applied to animal breeding and commercialization. These technologies are expected to be utilized as high-value industrial applications in livestock farming. In their study, Oliveira et al. [2] conducted a comprehensive examination of recent breakthroughs in computer vision systems and their utilization of deep-learning algorithms in the field of animal science. They specifically highlighted various deep-learning algorithms employed for tasks such as image classification, object detection, object segmentation, and feature extraction. Notable algorithms discussed in their review included Mask R-CNN, Faster R-CNN, YOLO (v3 and v4), DeepLab v3, and U-Net, all of which have found applications in animal science research. Similarly, Wurtz et al. [3] presented a systematic overview of the advancements made in automated high-throughput image detection of farm animal behavioral traits, considering both welfare and production implications. They observed that several studies tended to start from scratch rather than build upon existing research despite there being significant overlap in the methods used for analyzing animal behavior. Nasirahmadi et al. [4] proposed a monitoring system based on images captured from 3D depth cameras to automatically identify and manage various behaviors of livestock, such as feeding, drinking, locomotion, aggression, and reproduction in cattle and pigs. They evaluated the performance of their developed system using metrics such as accuracy, error rate, specificity, and sensitivity. The system proposed by them is expected to contribute significantly to easily detecting abnormal behaviors of livestock in large-scale livestock farming operations.

Livestock such as cattle, pigs, and chickens are commonly raised animals in barns. In this study, we reviewed previous research on the prediction of livestock weight. Wang et al. [5] discussed the preexisting studies that focused on predicting the weight of livestock using techniques such as feature extraction, feature selection, and regression learning models, especially image analysis-based weight prediction methods. Kollis et al. [6] designed and implemented a program based on image analysis to estimate the weight of pigs, along with hardware. Li et al. [7] reviewed various methods for pig weight detection, comparing the structures and pros and cons of the proposed approaches. Kashiha et al. [8] developed an image-processing method that estimates pig weights by calculating the position and area within an ellipse using an ellipse-fitting algorithm. Shi et al. [9] proposed a system for analyzing the correlation between body length (BL), withers height (WH), and weight of 10 pig breeds aged between 14 and 25 weeks in indoor farming conditions (R^2 range of 0.91–0.98). Doeschi-Wilson et al. [10] compared the growth curves of two pig breeds between 11 and 20 weeks of age and suggested that size measurements are a consistent indicator of pig growth compared to body weight. Jun et al. [11] proposed a non-contact method for estimating pig weight using 2D images that are not influenced by pig posture or the capture environment. Suwannakhun and Daungmala [12] proposed a system that combines neural networks with various functions such as color and texture analysis, center calculation, measurements of major and minor axis lengths, eccentricity determination, and area calculation for non-contact pig weight estimation. Fernandes et al. [13] also developed a system that includes body measurements, shape descriptors, and weight prediction. Yoshida and Kawasue [14] introduced a weight estimation system that utilizes a camera to screen pigs. By utilizing three-dimensional visual information captured in a single image, they estimated the body weight of pigs. The researchers confirmed the robustness and practicality of the proposed system in measuring moving animals within challenging environments like pig farms. Kaewtapee et al. [15] proposed a method for

pig weight estimation employing image processing and artificial neural networks. They utilized images of 88 crossbred pigs (Large White, Landrace, Duroc Jersey) to individually measure heart circumference, body length, and weight. Regression analysis and artificial neural networks were employed to develop a pig weight equation, and the performance of the developed model was evaluated using mean absolute deviation (MAD) and mean absolute percentage error (MAPE) as estimation error metrics. Cang et al. [16] suggested an approach based on deep neural networks for estimating the live weights of pigs in sow stalls. They devised a neural network that takes top-view depth images of the pig's back as input and produces weight estimates as output. The proposed network, based on a Faster-RCNN architecture with an added regressive branch, integrates pig detection, location, and weight estimation into an end-to-end network, facilitating simultaneous pig recognition and weight estimation. Yu et al. [17] studied how to build a computer vision system with weight prediction over time by collecting RGB-D video to capture top-view RGB (red, green, blue) and depth images of unrestrained growing pigs. They video recorded eight growers at six frames per second for 38 days for approximately three minutes/day. They manually weighed them using electronic scales to collect training materials and developed the system using the image-processing pipes in Python based on OpenCV. He et al. [18] conducted a study to assess the predictive value of feeding behavior data in estimating the body weight of finishing-stage pigs. They collected data from 655 pigs belonging to three different breeds (Duroc, Landrace, and Large White) aged between 75 and 166 days. To forecast the body weight of pigs aged 159–166 days, the researchers employed LASSO regression along with two machine learning algorithms, namely random forest and long short-term memory network [19,20]. They explored four scenarios: the individual-informed predictive scenario, the individual- and group-informed predictive scenario, the breed-specific individual- and group-informed predictive scenario, and the group-informed predictive scenario. For each scenario, they developed four models: Model_Age included only the age variable, Model_FB included only feeding behavior variables, Model_Age_FB and Model_Age_FB_FI incorporated feeding behavior and feed intake measures based on Model_Age as predictors. And deep-learning methods, such as those used in automatic detection systems for livestock breeding, as seen in CowXNet [21] and YOLACT++ [22], are employed.

Among various strategies for effective livestock management, the real-time prediction of livestock weight plays a significant role in determining feed allocation, breeding timing, and market readiness. Additionally, measuring livestock weight can serve as an indirect indicator of their health and growth status. Furthermore, drastic changes in weight can indicate the presence of diseases, the provision of a healthy environment, and the optimal timing for parturition. Therefore, continuous monitoring of weight fluctuations in livestock facilitates the implementation of appropriate measures in terms of feed distribution and addressing abnormal behavior, ultimately leading to the production of high-quality livestock.

Consequently, livestock farmers are faced with the challenge of accurately measuring livestock weight on a regular basis, which proves to be a difficult and labor-intensive task for both farmers and livestock. Moreover, this process can induce excessive stress on the livestock, potentially resulting in weight loss. To address these issues and enable the continuous tracking of livestock weight while reducing stress and workload for farmers, the development of automated prediction systems using image processing and artificial intelligence technologies is required.

In this study, we propose a method for implementing a measurement prediction system that can automatically measure the weight of livestock in real time in livestock farms raising pigs. The proposed system acquires livestock images by mounting a depth camera capable of capturing RGB-D images, segmenting livestock from the input image using the Raspberry Pi module, extracting features from the segmented image, and predicting the weight of pigs. It is configured with an appropriate small-size monitor that can display images. Furthermore, the proposed prediction system is manufactured using cost-efficient parts to enable livestock farmers to purchase it with ease.

The remainder of this paper is organized as follows. Section 2 introduces the process of collecting livestock image data and describes the overall structure of the proposed system. Section 3 presents an experiment to evaluate the performance of the proposed system. Finally, Section 4 concludes this paper.

2. Weight Prediction System for Pigs Using Raspberry Pi

In this section, we describe the collection and construction method of pig image data of the automatic weight prediction device. First, the process of collecting RGB-D images from livestock farms of pigs is described. Second, we describe the overall structure of a device that can automatically predict pig weight using the collected data. Third, the hardware and software used to build the prediction device will be described in detail.

2.1. Pig-Image Collection

The prediction system proposed in this study can be used to predict the weight of pigs with the most realistic applicability among various types of livestock. Therefore, pig images required to implement the system were collected for a certain period of time from actual farms. We selected 15 pigs from a pig-breeding farm, repeatedly photographed them three times with the developed portable device, and measured the actual weight of each pig using a scale. The breed of pigs consists of three crossbred varieties, and they are raised in enclosed facilities isolated from the external environment. During the summer season, cooling systems are employed to maintain temperatures between 20 and 25 degrees Celsius, and continuous lighting is provided 24 h a day. Figure 1 shows a developed portable device that captures pig weights on site and RGB color and depth images.



Figure 1. Developed portable device and color and depth images for pigs in a field.

2.2. Automatic Prediction System

2.2.1. Overall Structure of the Prediction System

Figure 2 shows the overall structure of the proposed prediction system. The proposed system consists of livestock image capture and input, livestock segmentation, appropriate feature extraction, and weight prediction. First, livestock images are acquired using an RGB-D camera. Second, in the division of the livestock area from the collected image; the object is divided by applying the image-processing technology to each of the RGB color and depth images. Third, as features suitable for weight prediction, shape features representing the appearance of livestock and biometric features representing physical properties are extracted. Finally, we implement a predictive model that can accurately predict the weight of livestock based on the extracted features using various machine learning techniques.

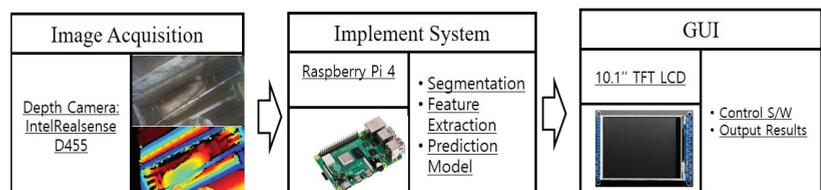


Figure 2. Overview of prediction system for pig weight.

2.2.2. Overall Structure of Implemented Hardware System

Figure 3 shows the overall structure of the hardware implementation of the prediction system. Figure 3a shows the modules for each part of the hardware system. First, an RGB-D depth camera was used to collect pig images. Second, a Raspberry Pi 4-embedded module was used to implement the prediction system. Third, a 10.1-inch touchscreen was used to visually express the prediction results. Figure 3b shows the product design of a portable imaging device packaged by combining each module from various angles.

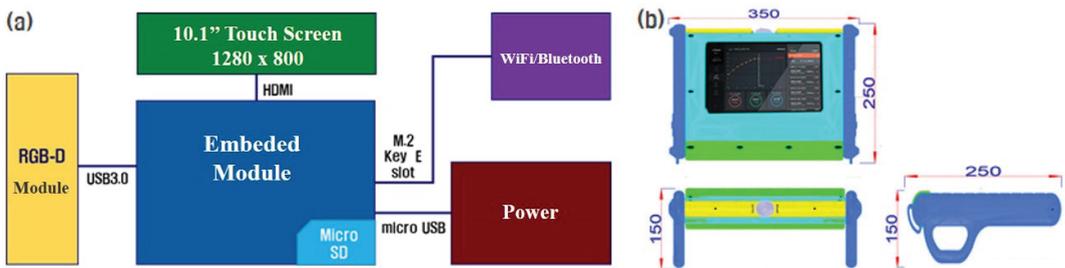


Figure 3. Overall structure of implemented hardware system: (a) internal structure of the system and (b) external structure of the system.

The detailed hardware module and software specifications used in the proposed portable measurement device are as follows.

- Depth Camera: Intel Real Sense D455;
- Raspberry Pi 4 Model B;
- Memory: 8 GB;
- CPU: 1.5 GHz Quad Core 64-bit Cortex-A72 (ARM v8);
- Giga-bit Ethernet;
- USB 3.0 \times 2, USB 2.0 \times 2;
- Dual 4 K Display (Micro-HDMI \times 2);
- Power: 1.5 V \times 4 EA Battery pack;
- Displayer: TFT 10" LCD;
- Wireless: dual-band 802.11ac wireless (2.4 GHz or 5 GHz);
- Bluetooth 5.0;
- Python 3.7;
- OpenCV 4.5 on Raspberry Pi 4.

2.2.3. Weight Prediction with GUI

Based on the hardware system, the weight prediction algorithm is implemented in the form of a GUI program for easy operation by users. GUI was developed based on QT, a cross-platform development widget toolkit that runs on various operating systems. The process of developing a platform that can predict the weight of a pig consists of the following four steps: detecting the pig region from the input image, segmenting the pig, extracting appropriate features from the segmented pig region, and obtaining the weight-prediction algorithm. Figure 4 shows the step-by-step process of constructing a platform for estimating pig weight.

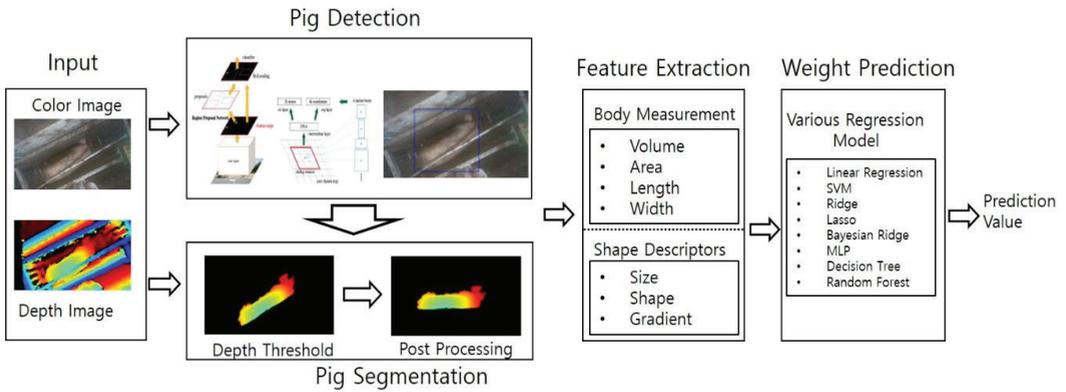


Figure 4. Platform for prediction of pig weight with GUI.

Here, we take a detailed look at the step-by-step processing of the proposed pig-weight prediction system. The first step is to detect the bounding box representing the pig region using a deep-learning algorithm from the input RGB color image. We used a single-shot multibox detector (SSD) [23] combined with the Inception V2 feature-extraction algorithm from RGB images to extract the bounding box. In the Single Shot Multibox Detector (SSD), the initial step involves partitioning the input images into smaller kernels across various feature maps to predict anchor boxes for each kernel. Through a single feed-forward Convolutional Neural Network (CNN), the SSD generates a sequence of bounding boxes and scores indicating the presence of objects within each box. Subsequently, the classification scores for each region are calculated based on the scores obtained in the preceding stage. Using the collected pig images, the SSD MobileNet was fine-tuned using Tensorflow object detection API to learn a deep-learning model for livestock detection [24]. Figure 5 shows an example image of the bounding box for the pig region provided by the SSD detector algorithm.

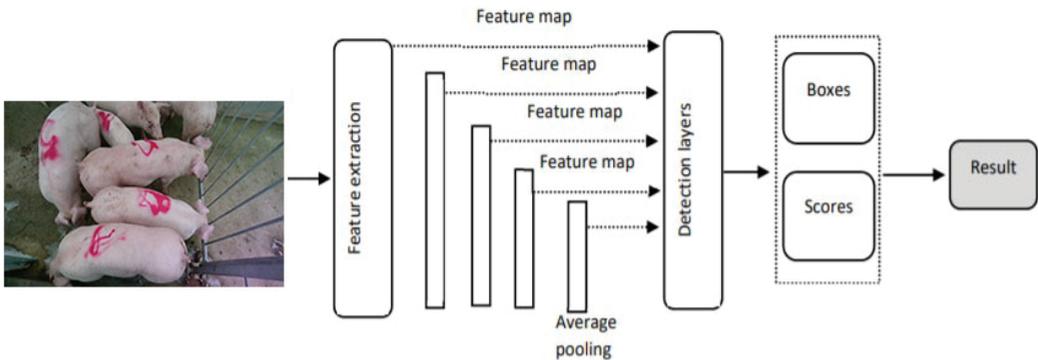


Figure 5. Bounding box for pig area provided by SSD detector algorithm.

The second step is to accurately segment the pig region by applying the information of the depth image to the pig image in the bounding box. We obtain the median value from the given depth information in the bounding box and select two thresholds: the lower and upper limits. Here, to detect the pig’s area as an RGB-D rectangular region, we set the distance from the camera to the ground as the upper limit and the distance from the camera to the pig as the lower limit in the depth image. If the depth of each pixel in the

image falls within the specified lower and upper thresholds, as indicated via Equation (1), a value of 255 is assigned; otherwise, a value of zero is assigned.

$$f(x,y) = \begin{cases} 255 & \text{if } T_a \leq d(x,y) \leq T_b \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

The image segmentation process based on depth images is illustrated in Figure 6. Pigs, especially young pigs, are curious; therefore, it is extremely difficult for them to remain in a position suitable for estimating their weight. Therefore, a method for adjusting the acquired pig image to an appropriate posture is required. We calculated the pig's center and rotation direction by fitting an ellipse to the segmented region to determine the pig's posture. Subsequently, the pig's posture was adjusted to be orthogonal to the camera using the fitted angle of the ellipse.

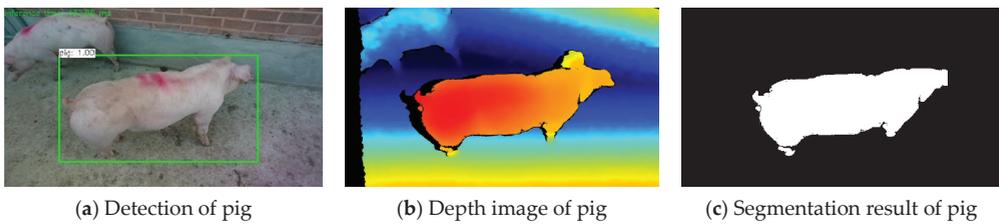


Figure 6. Segmentation process of pig from depth image.

In the third step, various feature vectors are extracted from the segmented 3D pig image, which is necessary for weight prediction. These feature vectors are divided into two types: pig body features and size and shape features. First, a pig's body features are given by the length, which is given as a straight line from the shoulder to the hip, and the girth, which is the circumference of the pig's waist. Figure 7 shows the process of calculating the length of the pig in the bounding box obtained from the single-shot detector and inferring the weight.

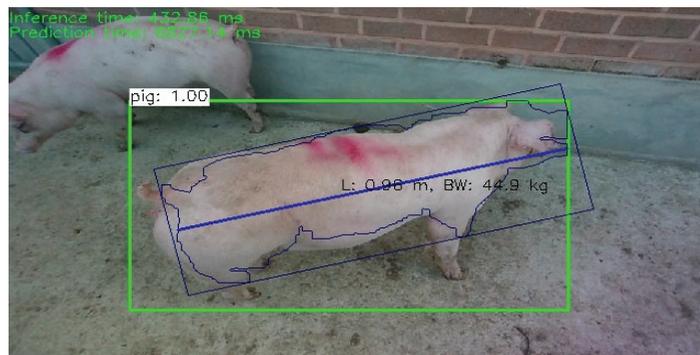


Figure 7. Process of calculating the length and girth of pig.

We obtain the size and shape features from the point cloud in the segmented pig image with additional information to infer a more accurate pig weight. Using the Intel RealSense SDK provided by the camera manufacturer, the image coordinates are mapped to real-world coordinates, resulting in the generation of a point cloud. Figure 8 illustrates the depth pig image and the corresponding point cloud coordinates for the pixels within the designated box.

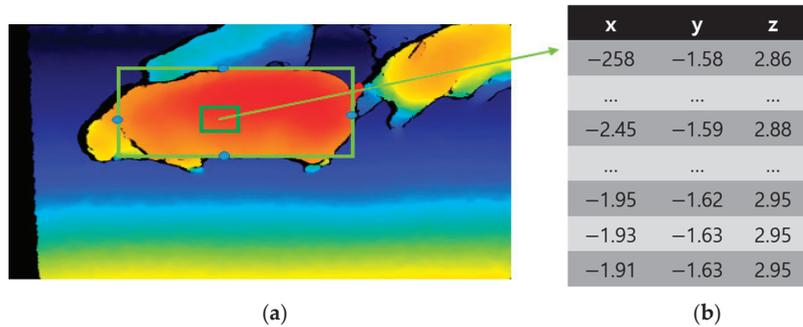


Figure 8. Point cloud data within a bounding box area. (a) Segmented pig image, (b) Three-dimensional coordinates for pixels in box region.

We transform depth images into 3D point clouds by assigning each pixel its corresponding 3D coordinate vector, which represents the size and shape of a pig. To capture the size cue, we calculate the Euclidean distance between each point and a reference point. The reference point is determined using evenly spaced basis vectors derived from the segmented image, with approximately 50 basis vectors in total. Let p_i , where i ranges from 1 to N , denote the i^{th} point cloud, and \bar{p} represent the reference point. The distance attribute of point p_i with respect to the reference point \bar{p} is given by $d_{p_i} = \| p_i - \bar{p} \|_2$. We construct an $N \times N$ distance matrix D_{p_i} . Next, we compute the top ten eigenvalues $(\lambda_1, \dots, \lambda_{10})$ and eigenvectors (E_1, \dots, E_{10}) from the distance matrix D in descending order of magnitude. The calculated eigenvalues are then normalized and utilized as the size descriptor for the pig.

$$F_{\text{Size}} = \left(\frac{\lambda_1}{\sum_{i=1}^{10} \lambda_i}, \dots, \frac{\lambda_{10}}{\sum_{i=1}^{10} \lambda_i} \right) \tag{2}$$

To calculate the shape features of the pig, we calculate the kernel distance for any two points p_i and p_j as follows and use them to create a kernel distance matrix K of size $(N \times N)$.

$$k_d(p_i, p_j) = \exp(-\gamma_k \| p_i - p_j \|^2), \quad (\gamma_k > 0) \tag{3}$$

By computing the kernel matrix K for the point cloud P and evaluating its top 10 eigenvalues, we conduct principal component analysis on the distance matrix K in the following manner.

$$Kv_l = \delta_l v_l \tag{4}$$

The eigenvectors are represented by v_l , and the corresponding eigenvalues are denoted by δ_l . After normalization, the calculated eigenvalues are used to define the kernel shape feature of the pig.

$$F_{\text{Shape}} = \left(\frac{\delta_1}{\sum_{i=1}^{10} \delta_i}, \dots, \frac{\delta_{10}}{\sum_{i=1}^{10} \delta_i} \right) \tag{5}$$

To capture gradient information in depth maps, we convert depth images into grayscale images. We apply the histogram of oriented gradient (HOG) feature extraction method to the depth image. In particular, we resize the segmented top-view image of the pig to (128×64) and divide it into (4×8) blocks of size (16×16) pixels. Each block is further divided into smaller cells of size (8×8) pixels. We calculate histograms of edge gradients with nine orientations from each local cell using Sobel filters. This results in a total of $1152 = 32 \times (4 \times 9)$ HOG features, which form a HOG feature vector. However, combining HOG features from all grid locations leads to a large number of features, so dimensionality reduction is necessary. Principal component analysis (PCA) is a commonly used technique

for this purpose. Given a set of M-dimensional feature $\{x_i, i = 1, \dots, N\}$, we compute the covariance matrix Σ as follows:

$$\Sigma = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})(x_i - \bar{x})^T \tag{6}$$

We performed PCA on the covariance matrix to obtain the top 10 eigenvalues $[\gamma_1, \dots, \gamma_{10}]$ and corresponding eigenvectors $[u_1, \dots, u_L]$.

$$\Sigma u_l = \gamma_l u_l, l = 1, \dots, 10. \tag{7}$$

After normalizing the calculated eigenvalues, we employed the gradient features of the pig as

$$F_{\text{Gradient}} = \left(\frac{\gamma_1}{\sum_{i=1}^{10} \gamma_i}, \dots, \frac{\gamma_{10}}{\sum_{i=1}^{10} \gamma_i} \right) \tag{8}$$

Figure 9 shows three kernel shape descriptors derived from arbitrarily selected pig images. By incorporating a prior distribution on the regression coefficient vector β using the Bayesian ridge regression model, we can leverage the posterior distribution of β for optimal estimation. In particular, we assume an independent normal distribution for each β_j , with a mean of zero and a variance of τ^2 , denoted, $\beta \sim \mathcal{N}(0, \tau^2 \mathbf{I})$, where τ is a constant. This formulation allows us to calculate the posterior distribution of β .

$$p(\beta|y, X) \propto p(\beta) \cdot p(y|X, \beta) \propto \exp \left[-\frac{1}{2} \beta^T \frac{1}{\tau^2} \mathbf{I} \beta \right] \cdot \exp \left[-\frac{1}{2} (y - X\beta)^T \frac{1}{\sigma^2} (y - X\beta) \right] \tag{9}$$

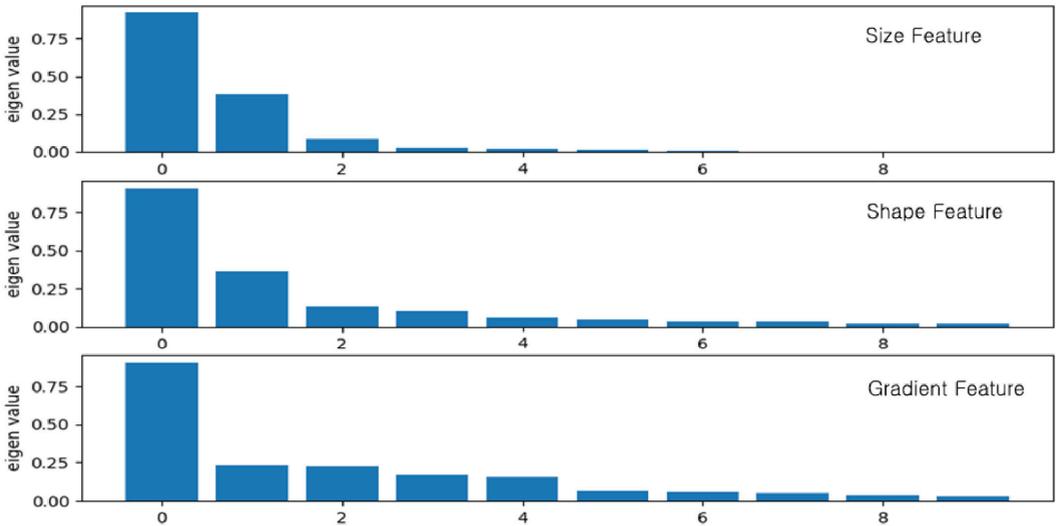


Figure 9. Three kernel shape descriptors derived from pig image.

From this expression, we can compute the mode of the posterior distribution, which is also known as the maximum a posteriori (MAP) estimate. It is given as follows.

$$\begin{aligned} \hat{\beta} &= \underset{\beta}{\operatorname{argmax}} \exp \left[-\frac{1}{2\sigma^2} (y - X\beta)^T (y - X\beta) - \frac{1}{2\tau^2} \|\beta\|_2^2 \right] \\ &= \underset{\beta}{\operatorname{argmin}} \left(\frac{1}{\sigma^2} (y - X\beta)^T (y - X\beta) + \frac{1}{\tau^2} \|\beta\|_2^2 \right) \\ &= \underset{\beta}{\operatorname{argmin}} \left((y - X\beta)^T (y - X\beta) + \frac{\sigma^2}{\tau^2} \|\beta\|_2^2 \right) \end{aligned} \tag{10}$$

The Bayesian ridge regression estimate can be obtained by setting $\lambda = \sigma^2/\tau^2$. In this way, Bayesian ridge regression can be seen as an extension of the Bayesian inference approach in general linear regression. Lastly, we utilized the estimated regression coefficients from the Bayesian ridge regression method to predict the weight of the pig using the following regression model.

$$\hat{y} = X\hat{\beta}$$

3. Experimental Results

To evaluate the performance of the portable measuring device, the predictive power was examined using the collected pig images. Figure 10 shows a two-dimensional scatterplot between the actual and predicted values. From the given scatterplot, it was confirmed that the proposed portable measuring device predicts the weight of pigs well.

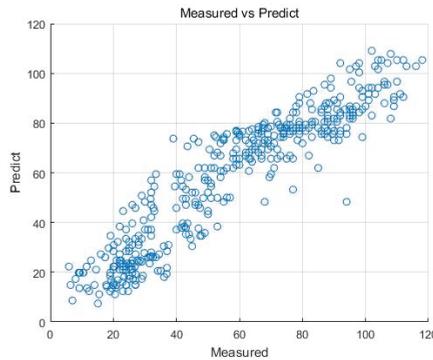


Figure 10. Scatterplot between actual and predicted values.

We calculated the correlation coefficient and the coefficient of determination between the predicted value and actual measurement to evaluate the extent to which the developed measuring device can predict the actual weight as presented in Table 1.

Table 1. Correlation and determination coefficient between the predicted and the actual measurements.

Prediction Index	Correlation Coefficient	Determination Coefficient
Value	0.9390	0.879

We calculated the root mean square error (RMSE), mean absolute predictive error (MAPE), and mean absolute scaled error (MASE) to determine the error between the predicted value and the actual observed weight using the proposed measuring device. Table 2 shows the measured values with respect to the three errors.

Table 2. RMSE, MAPE, and MASE between the predicted and actual measurements.

Prediction Index	RMSE	MAPE	MASE
Value	10.702	8.348	0.146

Finally, we evaluated how accurately the proposed prediction device can predict the actual weight of pigs from the various measures derived.

4. Conclusions

In this study, we proposed a portable device that can automatically predict the weight of pigs using Raspberry Pi with image-processing techniques and machine learning

methods. The proposed portable device is composed of two parts (hardware and software). The hardware part consists of a depth camera, microcontroller, and display device. The software part consists of algorithms for image segmentation, feature extraction, and weight prediction.

An experiment was conducted using RGB-D images of several pigs collected from specific farms to evaluate the performance of the proposed prediction system. From the experimental results, it was confirmed that the proposed system effectively predicted the actual weight of the pigs. Additionally, it was confirmed via various error measures that the prediction device did not significantly cause a difference between the predicted and measured values. Therefore, we believe that the proposed system will be useful for determining the release time of pigs in actual farms.

Still, there are limitations to the proposed system, the high activity levels of pigs make it more challenging to measure their weight accurately while they are feeding, and it is not easy to capture specific individuals separately when they move in groups. Additionally, environmental variables within the barn, such as changes in lighting, pose challenges in obtaining clean images. Since pigs are active, they have various poses. When measuring the weight, various movements of pigs cause deviations in measuring in the system. For a future study, we will upgrade the system that can measure robust pig weighting even in a natural daily breeding environment.

Author Contributions: The authors confirm their contribution to the paper as follows: study conception and design: M.H.N., W.H.C. and I.S.N.; data collection: S.K.K. and M.H.N.; analysis and interpretation of results: I.S.N., W.H.C. and M.H.N.; draft manuscript preparation: W.H.C. and I.S.N. All authors have read and agreed to the published version of the manuscript.

Funding: Korea Institute of Planning and Evaluation for Technology in Food, Agriculture and Forestry (IPET) and Smart Farm R&D Foundation (KosFarm) through the Smart Farm Innovation Technology Development Program, funded by the Ministry of Agriculture, Food and Rural Affairs (MAFRA) and Ministry of Science and ICT (MSIT), and Rural Development Administration (RDA) (421017-04).

Institutional Review Board Statement: Ethical review and approval were waived for this study due to the data collection through observational methods without direct interaction with animal subjects. Therefore, no separate ethical review or approval process was deemed necessary.

Data Availability Statement: The datasets generated or analyzed during this study are available from the corresponding author upon reasonable request, and partial data are not available due to commercial restrictions.

Acknowledgments: We deeply appreciate the assistance provided by Turnitin.com (accessed on September 2023) for conducting plagiarism checks.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Femandes, A.F.A.; Dorea, J.R.R.; Rosa, G.J.M. Image analysis and computer vision applications in animal sciences: An overview. *Front. Vet. Sci.* **2020**, *7*, 551269. [CrossRef] [PubMed]
2. Oliveira, D.A.B.; Pereira, L.G.R.; Bresolin, T.; Ferreira, R.E.P.; Dorea, J.R.R. A review of deep learning algorithms for computer vision systems in livestock. *Livest. Sci.* **2021**, *253*, 104700. [CrossRef]
3. Wurtz, K.; Camerlink, I.; D'Eath, R.B.; Fernandez, A.P.; Norton, T.; Steibel, J.; Siegford, J. Recording behavior of indoor-housed farm animals automatically using machine vision technology: A systematic review. *PLoS ONE* **2019**, *14*, e0226669. [CrossRef] [PubMed]
4. Nasirahmadi, A.; Edwards, S.A.; Sturm, B. Implementation of machine vision for detecting behavior of cattle and pigs. *Livest. Sci.* **2017**, *202*, 25–38. [CrossRef]
5. Wang, Z.; Shadpour, S.; Chan, E.; Rotondo, V.; Wood, K.M.; Tulpan, D. ASAS-NANP SYMPOSIUM: Application of machine learning for livestock body weight prediction from digital images. *J. Anim. Sci.* **2021**, *99*, skab022. [CrossRef] [PubMed]
6. Kollis, K.; Phang, C.S.; Banhazi, T.M.; Searle, S.J. Weight estimation using image analysis and statistical modeling: A preliminary study. *Appl. Eng. Agric.* **2007**, *23*, 91–96. [CrossRef]

7. Li, Z.; Luo, C.; Teng, G.; Liu, T. Estimation of pig weight by machine vision: A review. In *Computer and Computing Technologies in Agriculture VII, Proceedings of the CCTA 2013, Part II, IFIP AICT, Beijing, China, 18–20 September 2013*; Springer: Berlin/Heidelberg, Germany, 2014; Volume 420, pp. 42–49.
8. Kashiha, M.; Bahr, C.; Ott, S.; Moons, C.P.H.; Niewold, T.A.; Ödberg, F.O.; Berckmans, D. Weight estimation of pigs using top-view image processing. In *Image Analysis and Recognition, Proceedings of the ICIAR 2014, Part I, LNCS, Vilamoura, Portugal, 22–24 October 2014*; Springer: Berlin/Heidelberg, Germany, 2014; Volume 8814, pp. 496–503.
9. Shi, C.; Teng, G.; Li, Z. An approach of pig weight estimation using binocular stereo system based on LabVIEW. *Comput. Electron. Agric.* **2016**, *129*, 37–43. [CrossRef]
10. Doeschi-Wilson, A.B.; Whitemore, C.T.; Knap, P.W.; Schofield, C.P. Using visual image analysis to describe pig growth in terms of size and shape. *Anim. Sci.* **2016**, *19*, 415–427.
11. Jun, K.; Kim, S.J.; Ji, H.W. Estimating pig weights from images without constraint on posture and illumination. *Comput. Electron. Agric.* **2018**, *153*, 169–176. [CrossRef]
12. Suwannakhun, S.; Daungmala, P. Estimating pig weight with digital image processing using deep learning. In Proceedings of the 2018 14th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), IEEE Computer Society, Las Palmas de Gran Canaria, Spain, 26–29 November 2018; pp. 320–325.
13. Fernandes, A.F.A.; Dorea, J.R.R.; Fitzgerald, R.; Herring, W.; Rosa, G.J.M. A novel automated system to acquire biometric and morphological measurements and predict body weight of pigs via 3D computer vision. *J. Anim. Sci.* **2019**, *97*, 496–508. [CrossRef] [PubMed]
14. Yoshida, K.; Kawasue, K. Robust 3D pig measurement in pig farm. In *Computer Vision—ECCV 2018 Workshops, Proceedings of the ECCV 2018 LNCS, Munich, Germany, 8–14 September 2018*; Springer: Cham, Switzerland, 2018; Volume 11129, pp. 387–400.
15. Kaewtapee, C.; Rakangtong, C.; Bunchasak, C. Pig weight estimation using image processing and artificial neural networks. *J. Adv. Agric. Technol.* **2019**, *6*, 253–256. [CrossRef]
16. Cang, Y.; He, H.; Qiao, Y. An intelligent pig weights estimate method based on deep learning in sow stall environment. *IEEE Access* **2019**, *7*, 164867–164875. [CrossRef]
17. Yu, H.; Lee, K.; Morota, G. Forecasting dynamic body weight of non-restrained pigs from images using an RGB-D sensor camera. *Transl. Anim. Sci.* **2021**, *5*, txab006. [CrossRef] [PubMed]
18. He, Y.; Tiezzi, F.; Howard, J.; Maltecca, C. Predicting body weight in growing pigs from feeding behavior data using machine learning algorithms. *Comput. Electron. Agric.* **2021**, *184*, 106085. [CrossRef]
19. Hong, J.K. LSTM-based sales forecasting model. *KSII Trans. Internet Inf. Syst.* **2021**, *15*, 1232–1245.
20. Tai, D.N.; Na, I.S.; Kim, S.H. HSFE network and fusion model based dynamic hand gesture recognition. *KSII Trans. Internet Inf. Syst.* **2020**, *14*, 3924–3940.
21. Lodkaew, T.; Pasupa, K.; Loo, C.K. CowXNet: An automated cow estrus detection system. *Expert Syst. Appl.* **2023**, *211*, 118550. [CrossRef]
22. Yang, G.; Li, R.; Zhang, S.; Wen, Y.; Xu, X.; Song, H. Extracting cow point clouds from multi-view RGB images with an improved YOLACT++ instance segmentation. *Expert Syst. Appl.* **2023**, *230*, 120730. [CrossRef]
23. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 8–16.
24. Zhao, L.; Wang, L. A new lightweight network based on MobileNetV3. *KSII Trans. Internet Inf. Syst.* **2022**, *16*, 1–15. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.



Article

Modeling of Path Loss for Radio Wave Propagation in Wireless Sensor Networks in Cassava Crops Using Machine Learning

Alexis Barrios-Ulloa ¹, Alejandro Cama-Pinto ^{2,*}, Emiro De-la-Hoz-Franco ², Raúl Ramírez-Velarde ³ and Dora Cama-Pinto ^{4,5,*}

¹ Department of Electronic Engineering, Universidad de Sucre, Sincelejo 700001, Colombia; alexis.barrios@unisucre.edu.co

² Department of Computer Science and Electronics, Universidad de la Costa, Barranquilla 080002, Colombia; edelahoz@cuc.edu.co

³ School of Engineering and Sciences, Instituto Tecnológico y de Estudios Superiores de Monterrey, Monterrey 64849, Mexico; rramirez@tec.mx

⁴ Faculty of Industrial Engineering, Universidad Nacional Mayor de San Marcos, Lima 15081, Peru

⁵ Department of Computer Architecture and Technology, University of Granada, 18071 Granada, Spain

* Correspondence: acama1@cuc.edu.co (A.C.-P.); doracamapinto@correo.ugr.es (D.C.-P.)

Abstract: Modeling radio signal propagation remains one of the most critical tasks in the planning of wireless communication systems, including wireless sensor networks (WSN). Despite the existence of a considerable number of propagation models, the studies aimed at characterizing the attenuation in the wireless channel are still numerous and relevant. These studies are used in the design and planning of wireless networks deployed in various environments, including those with abundant vegetation. This paper analyzes the performance of three vegetation propagation models, ITU-R, FITU-R, and COST-235, and compares them with path loss measurements conducted in a cassava field in Sincelejo, Colombia. Additionally, we applied four machine learning techniques: linear regression (LR), k-nearest neighbors (K-NN), support vector machine (SVM), and random forest (RF), aiming to enhance prediction accuracy levels. The results show that vegetation models based on traditional approaches are not able to adequately characterize attenuation, while models obtained by machine learning using RF, K-NN, and SVM can predict path loss in cassava with RMSE and MAE values below 5 dB.

Keywords: agriculture; cassava crops; machine learning; radio wave propagation models; wireless sensor networks

Citation: Barrios-Ulloa, A.;

Cama-Pinto, A.; De-la-Hoz-Franco, E.; Ramírez-Velarde, R.; Cama-Pinto, D. Modeling of Path Loss for Radio Wave Propagation in Wireless Sensor Networks in Cassava Crops Using Machine Learning. *Agriculture* **2023**, *13*, 2046. <https://doi.org/10.3390/agriculture13112046>

Academic Editors: Gniewko Niedbala, Sebastian Kujawa, Magdalena Piekutowska and Tomasz Wojciechowski

Received: 15 September 2023

Revised: 10 October 2023

Accepted: 18 October 2023

Published: 25 October 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Agriculture is an activity of great relevance in several countries, and in many of them, it is considered a priority within their national security policies because of its importance in sustaining the population [1]. Like other industrial sectors, agriculture has undergone a significant evolution in the last two decades, transitioning from a model based on variable monitoring to one of greater autonomy and automation in cultivation. This evolution can encompass the entire production chain, from planting to the final marketing stage [2]. In this regard, precision agriculture (PA) is one of the most significant concepts in the modern agricultural industry, owing to the adoption of management strategies that enable the collection, processing, and analysis of data. All of this is aimed at enhancing the sustainability of production in the cultivation fields [3]. To achieve this, it makes use of different electronic tools and information and communication technologies (ICT), including machine learning (ML) and wireless sensor networks (WSN), which have been increasingly used in the agricultural sector in recent years [4,5].

The implementation of WSNs in agriculture allows farmers to remotely monitor large areas of the plantation, providing information that improves decision-making and

optimizes the economic performance of the activity [6]. However, the efficient transmission of the data obtained depends to a large extent on the correct deployment of the motes in the field [7], also allowing infrastructure costs not to increase because of a higher-than-required node usage or, on the contrary, inefficient coverage in the field because of a lower number of motes than needed [8]. Hence, it is necessary to efficiently model the radio signal losses caused by the path and the environment in which the WSN infrastructure is deployed [9]. Based on the above, WSN planning demands the utilization of suitable propagation models that optimize the deployment of sensor nodes in the field. Each scenario with vegetation presence (e.g., crop fields) possesses unique propagation characteristics that impact radio signal attenuation. In addition to frequency and distance, which are generally considered the main attenuating factors of wireless signals, other factors stand out, such as plant height and the presence of leaves [10].

Various radio wave propagation models have been developed for use in crop environments or where vegetation is present. Examples include those based on the MED (exponential decay model), among which stand out the ITU-R (recommended by the International Telecommunication Union), the FITU-R (ITU-R adjusted), and COST-235 (Cooperation in Science and Technology 235). Despite their recognition and widespread use, their implementation can produce results with high levels of error [5,11].

1.1. Background

Cassava (*Manihot esculenta*) is nowadays considered a very relevant product to ensure food sustainability. It is grown in more than 100 countries and is reputed to be an excellent and inexpensive source of nutrition [12]. Additionally, it serves as input in the industrial production of starch, alcohol, and fermented beverages [13]. Regarding the implementation of ICTs and electronic tools, there are not many specific developments for this type of cultivation. In [14], they implemented an agronomic variable monitoring system using WSN in a municipality located in the northern region of Colombia, while other studies focused on leaf disease detection through deep learning [15] or deep neural networks [16]; however, only in [17] do they utilize sensors and the Internet of Things (IoT) as part of a comprehensive solution aimed at PA.

Regarding radio wave propagation, the systematic literature review does not show any evidence of comparative studies or the development of propagation models in cassava plantations. However, these investigations have been conducted in a wide variety of crops; such is the case with corn [6], rice [18], sugarcane [19], banana [20], oil palm [21], citrus [22], and various greenhouse products like cucumber [23] and tomato [8].

In relation to the techniques used in the development of propagation models, there is a growing trend in the use of ML, which is one of the most representative and powerful subfields of artificial intelligence. ML can identify patterns and detect correlations between variables in a large dataset [24]. Some examples applied in research related to characterizing behavior using these types of techniques in vegetation-present environments have considered settings with a predominance of grasslands [19], forests [25], or greenhouse tomato crops [26].

1.2. Motivation and Objectives

This paper presents the results of a radio channel measurement campaign in a cassava crop on a farm near the city of Sincelejo, Sucre, Colombia. Based on the data collected, traditional vegetation models were compared to assess their effectiveness in cassava plantations. As mentioned in the previous section, there are no prior records of similar studies in this type of cultivation. Only [14] obtained received signal strength indicator (RSSI) values but did not characterize the attenuation caused by the plants. Given the above and considering the significance of cassava cultivation, it is necessary to understand the radio signal propagation behavior in these plantations, which will facilitate the future implementation of AP-oriented technologies. The main contributions of this research are as follows:

- We have analyzed the effectiveness of widely recognized vegetation propagation models based on measurements in a real scenario, leaving aside the simulations chosen by some studies;
- We present the results of the modeling of radio wave propagation in cassava crops obtained by ML from data collected in a cassava crop.

The rest of the paper is organized as follows: Section 2 presents the methodology and materials used in obtaining and analyzing the data; Section 3 shows the comparative analysis between different models and the results of the ML propagation modeling; finally, Section 4 contains the discussion.

2. Materials and Methods

This section first presents the vegetation loss models considered in the comparative analysis to determine their effectiveness in predicting attenuation in cassava crops. Below, we describe the testing environment, as well as the tools used in both the measurement and evaluation of the results. Finally, we present an overview of the ML techniques used in propagation modeling.

2.1. Vegetation Propagation Models

There are a large number of models of radio wave propagation. They can be obtained empirically, stochastically, or deterministically. When transeiver equipment is in scenarios with significant vegetation or foliage presence (e.g., forests, crops, or gardens), it is important to consider their effect on attenuation. Additionally, there are significant differences in the arrangement of signal-obstructing elements in various environments, as well as their physical and geometrical characteristics, preventing the use of a generalized method for predicting losses [27]. However, there are recognized and widely used models for estimating attenuation caused by vegetation.

The COST-235 model is derived from the MED and is one of the most widely used models to characterize the signal attenuation caused by bushes in radio propagation. One of its main features is to consider the presence or absence of leaves on plants. The equation for this model is given in Equation (1) [28]:

$$PL_{COST-235}(dB) = \begin{cases} 15.6 \times f^{-0.009} \times d^{0.26} & \text{on the leaves} \\ 26.6 \times f^{0.2} \times d^{0.5} & \text{out of the leaves} \end{cases} \quad (1)$$

where f is the operating frequency given in MHz, and d is the distance between the transmitting antenna (Tx) and the receiving antenna (Rx) in meters. A relevant condition of this model is that it can be used for distances of up to 200 m [29]. Like the COST-235 model, the ITU-R model is derived from the MED and is widely used to model radio channel losses in vegetated environments. It is suggested to use this when antennas are located near bushes and a significant part of the signal propagates through the foliage [30]. Equation (2) expresses the losses of this model [31].

$$PL_{ITU-R}(dB) = 0.2 \times f^{0.3} \times d^{0.6} \quad (2)$$

where f is the frequency in MHz, and d is the distance in meters. In this model, the distance between Tx and Rx should not exceed 400 m [31]. A modification of the ITU-R model is the ITUF-R model, derived from measurements in the 11.2 GHz to 20 GHz range. It takes into account the presence of leaves in the bush [32]. Equation (3) shows this [33]:

$$PL_{FITU-R}(dB) = \begin{cases} 0.39 \times f^{0.39} \times d^{0.25} & \text{on the leaves} \\ 0.37 \times f^{0.18} \times d^{0.59} & \text{out of the leaves} \end{cases} \quad (3)$$

where f is expressed in MHz and d in meters. Like ITU-R, the adjusted model considers a maximum distance of 400 m between Tx and Rx. Another model widely used in this type

of study is free space path loss (FSPL). This is not a vegetative model, and it assumes ideal propagation. Equation (4) allows the calculation of free space losses [34].

$$FSPL(dB) = 32.44 + 20 \log d + 20 \log f \tag{4}$$

where distance is expressed in kilometers and frequency in Megahertz.

2.2. Measuring Equipment and Methodology

To analyze the propagation characteristics of the wireless channel, RSSI data were collected in a bitter cassava crop, which is mainly used as a raw material for starch production. The measurements were taken on a farm in the rural area of Sincelejo, capital of the department of Sucre, one of the largest producers in Colombia. The measurements were conducted on sunny days during the second planting period (August to March) at coordinates 9°2'53" N latitude and 75°25'33" W longitude (see Figure 1).

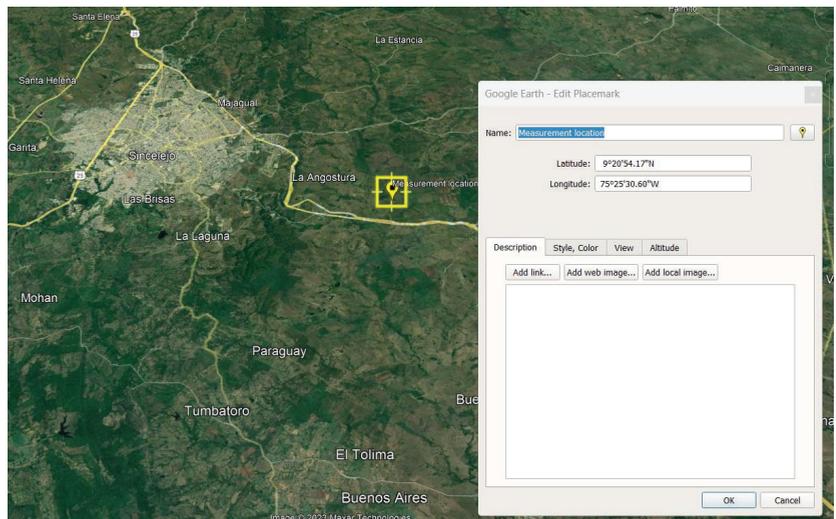


Figure 1. Geographical location of the measurement site.

The cassava bushes were planted in a commonly used arrangement for this type of crop, with 120 cm between rows and 93 cm between plants. Figure 2 depicts the positioning of bushes in the selected cultivation. The measurement campaign was carried out using Digi Xbee XK3-Z8S-WZM transceiver modules operating on the IEEE 802.15.4 standard, capable of functioning as both Tx and Rx in the 2400 MHz band. The antennas at both ends are omnidirectional, which are commonly used in WSN deployments in agricultural environments [8]. The system parameters are presented in Table 1.

Table 1. System parameters.

Parameter	Value
Frequency	2400 MHz
Antenna gain	2.1 dBi
Transmission power	8 dBm
Receiver sensitivity	−103 dBm
Antenna type	Omnidirectional



Figure 2. Arrangement of cassava plants in the crop.

Data were collected and stored during three stages of bush growth. In each round, the Tx and Rx antennas were adjusted to heights of 80 cm, 130 cm, and 180 cm, with no obstructions other than the crop itself. The transmitting antenna was positioned at a fixed point, and the receiver was moved in 5 m increments, with equal antenna heights at each step. At each measurement point, 20 RSSI values were obtained and averaged, and additionally, tests were conducted with the transmission of 100 packets at intervals of 1000 ms. Because the spaces between rows of plants are commonly used as pathways by the individuals responsible for the plantation, the nodes were positioned along the rows of bushes. The receiver was moved until it reached the maximum distance between the rows of the crop, which in this case was 50 m, or until packet losses were recorded. The information obtained was stored on a computer connected to the transmitter module via USB. Figure 3a,b show a diagram of the applied set-up and the arrangement of the equipment in the culture.

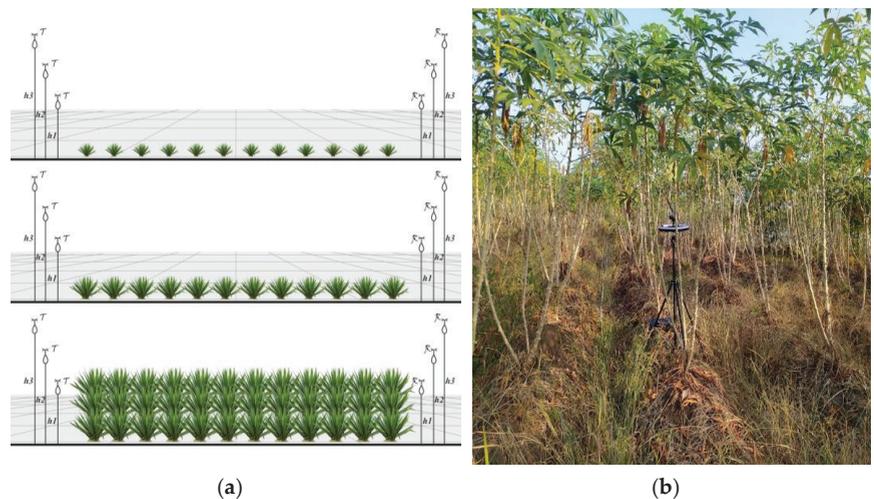


Figure 3. (a) Schematic of the measurement system in place and (b) arrangement of the cassava plants in the crop.

Since the geographical area where the measurements were taken has limited mobile phone coverage by the operators providing service in the region, it was decided not to upload the data to any cloud platform. Instead, the data were backed up and isolated on an external memory stick, thus ensuring data protection.

2.3. Machine Learning Techniques

In machine learning, there are mainly two learning methods: supervised and unsupervised. The former requires initial data to detect patterns, while the latter learns the variable space without the use of labeled data [35]. In this research, we calculate the radio signal loss between a transmitter and a receiver in a cassava crop using a supervised approach with techniques that determine the attenuation from a set of measurements with varying values of distance, plant height, and antenna height.

One of the simplest and most widely used methods in supervised learning is linear regression (LR), where the prediction is made from the weighted sum of the input features and a constant called the intercept, as shown in Equation (5).

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n, \quad (5)$$

where y is the variable to predict, β_i are the model parameters, x_i is the i -th value of the features, and n is the number of features. The goal when training an LR model in ML is to find a value beta that minimizes the error [36]. Another advantage of LR is that models using this technique are less likely to overfit the data. However, as a drawback, it exhibits high sensitivity to outliers.

Support vector machine (SVM) is another regression technique widely used in ML but more advanced than LR. In SVM, the goal is to fit a hyperplane in a high-dimensional space that maximizes the distance to the nearest training data points of any class [37]. The hyperplane in the feature space can be described as the linear function shown in Equation (6) [38].

$$f(x) = w^T x + b, \quad (6)$$

where w corresponds to the vector determining the direction of the hyperplane, x is the input feature vector, and b is the bias term. The prediction function is presented in Equation (7).

$$f(x) = \sum_{i=1}^N (a_i - a_i^*) K(x_i, x) + b, \quad (7)$$

where a_i and a_i^* are the Lagrange multipliers, K represents the kernel used in the mapping, and i represents the sample number. Unlike other regression techniques, SVM is less sensitive to outliers in the data [39] but is dependent on the kernel used and data scaling [38]. Due to its advantages and effectiveness, SVM is a supervised learning technique successfully used in predicting path loss in various scenarios [40].

K-nearest neighbors (K-NN) is a technique that can be used in both regression and classification problems. It makes predictions on test vectors based on the learning obtained from training vectors [41]. K-NN uses distance functions as metrics to calculate the mean of the numerical target of the k -nearest neighbors. Equations (8)–(10) show the distance functions used in K-NN [42].

$$\text{Euclidean} : D = \sqrt{\sum_{i=1}^k (x_i - y_i)^2}, \quad (8)$$

$$\text{Manhattan} : D = \sum |x_i - y_i|, \quad (9)$$

$$\text{Minkowski : } D = \left(\sqrt[q]{\sum_{i=1}^k (|x_i - y_i|)^q} \right)^{\frac{1}{q}}, \quad q \geq 1, \quad (10)$$

where x is the measured value, and y is the predicted value. Although K-NN is a method that can yield good results when fitting the data, the computational effort increases as the size of the data used grows [42].

The fourth technique considered in this research is random forest (RF). It is a non-parametric approach used in regression and classification tasks and was developed as an improvement over decision trees [43]. Instead of using a single tree, random forest (RF) creates a ‘forest’ of random trees, each trained on a random sample of the data and a random selection of features. The predictions from all the trees are used to generate a final prediction [40,44]. Each new prediction at a point x with RF is obtained by taking the mean value of the predictions of each tree, $\hat{f}_1 \dots \hat{f}_{N_{Tree}}$, as shown in Equation (11) [45].

$$\hat{f}_{RF}(x) = \frac{1}{N_{Tree}} \sum_{K=1}^{N_{Tree}} \hat{f}_K(x), \quad (11)$$

These techniques, along with others in the field of ML, can be considered valid alternatives when modeling radio wave propagation in different scenarios. However, since no validation method allows the best one to be selected in advance, it is common practice to try different algorithms to see which one performs better for a given problem [41]. Therefore, due to the amount of data and variables used in the training process, as well as the ease of implementation, interpretation, and computational efficiency, the following algorithms have been chosen for this research: LR, SVM, K-NN, and RF.

3. Results

3.1. Comparison of Models

We evaluated the effectiveness of the vegetative models presented in Section 2.1 by comparing them with results obtained from the measurement campaign. Since it is preferable to monitor the crop from the beginning of the planting process, measurements were taken at three different stages of average plant growth (50 cm, 150 cm, and 190 cm). The obtained RSSI values indicate that the radio signal range was greater when the antennas were configured at heights of 180 cm for plant sizes of 50 cm and 150 cm, as they exhibited line-of-sight (LoS) conditions. However, during the stage of maximum bush growth and with the same configuration of experimental conditions applied in the first two measurement stages, the best coverage was achieved with antennas placed 80 cm above ground level. The path loss was calculated from the RSSI measurements at each measurement point using Equation (12).

$$PL(dB) = P_{Tx} + G_{Tx} + G_{Rx} - P_{Rx}, \quad (12)$$

where P_{Tx} is the transmission power, G_{Tx} and G_{Rx} are the gains of the transmitter and receiver antennas, respectively, and P_{Rx} is the received power at the receiver determined from the RSSI. Since the vegetative models only consider the loss caused by foliage and not the total loss along the link, the analysis was complemented with the free space path loss (FSPL) model to calculate the total attenuation along the path [46,47]. Figure 4a–c depict the behavior of the measured losses concerning the measurement distance in each of the three stages of cultivation where measurements were taken, along with the estimations obtained using the vegetative models.

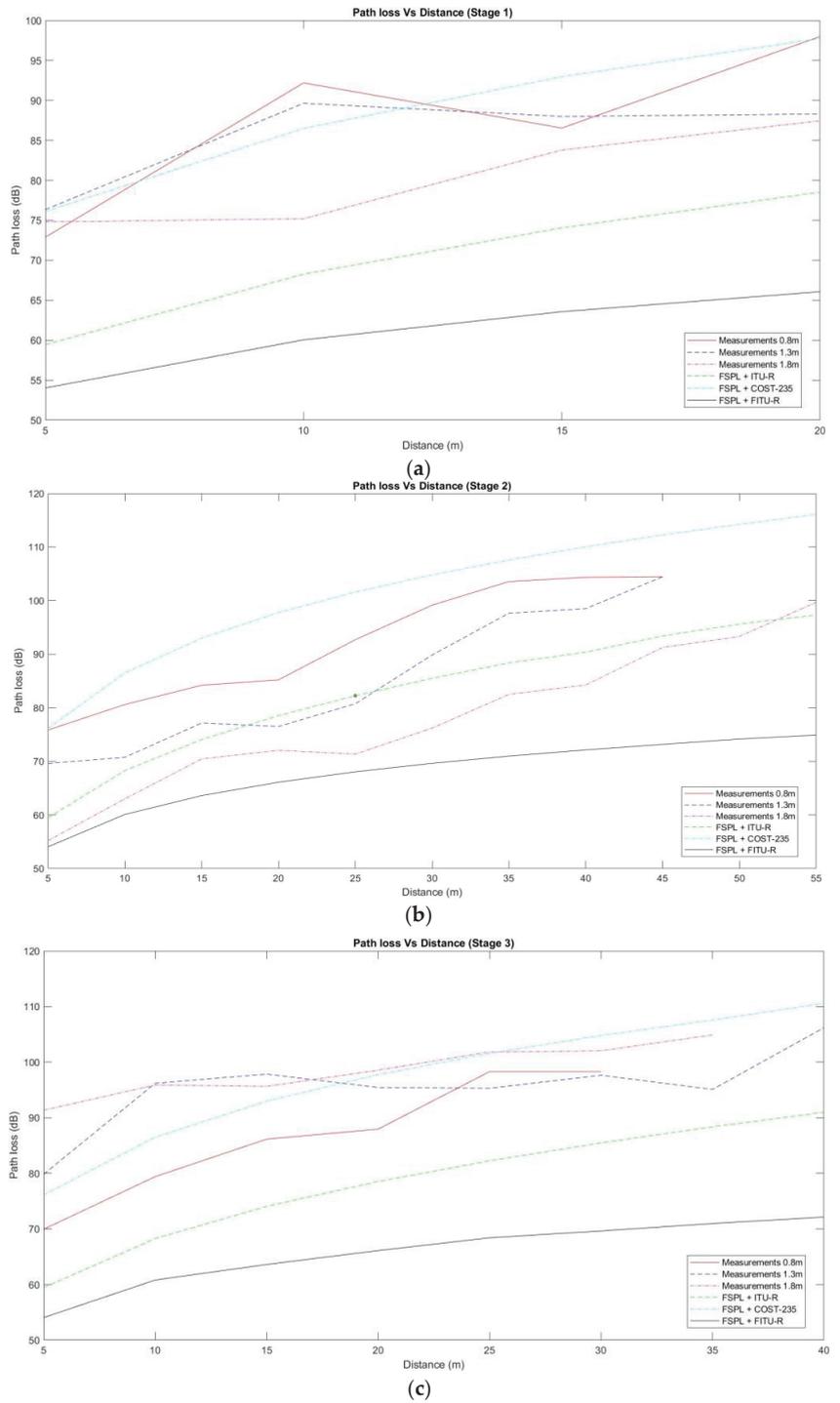


Figure 4. Path loss as a function of distance with bush heights of (a) 50 cm, (b) 150 cm, and (c) 190 cm.

Additionally, the root mean square error (RMSE) was calculated, which is one of the most used tools for evaluating the effectiveness of radio wave propagation models. This is presented in Equation (13).

$$RMSE = \sqrt{\frac{1}{N} \times \sum_{i=1}^N (x_i - \hat{x}_i)^2}, \tag{13}$$

where x_i and \hat{x}_i are the measured and predicted values, respectively, and N corresponds to the number of samples. In this case, the higher the RMSE value, the lower the ability of the models to accurately predict actual crop losses.

Figure 4a–c, along with the obtained RMSE, indicate that in all measurement stages and for each antenna height configuration, the FSPL+ITU-R combination underestimated the measurements, reaching minimum RMSE values of 19.99 and maximum values of 29.86. For FSPL+ITU-R, it underestimated the losses in stages 1 and 3, but the RMSE decreased to values of 8.72 and 19.05, respectively. Meanwhile, in stage 2, it overestimated the measurements when the antennas were placed at 80 cm and 130 cm above the ground and underestimated them at 190 cm, giving an RMSE of 15.21. Finally, FSPL+ COST-235 exhibited the best performance in stages 1 and 3, achieving the lowest error values in estimating the losses. It showed a trend of losses with values higher than the measurements and RMSE of 6.81 and 6.39, respectively. However, in stage 2, the estimation error increased, reaching a value of 17.4. These results show that the propagation and the range of the modules are affected by the modifications of the different experimental conditions that have been proposed. Furthermore, the vegetative models considered in this study do not estimate cassava crop losses with error levels considered acceptable. Therefore, it is necessary to adjust or develop a model for this specific environment.

3.2. New Model

In this study, we calculated the loss of radio signals between a transmitter and a receiver in a cassava crop using a supervised approach with algorithms that determine attenuation based on a series of measurements at different distances, plant heights, and antenna heights. The correlation matrix with the contribution of each attribute considered in the study is shown in Figure 5. It is observed that losses in cassava crops are highly affected by clearing losses (considering frequency and distance), followed by cassava bush height and antenna height.

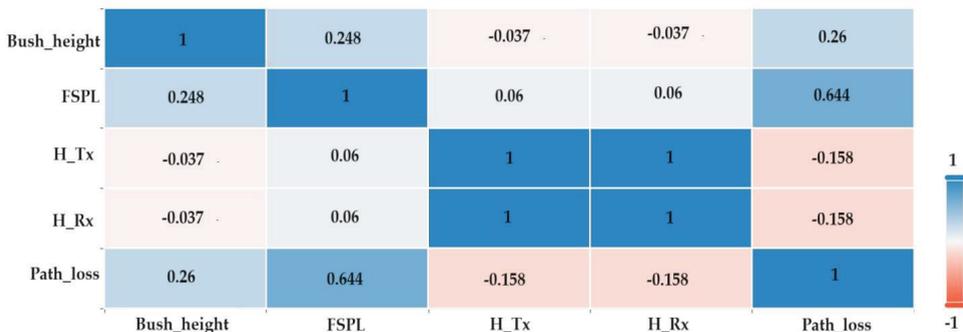


Figure 5. Correlation matrix.

In the proposed methodology, we selected 70% of the dataset for training and 30% for testing, applying this procedure with the evaluated techniques LR, K-NN, SVM, and RF. In the case of K-NN, the used kernel was optimal with k-neighbors equal to 16, while in SVM, the implemented kernel is a radial basis function, with data scaling. Regarding RF, we adjusted the number of trees to 20. Taking into account the evaluation metrics

used in similar studies [38,48–50], in this work, we have considered the RMSE presented in Section 3, as well as the mean absolute error (MAE) and the coefficient of determination (referred to as R^2), calculated using Equations (14) and (15), respectively.

$$MAE = \frac{1}{n} \sum_{i=1}^n x_i - \hat{x}_i, \quad (14)$$

$$R^2 = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{\sum_{i=1}^n (\hat{x}_i - \bar{x})^2}, \quad (15)$$

where x_i , \bar{x} , and \hat{x}_i are the measured values, mean of the values, and predicted values, respectively. Additionally, n represents the number of samples. In the case of RMSE and MAE, the models exhibit a better fit of the data as they tend toward zero, while in the case of R^2 , it indicates that it is capable of accurately representing the data as it tends toward 1.

The performance of the models obtained from the use of LR, K-NN, SVM, and RF with the training and testing dataset is presented in Table 2. The results of the metrics indicate that the models obtained from RF, K-NN, and SVM predict path losses in the studied crop with low levels of error, significantly outperforming the vegetative models UIT-R, FITU-R, and COST-235. However, when using LR as an algorithm in ML, the error values increase considerably, making it the technique with the poorest performance.

Table 2. Performance comparison of ML models.

Model	RMSE	MAE	R^2
LR	8.81	7.29	0.4650
K-NN	2.62	1.76	0.9575
SVM	2.66	1.85	0.9507
RF	2.51	1.78	0.9604

Further analysis of the results reveals that RF offers the best performance ($RMSE = 2.51$, $MAE = 1.78$, and $R^2 = 0.9604$). The evaluation metrics demonstrate that K-NN can perform regression based on the training data, with minimal variation compared to the results obtained with RF. As for the SVM using the radial basis function, it is evident that it is also capable of accurate path loss predictions in the study scenario, keeping the $RMSE$ and MAE values below 5 dB and an R^2 slightly above 95%. In addition, Figure 6a–d show the scatter plots of the values, showing the difference between the expected prediction and the results obtained with the models. Black lines represent ideal predictions for each model, while red dots represent predicted path loss values. In the case of LR, the predictor exhibits a negative bias when the path loss values exceed 94 dB and a positive bias around 85 dB. Regarding K-NN, SVM, and RF, it demonstrates a bias very close to zero but becomes positive for path loss values around 88 dB.

Although the evaluation metrics indicate that RF gives better results than SVM, they are not significantly different. However, since SVM has proven to be a superior method compared to other supervised learning techniques [47], it could be considered a preferable alternative in the modeling of WSN systems in crops, as it is less computationally demanding than RF. Regarding LR, it is a simple method that generally yields good results. Nevertheless, in this study, it turned out to be the least performing one with $RMSE$ or MAE values too high, and given the goodness of fit below 70%, its application in modeling path losses in the studied crop is not considered useful.

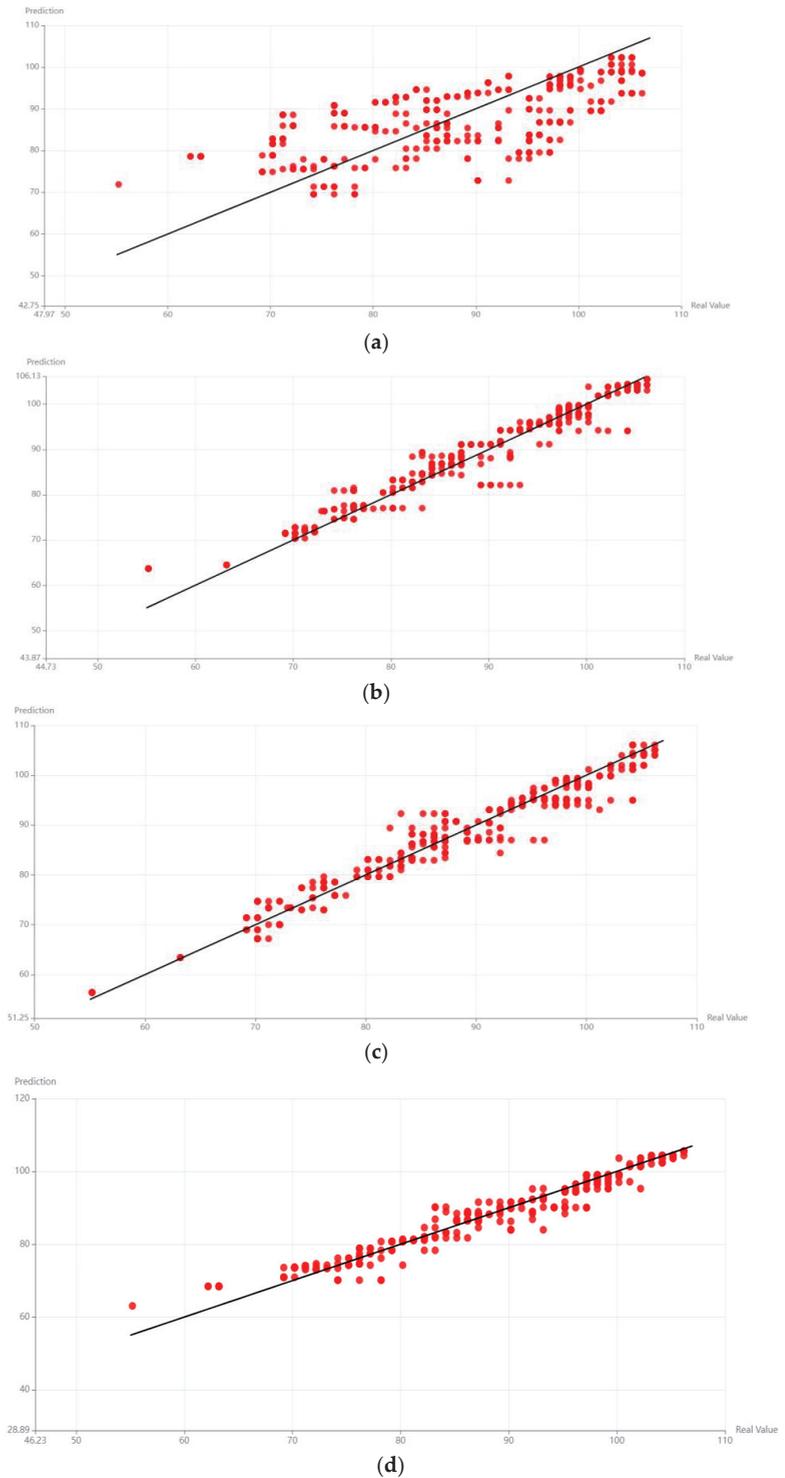


Figure 6. Scatter plots: (a) linear regression, (b) K-NN, (c) SVM, and (d) RF.

4. Discussion

In this paper, we compare three vegetative propagation models (ITU-R, FITU-R, and COST-235) with path loss data from a cassava crop. For this purpose, we carried out measurements at three stages of crop growth and under different conditions of distance and height of the transmitting and receiving antennas. The results demonstrate that none of the considered vegetative models could estimate the losses with an acceptable level of error. On the contrary, the RMSE values were above 10 dB in most of the experimental conditions.

Furthermore, in this study, we departed from the traditional approach of modeling losses from a physics perspective. On the contrary, we have embraced a more contemporary orientation based on machine learning (ML) by employing LR, K-NN SVM, and RF techniques within a training–testing method. From this approach, loss prediction is substantially improved. The use of RF, K-NN, and SVM allowed for a reduction in errors, achieving RMSE values below 3 dB and MAE values below 2 dB. Furthermore, the R^2 results obtained demonstrate that ML models can characterize losses with a high level of accuracy. However, it is important to note that not all ML techniques applied were successful, as LR did not yield satisfactory evaluation metrics ($RMSE = 8.81$, $MAE = 7.29$, and $R^2 = 0.4650$). Despite the favorable outcomes, it is crucial to consider additional factors when selecting an ML technique to predict path losses in cassava crops—for instance, the volume of data, computational requirements, and the performance of predictions across the entire range of experimental choices.

5. Conclusions

Though the results achieved through ML usage are satisfactory, it is important to emphasize the study's limitations, particularly the omission of additional factors that significantly influence radio signal attenuation. For instance, our experimentation was conducted in the 2400 MHz band and was not extended to other frequencies. Another factor to consider is that changes in environmental conditions (such as bushes of different varieties or changes in the arrangement of plants in the crop) can significantly affect the results, which is inherent in empirical propagation models.

Despite its advantages, ML does not define the variables that should be used in model generation; these are left to the researchers' discretion. Furthermore, there is no standardized methodology that outlines the conditions for data collection and model development. Therefore, a profound understanding of propagation matters, and the conditions under which measurements should be conducted remains essential, ensuring the proper correlation between the study's aspects and the acquired data.

To extend the results, it is recommended that future work extends the dataset. An alternative is to expand measurements throughout the entire process of bush growth. In addition, experiments should be carried out in other ISM frequency bands commonly used for WSN deployment (e.g., the 900 MHz band). It would be equally relevant to consider variables related to the shrub, such as stem thickness and leaf dimensions, as well as climatic factors, such as rainfall and humidity. The study could also be extended by analyzing data from other crops considered important for food sustainability, such as maize, rice, potatoes, or fruits. Future studies could also involve mixed crop or foliage scenarios in rural environments, such as gardens or parks.

For future work, they might consider using other ML techniques, such as lasso regression or neural networks, so that their results help to improve the analysis. Other regression techniques, such as quadratic or cubic regression, could also be further evaluated to see if they improve levels of fit and decrease error metrics. Furthermore, it is feasible to extend the study to other crops of importance for food sustainability, such as rice or potatoes. Furthermore, future research should address strengthening the security of the data collected during the measurement process to preserve their integrity and reduce vulnerability to potential cyberattacks.

Author Contributions: Conceptualization, A.B.-U. and R.R.-V.; Methodology, A.B.-U., A.C.-P., D.C.-P. and R.R.-V.; Validation, A.B.-U. and R.R.-V.; Formal Analysis, A.C.-P. and E.D.-I.-H.-F.; Investigation, A.B.-U.; Data Curation, A.B.-U. and R.R.-V.; Writing—original draft preparation D.C.-P. and A.C.-P.; Writing—review and editing, E.D.-I.-H.-F. and D.C.-P.; Visualization, A.C.-P., E.D.-I.-H.-F. and D.C.-P.; Supervision, E.D.-I.-H.-F. and D.C.-P.; Project Administration, A.C.-P. and D.C.-P.; resources, A.C.-P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Beckman, J.; Countryman, A.M. The Importance of Agriculture in the Economy: Impacts from COVID-19. *Am. J. Agric. Econ.* **2021**, *103*, 1595–1611. [CrossRef] [PubMed]
2. Arrubla-Hoyos, W.; Ojeda-Beltrán, A.; Solano-Barliza, A.; Rambauth-Ibarra, G.; Barrios-Ulloa, A.; Cama-Pinto, D.; Arrabal-Campos, F.M.; Martínez-Lao, J.A.; Cama-Pinto, A.; Manzano-Agugliaro, F. Precision Agriculture and Sensor Systems Applications in Colombia through 5G Networks. *Sensors* **2022**, *22*, 7295. [CrossRef] [PubMed]
3. International Society of Precision Agriculture Precision AG Definition. Available online: <https://www.ispag.org/about/definition> (accessed on 3 April 2023).
4. Zhang, C.; Lu, Y. Study on artificial intelligence: The state of the art and future prospects. *J. Ind. Inf. Integr.* **2021**, *23*, 100224. [CrossRef]
5. Barrios-Ulloa, A.; Ariza-Colpas, P.P.; Sánchez-Moreno, H.; Quintero-Linero, A.P.; De la Hoz-Franco, E. Modeling Radio Wave Propagation for Wireless Sensor Networks in Vegetated Environments: A Systematic Literature Review. *Sensors* **2022**, *22*, 5285. [CrossRef]
6. Sander-Frigau, M.; Zhang, T.; Lim, C.Y.; Zhang, H.; Kamal, A.E.; Somani, A.K.; Hey, S.; Schnable, P. A Measurement Study of TVWS Wireless Channels in Crop Farms. In Proceedings of the 2021 IEEE 18th International Conference on Mobile Ad Hoc and Smart Systems (MASS), Denver, CO, USA, 4–7 October 2021; pp. 344–354. [CrossRef]
7. Pal, P.; Sharma, R.P.; Tripathi, S.; Kumar, C.; Ramesh, D. 2.4 GHz RF Received Signal Strength Based Node Separation in WSN Monitoring Infrastructure for Millet and Rice Vegetation. *IEEE Sens. J.* **2021**, *21*, 18298–18306. [CrossRef]
8. Cama-Pinto, D.; Damas, M.; Holgado-Terriza, J.A.; Gómez-Mula, F.; Cama-Pinto, A. Path loss determination using linear and cubic regression inside a classic tomato greenhouse. *Int. J. Environ. Res. Public Health* **2019**, *16*, 1744. [CrossRef]
9. Ganev, Z. Log-normal shadowing model for outdoor propagation between sensor nodes. In Proceedings of the 2018 20th International Symposium on Electrical Apparatus and Technologies (SIELA), Bourgas, Bulgaria, 3–6 June 2018; pp. 9–12.
10. Miao, Y.; Zhao, C.; Wu, H. Non-uniform clustering routing protocol of wheat farmland based on effective energy consumption. *Int. J. Agric. Biol. Eng.* **2021**, *14*, 163–170. [CrossRef]
11. Navarro, A.; Guevara, D.; Florez, G.A. An Adjusted Propagation Model for Wireless Sensor Networks in Corn Fields. In Proceedings of the 020 XXXIIIrd General Assembly and Scientific Symposium of the International Union of Radio Science, Rome, Italy, 29 August–5 September 2020; pp. 1–3.
12. FAO. Save and Grow: Cassava a Guide to Sustainable Production Intensification. Available online: <https://www.fao.org/publications/card/en/c/3ef3b0e-492e-5ea8-816c-8d1e3b86d66b/> (accessed on 4 April 2023).
13. OECD-FAO Agricultural Outlook 2020–2029. Available online: https://www.oecd-ilibrary.org/agriculture-and-food/oecd-fao-agricultural-outlook-2020-2029_1112c23b-en (accessed on 4 April 2023).
14. Caicedo-Ortiz, J.G.; De-la-Hoz-Franco, E.; Morales Ortega, R.; Piñeres-Espitia, G.; Combata-Niño, H.; Estévez, F.; Cama-Pinto, A. Monitoring system for agronomic variables based in WSN technology on cassava crops. *Comput. Electron. Agric.* **2018**, *145*, 275–281. [CrossRef]
15. Manick; Srivastava, J. Cassava Leaf Disease Detection Using Deep Learning. In Proceedings of the 2022 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), Toronto, ON, Canada, 1–4 June 2022. [CrossRef]
16. Maryum, A.; Akram, M.U.; Salam, A.A. Cassava Leaf Disease Classification using Deep Neural Networks. In Proceedings of the 2021 IEEE 18th International Conference on Smart Communities: Improving Quality of Life Using ICT, IoT and AI (HONET), Karachi, Pakistan, 11–13 October 2021; pp. 32–37. [CrossRef]
17. Chen, C.C.; Ba, J.Y.; Li, T.J.; Chan, C.C.K.; Wang, K.C.; Liu, Z. EfficientNet: A low-bandwidth IoT image sensor framework for cassava leaf disease classification. *Sens. Mater.* **2021**, *33*, 4031–4044. [CrossRef]
18. Gao, Z.; Li, W.; Zhu, Y.; Tian, Y.; Pang, F.; Cao, W.; Ni, J. Wireless channel propagation characteristics and modeling research in rice field sensor networks. *Sensors* **2018**, *18*, 3116. [CrossRef]
19. Pal, P.; Sharma, R.P.; Tripathi, S.; Kumar, C.; Ramesh, D. Machine Learning Regression for RF Path Loss Estimation Over Grass Vegetation in IoWSN Monitoring Infrastructure. *IEEE Trans. Ind. Inform.* **2022**, *18*, 6981–6990. [CrossRef]

20. Phokharatkul, P.; Phaiboon, S. Path Loss Model for the Bananas and Weeds Environment Based on Grey System Theory. In Proceedings of the 2021 Photonics & Electromagnetics Research Symposium (PIERS), Hangzhou, China, 21–25 November 2021; pp. 413–418. [CrossRef]
21. Anzum, R.; Hadi Habaebi, M.; Islam, R.; Hakim, G.P.N. Modeling and Quantifying Palm Trees Foliage Loss using LoRa Radio Links for Smart Agriculture Applications. In Proceedings of the 2021 IEEE 7th International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA), Bandung, Indonesia, 23–25 August 2021; pp. 105–110. [CrossRef]
22. Juan-Llacer, L.; Molina-Garcia-Pardo, J.M.; Sibille, A.; Torrico, S.A.; Rubiola, L.M.; Martinez-Ingles, M.T.; Rodriguez, J.V.; Pascual-Garcia, J. Path Loss Measurements and Modelling in a Citrus Plantation in the 1800 MHz, 3.5 GHz and 28 GHz in LoS. In Proceedings of the 2022 16th European Conference on Antennas and Propagation (EuCAP), Madrid, Spain, 27 March–1 April 2022. [CrossRef]
23. Wu, H.; Zhu, H.; Han, X.; Xu, W. Layout optimization for greenhouse WSN based on path loss analysis. *Comput. Syst. Sci. Eng.* **2021**, *37*, 89–104. [CrossRef]
24. Kale, A.; Nguyen, T.; Harris, F.C.; Li, C.; Zhang, J.; Ma, X. Provenance documentation to enable explainable and trustworthy AI: A literature review. *Data Intell.* **2023**, *5*, 139–162. [CrossRef]
25. Oroza, C.A.; Zhang, Z.; Watteyne, T.; Glaser, S.D. A Machine-Learning-Based Connectivity Model for Complex Terrain Large-Scale Low-Power Wireless Deployments. *IEEE Trans. Cogn. Commun. Netw.* **2017**, *3*, 576–584. [CrossRef]
26. Kochhar, A.; Kumar, N.; Arora, U. Signal Assessment Using ML for Evaluation of WSN Framework in greenhouse monitoring. *Int. J. Sensors Wirel. Commun. Control* **2022**, *12*, 669–679. [CrossRef]
27. ITU-R. *ITU-R Recommendation P.833-7 Attenuation in Vegetation*; ITU-R: Geneva, Switzerland, 2012; Volume 7.
28. Olasupo, T.O.; Otero, C.E. The Impacts of Node Orientation on Radio Propagation Models for Airborne-Deployed Sensor Networks in Large-Scale Tree Vegetation Terrains. *IEEE Trans. Syst. Man Cybern. Syst.* **2020**, *50*, 256–269. [CrossRef]
29. Sabri, N.; Mohammed, S.S.; Fouad, S.; Syed, A.A.; Al-Dhief, F.T.; Raheemah, A. Investigation of Empirical Wave Propagation Models in Precision Agriculture. *MATEC Web Conf.* **2018**, *150*, 06020. [CrossRef]
30. Raheemah, A.; Sabri, N.; Salim, M.S.; Ekan, P.; Ahmad, R.B. New empirical path loss model for wireless sensor networks in mango greenhouses. *Comput. Electron. Agric.* **2016**, *127*, 553–560. [CrossRef]
31. Anzum, R.; Habaebi, M.H.; Islam, M.R.; Hakim, G.P.N.; Khandaker, M.U.; Osman, H.; Alamri, S.; AbdElrahim, E. A Multiwall Path-Loss Prediction Model Using 433 MHz LoRa-WAN Frequency to Characterize Foliage's Influence in a Malaysian Palm Oil Plantation Environment. *Sensors* **2022**, *22*, 5397. [CrossRef]
32. Dogan, H. A new empirical propagation model depending on volumetric density in citrus orchards for wireless sensor network applications at sub-6 GHz frequency region. *Int. J. RF Microw. Comput. Eng.* **2021**, *31*, e22778. [CrossRef]
33. Shaik, M.; Kabanni, A.; Nazeema, N. Millimeter wave propagation measurements in forest for 5G Wireless sensor communications. In Proceedings of the 2016 16th Mediterranean Microwave Symposium (MMS), Abu Dhabi, United Arab Emirates, 14–16 November 2016; pp. 1–4. [CrossRef]
34. Olasupo, T.O.; Alsayyari, A.; Otero, C.E.; Olasupo, K.O.; Kostanic, I. Empirical path loss models for low power wireless sensor nodes deployed on the ground in different terrains. In Proceedings of the 2017 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT), Aqaba, Jordan, 11–13 October 2017; pp. 1–8.
35. Burkov, A. *The Hundred-Page Machine Learning*; Burkov, A., Ed.; Andriy Burkov: Quebec City, QC, Canada, 2019; ISBN 978-1-9995795-1-7.
36. Géron, A. *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 2nd ed.; Tache, N., Ed.; O'Reilly Media: Sebastopol, CA, USA, 2019; ISBN 1098125975.
37. Zhang, J.; Liu, L.; Fan, Y.; Zhuang, L.; Zhou, T.; Piao, Z. Wireless Channel Propagation Scenarios Identification: A Perspective of Machine Learning. *IEEE Access* **2020**, *8*, 47797–47806. [CrossRef]
38. Zhang, Y.; Wen, J.; Yang, G.; He, Z.; Wang, J. Path loss prediction based on machine learning: Principle, method, and data expansion. *Appl. Sci.* **2019**, *9*, 1908. [CrossRef]
39. Theobald, O. *Machine Learning for Absolute Beginners*; Independently published, 2017; ISBN 978-1520951409.
40. Moraitis, N.; Tsipli, L.; Vouyioukas, D.; Gkioni, A.; Louvros, S. Performance evaluation of machine learning methods for path loss prediction in rural environment at 3.7 GHz. *Wirel. Netw.* **2021**, *27*, 4169–4188. [CrossRef]
41. Vergos, G.; Sotiroudis, S.P.; Athanasiadou, G.; Tsoulos, G.V.; Goudos, S.K. Comparing Machine Learning Methods for Air-to-Ground Path Loss Prediction. In Proceedings of the 2021 10th International Conference on Modern Circuits and Systems Technologies, MOCAST, Thessaloniki, Greece, 5–7 July 2021; pp. 1–4.
42. Elmezughi, M.K.; Salih, O.; Afullo, T.J.; Duffy, K.J. Comparative Analysis of Major Machine-Learning-Based Path Loss Models for Enclosed Indoor Channels. *Sensors* **2022**, *22*, 4967. [CrossRef] [PubMed]
43. Breiman, L. Random Forest. *Mach Learn* **2001**, *45*, 5–32. [CrossRef]
44. Probst, P.; Wright, M.N.; Boulesteix, A.L. Hyperparameters and tuning strategies for random forest. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2019**, *9*, e1301. [CrossRef]
45. Antoniadis, A.; Lambert-Lacroix, S.; Poggi, J.-M. Random forests for global sensitivity analysis: A selective review. *Reliab. Eng. Syst. Safe* **2021**, *206*, 107312. [CrossRef]
46. Cama-Pinto, D.; Damas, M.; Holgado-Terriza, J.A.; Arrabal-Campos, F.M.; Gómez-Mula, F.; Lao, J.A.M.; Cama-Pinto, A. Empirical model of radio wave propagation in the presence of vegetation inside greenhouses using regularized regressions. *Sensors* **2020**, *20*, 6621. [CrossRef]

47. Vougioukas, S.; Anastassiou, H.T.; Regen, C.; Zude, M. Influence of foliage on radio path losses (PLs) for Wireless Sensor Network (WSN) planning in orchards. *Biosyst. Eng.* **2013**, *114*, 454–465. [CrossRef]
48. Nagao, T.; Hayashi, T. Fine-Tuning for Propagation Modeling of Different Frequencies with Few Data. In Proceedings of the 2022 IEEE 96th Vehicular Technology Conference (VTC2022-Fall), London, UK, 26–29 September 2022; pp. 1–5. [CrossRef]
49. Goudos, S.K.; Athanasiadou, G.; Tsoulos, G.V.; Rekkas, V. Modelling Ray Tracing Propagation Data Using Different Machine Learning Algorithms. In Proceedings of the 2020 14th European Conference on Antennas and Propagation (EuCAP), Copenhagen, Denmark, 15–20 March 2020.
50. Jo, H.S.; Park, C.; Lee, E.; Choi, H.K.; Park, J. Path loss prediction based on machine learning techniques: Principal component analysis, artificial neural network and gaussian process. *Sensors* **2020**, *20*, 1927. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.



Article

Detection of Bagworm Infestation Area in Oil Palm Plantation Based on UAV Remote Sensing Using Machine Learning Approach

Siti Nurul Afiah Mohd Johari ¹, Siti Khairunniza-Bejo ^{1,2,3,*}, Abdul Rashid Mohamed Shariff ^{1,2,3}, Nur Azuan Husin ^{1,2}, Mohamed Mazmira Mohd Masri ⁴ and Noorhazwani Kamarudin ⁴

¹ Department of Biological and Agricultural Engineering, Faculty of Engineering, Universiti Putra Malaysia (UPM), Serdang 43400, Selangor, Malaysia; gs55369@student.upm.edu.my (S.N.A.M.J.); rashidpls@upm.edu.my (A.R.M.S.); nurazuan@upm.edu.my (N.A.H.)

² Smart Farming Technology Research Centre, Universiti Putra Malaysia (UPM), Serdang 43400, Selangor, Malaysia

³ Institute of Plantation Studies, Universiti Putra Malaysia (UPM), Serdang 43400, Selangor, Malaysia

⁴ Malaysian Palm Oil Board (MPOB), No. 6, Persiaran Institusi, Bandar Baru Bangi, Kajang 43000, Selangor, Malaysia; mazmira@mpob.gov.my (M.M.M.M.); hazwani.km@mpob.gov.my (N.K.)

* Correspondence: skbejo@upm.edu.my; Tel.: +60-397694332

Abstract: Due to its rapid reproduction rate and brief life cycle, the most well-known oil palm pest, *Metisa plana* (Lepidoptera: Psychidae), also known as the bagworm, can spread to epidemic proportions. The outbreak can significantly reduce oil palm yield by resulting in 40% crop losses and 10% to 13% leaf defoliation. A manual census was conducted to count the number of pests and determine the category of infestation; however, when covering a large area, it typically takes more time and labour. Therefore, this study used unmanned aerial vehicles (UAVs) as a quick way to detect the severity levels of infestation in oil palm plantations, including healthy (zero), low, mild, and severe infestation using DJI Inspire 2 with Micasense Altum-PT multispectral camera at an altitude of 70 m above ground. Three combinations were created from the most significant vegetation indices: NDVI and NDRE, NDVI and GNDVI, and NDRE and GNDVI. According to the results, the best combination in classifying healthy and low levels was found to be NDVI and GNDVI, with 100% F1 score. In addition, the combination of NDVI and NDRE was found to be the best combination in classifying mild and severe level. The most important vegetation index that could detect every level of infestation was NDVI. Furthermore, Weighted KNN become the best model that constantly gave the best performance in classifying all the infestation levels (F1 score > 99.70%) in all combinations. The suggested technique is crucial for the early phase of severity-level detection and saves time on the preparation and operation of the control measure.

Keywords: multispectral image; bagworm; infestation; vegetation index; unmanned aerial vehicle; machine learning

Citation: Johari, S.N.A.M.; Khairunniza-Bejo, S.; Shariff, A.R.M.; Husin, N.A.; Masri, M.M.M.; Kamarudin, N. Detection of Bagworm Infestation Area in Oil Palm Plantation Based on UAV Remote Sensing Using Machine Learning Approach. *Agriculture* **2023**, *13*, 1886. <https://doi.org/10.3390/agriculture13101886>

Academic Editors: Gniewko Niedbala, Sebastian Kujawa, Magdalena Piekutowska and Tomasz Wojciechowski

Received: 6 September 2023

Revised: 19 September 2023

Accepted: 21 September 2023

Published: 27 September 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Bagworms (Lepidoptera: Psychidae) are tiny insect pest larvae that are prevalent worldwide in arborvitae as well as other fruit and flower crops like apple, maple, elm, poplar, oak, birch, black locust, cypress, juniper, willow, and juniper. In Malaysia, bagworm, especially *Metisa plana*, is the most serious insect pest which is capable of reaching epidemic proportions in oil palm plantations by presenting higher numbers than usual [1]. Bagworms are “naturally created” to easily become pests due to their high reproductive rate and short life cycle, which are gifts of natural advantages, along with their unique dispersal mode, case construction, and silk thread as survival mechanisms [2]. Bagworm outbreaks are frequent in oil palm plantations and can result in up to 40% crop losses and 10% to 13% leaf defoliation, both of which have a significant negative economic impact on oil palm yield [1,3]. According to Chung [4], small holes from feeding are the first signs of bagworm

damage on fronds. The outbreaks are noticeable because the bagworms start eating as soon as they hatch, scraping the top surface of the leaf until it dries out and leaving holes. The palms with severe bagworm infestations suffer increased amounts of foliage damage until all the fronds are lost, typically in the upper part of the palm fronds, which appear brown in colour. Brownish-coloured frond damage results from severe bagworm infestation. Furthermore, the severely damaged leaves caused the lower and central crown to appear greyish brown [5]. When pest populations reached their maximum growth potential, they frequently reached levels that significantly reduced leaf cover over large areas and had a propensity to recur [6]. Single-species outbreaks were frequently reported, while mixed species outbreaks could affect both young and old palms, though the area of the outbreak is typically larger on older palms [7]. According to Aziz et al. [8], precise estimation of the infestation based on the oil palm foliar damage is difficult to identify. Thus, the estimation of the damage severity rating was used which corresponds to the bagworm infestation. The severity rating was divided into four levels, which started with zero infestation, followed by light damage, medium damage, and serious damage. All the severity ratings are summarized in Table 1.

Table 1. Classification of damage due to bagworm infestation [8].

Infestation	Classification	Description
0	NIL	<ul style="list-style-type: none"> There is no obvious bagworm harm.
1	Light	<ul style="list-style-type: none"> A leaflet with very few bagworm larvae and pupae. Leaflets start to have small holes and necrosis.
2	Medium	<ul style="list-style-type: none"> Most leaflets contain pupae and larvae of bagworms. Leaflets with several holes and light necrosis.
3	Serious	<ul style="list-style-type: none"> Numerous bagworm larvae and pupae on the leaflet. Lots of necrosis and numerous holes on the leaflets and drying out and turning brown.

A census must be conducted to effectively control bagworm in an oil palm plantation. It is carried out to count the number of insect pests directly, which involves a superficial inspection for signs of pest incidence [5]. It is conducted by cutting down one frond to count the number of larvae on both sides of the frond. According to the Standard Operating Procedure (SOP) of bagworm control by the Malaysian Palm Oil Board (2016) [9], a census is conducted on 1% of the infested area, subject to the entire infested area, where one palm of every ten is sampled. Critical early defoliation can be considered present when there are ten larvae on each frond [10]. When it comes to covering a large area, this method typically requires more time and labour. Therefore, a quick and trustworthy method utilising remote sensing would be helpful in determining the degree of bagworm-infested area for prompt decisions regarding outbreak control measures.

Technology advancements that replace manual sampling techniques have benefited the agriculture sector, particularly in terms of increasing crop production. Unmanned aerial vehicles (UAV), also known as drones, are an increasingly important component of remote sensing tools in the context of precision farming. UAVs have limitless potential in agriculture, and they have the power to revolutionise the industry along with smart farming and new data management techniques [11]. Usually, UAV platforms equipped with a wide range of sensor types, such as visual RGB (Red, green, blue) cameras, multispectral cameras, hyperspectral cameras, and thermal cameras that can capture images with flexible revisit scheduling at low altitudes with ultra-spatial and temporal resolutions have allowed for the observation of small individual plants and the extraction of information at a fine scale

that can aid farmers in making decisions, improve agricultural production, and maximise resource utilisation [12–14].

Moreover, UAV images and machine learning (ML) techniques have developed new ways to examine datasets recently, particularly in precision agriculture. These models can be powerful and useful tools for the prediction of various crop parameters using data obtained from UAV images. Vegetation Indices (VI) are algebraic combinations of different spectral bands that are used to highlight the vigour and other characteristics of vegetation (i.e., canopy biomass, absorbed radiation, chlorophyll content, etc.) [15]. Many VIs can be obtained from RGB cameras or multispectral cameras that consist of five channels (i.e., red, green, blue, near infrared, and red edge), such as the normalized difference vegetation index (NDVI), the green normalised difference vegetation index (GNDVI), the normalized difference red edge (NDRE), the simple ratio (SR), and the chlorophyll index (CI). These VIs were usually used to provide significant information in analysing vegetation traits such as plant diseases, pests, and stress detection. For instance, Klouček et al. [16] calculated selected vegetation indices (i.e., greenness index (GI), simple ratio (SR), green ratio vegetation index (GRVI), normalized difference vegetation index (NDVI), and green normalized difference vegetation index (GNDVI)) and evaluated them based on visual differences in the spectral curves of bark-beetle-infested tree and healthy trees. Minařík et al. [17] extracted elevation features (crown area, height percentiles) and three vegetation indices (i.e., NDVI, NDRE, and Enhanced Normalized Difference Vegetation Index (ENDVI)) to detect a bark beetle disturbance in a mixed urban forest. According to Tsouros et al. [18], the most popular techniques for analysing UAV imagery for precision agriculture include vegetation indices and machine learning. These techniques were used to detect pests and diseases in a variety of crops, including coffee [19], wheat [20], citrus [21], cotton [22] and forests [16,23], as summarized in Table 2. Despite the number of various remote sensing approaches that were used to monitor pests and diseases in oil palm plantations [24,25], their application at the UAV platform together with vegetation indices and machine learning techniques is still limited [26,27]. Based on Table 2, it also can be concluded that the same vegetation indices can be used to detect different types of pests and diseases in different types of crops. Therefore, it also has the potential to be used for detecting bagworm infestation areas in oil palm plantations.

Table 2. Summary of the application of UAV-based imagery with machine learning technique.

Crop Type	Purpose of Study	Sensor Type	Vegetation Indices	Machine Learning	Classification Performance	References
Coffee	Coffee leaf rust disease	Multispectral camera	Normalized difference vegetation index (NDVI) Green Normalized Difference Vegetation Index (GNDVI) Normalized Difference Red Edge (NDRE) Modified normalized vegetation red edge (MNDRE) Modified Green Simple Ratio (MGSR)	Logistic model tree (LMT)	F1 score: 91.50% (early stage) F1 score: 87.50% (late stage)	[19]
Wheat	Yellow rust disease	Multispectral camera	Ratio vegetation index (RVI) Normalized difference vegetation index (NDVI) Optimized soil adjusted vegetation index (OSAVI)	Random forest (RF)	Precision: 89.20% Recall: 89.40% Accuracy: 89.3% (45 days after inoculation)	[20]

Table 2. Cont.

Crop Type	Purpose of Study	Sensor Type	Vegetation Indices	Machine Learning	Classification Performance	References
Forest	Bark beetle infestation	RGB camera NIR customized sensor	Greenness Index (GI), Simple Ratio (SR), Green Ratio Vegetation Index (GRVI), Normalized Difference Vegetation Index (NDVI), Green Normalized Difference Vegetation Index (GNDVI)	Maximum likelihood classifier (MLC)	Overall accuracy: 78–96% (across the time periods)	[16]
	Pine wilt disease	Hyperspectral camera (HI) and LiDAR	Normalized Difference Vegetation Index (NDVI), Simple Ratio (SR), Chlorophyll Index (CI),	Random forest (RF)	Overall accuracy (HI + LiDAR): 73.96% (Early-stage PWD)	[23]
Citrus	Greening disease	Multispectral camera	Normalized difference vegetation index (NDVI) Simple ratio (SR) Chlorophyll index (CI) Green Normalized Difference Vegetation Index (GNDVI) Normalized Difference Red Edge (NDRE)	Support vector machine (SVM)	Overall accuracy: 81.50%	[21]
Cotton	Cotton rot disease	Multispectral camera	Green, red and NIR band (CIR)	Unsupervised: k-means Supervised: support vector machine (SVM), minimum distance, maximum likelihood, Mahalanobis distance	Overall accuracy: 88.5%	[22]
Oil palm	Bud rot (BR) and red ring disease (RRD)	Multispectral camera	Normalized difference vegetation index (NDVI), Green Normalized Difference Vegetation Index (GNDVI), Green vegetation index (GVI) Visible atmospherically resistant index (VARI)	Lowest Significant Difference (LSD)	Statistically significant differences between healthy and diseased palms, (generating the baseline of early responses of BR and RRD)	[26]
	Bagworm infestation	Multispectral camera	Three band combinations; Green_Red_RedEdge Green_Red_NIR Red_RedEdge_NIR Green_RedEdge_NIR	- (Visual analysis)	Green_Red_RedEdge is the best to visually differentiate healthy and infested oil palm	[27]
	Ganoderma disease	Multispectral camera	Three band combinations; Green_Red_RedEdge Green_Red_NIR Red_RedEdge_NIR Green_RedEdge_NIR	Object-based- image-analysis method (OBIA)	>80%	[27]

Previously, several bagworm studies were conducted in ground-based and aerial-based detection. Ground-based detection normally was carried out to detect the presence of bagworm, as performed by Ahmad et al. [28], who identified both live and bag-

worms *Metisa plana* using a motion tracking technique on oil palm fronds. Nevertheless, this method was only applied to live and dead bagworms without knowing the specific instar stage. Since classifying bagworm instar stages is essential for early prevention, Mohd Johari et al. [29] used machine learning to identify bagworm instar stages based on spectral properties and upgraded to automatic detection using the transfer learning approach [30]. However, these studies did not address the infestation. Previously, the detection of foliar damaged was carried out by Aziz et al. [8], who discovered that the most sensitive wavelengths (i.e., 570 nm, 680 nm, 734 nm, 787 nm, 996 nm, and 1047 nm) to detect bagworm-infested foliar damage using ground-based spectrometer. However, these studies were not appropriate to be applied in a large plantation area, as they require high labour costs and are time consuming; thus, an aerial-based approach was proposed to recognise the issue. Anuar et al. [27] applied a multispectral camera mounted on an UAV to detect the bagworm-infested area and compare it with healthy area and concluded that a multispectral false-colour composite has the capability to differentiate between healthy and bagworm-infested oil palm. Nonetheless, this study only focuses on two areas (i.e., healthy, and infested) and does not distinguish between different infested areas, such as low-infestation areas, mildly infested areas, and severely infested areas.

Based on the literature listed above, the study that assessed the potential of UAV images in detecting different severity levels of bagworm infestation in oil palm plantation was limited, and more needed to be discovered. Therefore, this study uses UAV-acquired images and machine learning techniques to locate the bagworm *Metisa plana* infestation area. This study focuses exclusively on the ability of machine learning to categorise the severity level of infestation as healthy, low infestation, mild infestation, and severe infestation using vegetation indices extracted from UAV images.

2. Materials and Methods

2.1. Overview

Figure 1 shows a flowchart of this study. It began with the data collection, including study site selection for each category of bagworm infestation, i.e., healthy, low, mild, and severe infestation, and also the ground assessment of the level of infestation. Image acquisition was conducted using an UAV, and all the captured images were then processed for mosaicking and exported in TIFF format. Six selected vegetation indices were derived and extracted from the imagery. Statistical analysis was carried out to identify the significant differences between the indices. The three most significant indices were selected for the classification model. Then, the performance of the model was evaluated based on the value of accuracy and F1 score. Due to the imbalanced dataset, undersampling and oversampling methods were used to achieve the balance distribution of the dataset for the further analysis. A classification model was developed using the combination of significant vegetation indices. The model was then tested with the original dataset, and the performance of the model was evaluated. The most insensitive model with the highest F1 score was selected as the best model in this study.

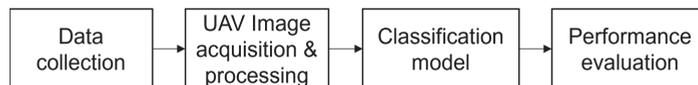


Figure 1. Flowchart of the study.

2.2. Data Collection

This study was conducted in three different plantation areas, which covers four categories of infestation, as described in Table 3. These study areas are presented in Figure 2 using a Google Earth imagery (2020) and labelled in the red frames. A healthy plantation area that showed no sign of infestation was located at Serdang, Selangor, covering about 6 hectares at coordinate location (2°59'13" N, 101°43'34" E) (Figure 2a). Low and mild infestations were located at Pagoh, Johor (Figure 2b,c), covering 7.0 and 2.0 hectares,

respectively. Meanwhile, a severe infestation area was located at Ayer Kuning, Perak, with coordinate location (4°11'56" N, 101°07'38" E), covering 10 hectares (Figure 2d).

Table 3. Information about selected oil palm plantations.

Infestation	Location	Area (ha)	Coordinates	Number of Trees
Healthy	Serdang, Selangor	6.0	2°59'13" N 101°43'34" E	750
Low	Pagoh, Johor	7.0	2°10'39" N 102°47'01" E	800
Mild		2.0	2°10'48" N 102°47'11" E	300
Severe	Ayer Kuning, Perak	10.0	4°11'56" N 101°07'38" E	1000

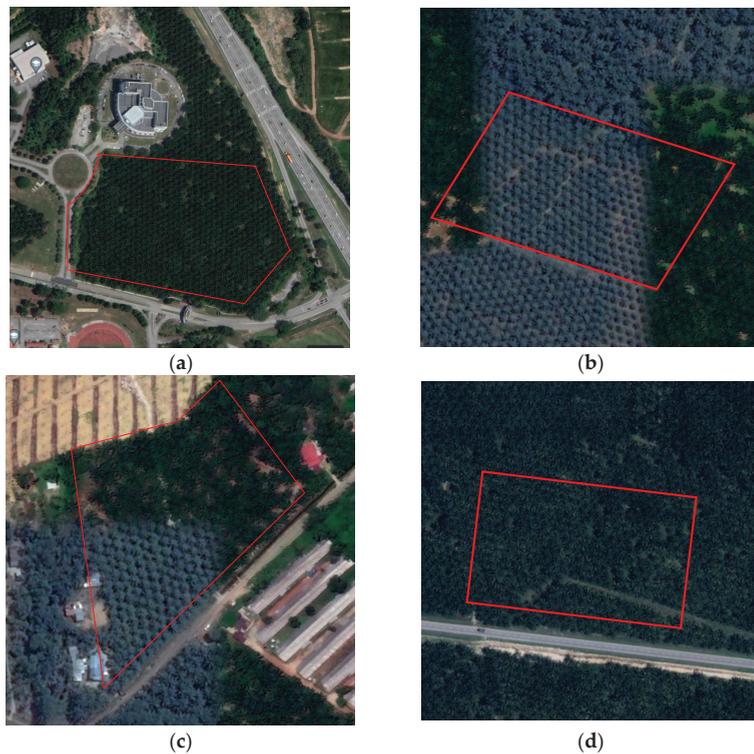


Figure 2. The study sites, (a) Serdang, Selangor, (b) Pagoh, Johor, (c) Pagoh, Johor, and (d) Ayer Kuning, Perak.

The incidence of oil palm trees was determined to assess the bagworm infestation in the oil palm plantation using a quantitative assessment, which was carried out based on bagworm-infestation symptoms. The assessment was carried out based on the number of infested fronds over the total number of fronds of each oil palm tree, as suggested by Thaer et al. [31], using the following formula in Equation (1).

$$incidence\ rate = \frac{number\ of\ infested\ frond}{number\ of\ total\ frond} \tag{1}$$

Usually, each palm consists of 30–40 frond leaves. The infected fronds were identified based on the condition of the frond and the number of bagworms detected per frond. The incidence rate was ranked according to infestation and severity of the bagworm infestation, subject to the severity scale mentioned previously. In addition, a manual census was also carried out to identify the bagworm instar stage and to sum up the existence of the bagworms in each frond. The process involved cutting the frond randomly and observing the bagworms. The incidence rate and the number of bagworms per frond are summarized in Table 4.

Table 4. Incidence rate and details of bagworm infestation.

Classification	Percentage of Incident Rate	Details of Incident Rate	Number of Bagworms per Frond
NIL (Healthy)	0%	No infested frond detected	No bagworm detected
Low	1–33%	Light necrosis was detected	10 and below
Mild	34–67%	Moderate necrosis and hole detected	11–50
Severe	68–100%	Serious necrosis and hole detected	50 and above

2.3. UAV Image Acquisition and Processing

The images were taken with fixed exposure settings between 10:00 and 11:00 am local time on a clear and non-cloudy day using a DJI Inspire 2 UAV (DJI Sky City, Shenzhen, China) with a rotary wing, known as a quadcopter, equipped with a Micasense Altum-PT multispectral camera (Seattle, Washington, DC, USA) (Figure 3).

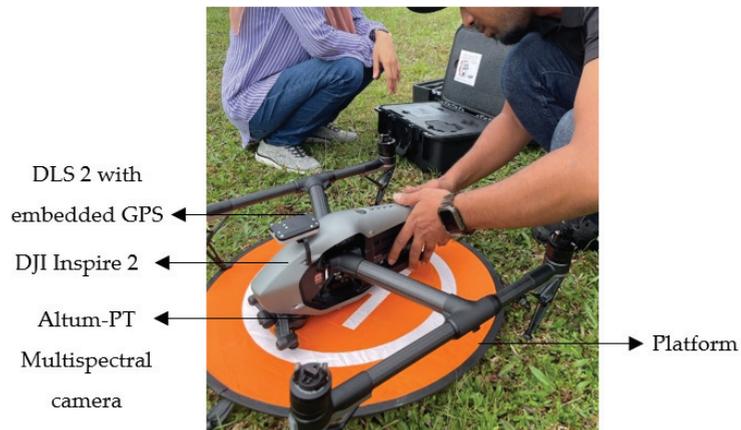


Figure 3. DJI Inspire 2 with Altum-PT Multispectral camera.

The Micasense series, which has five bands and can record data in the RGB, near-infrared, and red-edge regions (400–900 nm), was the pioneer of multispectral cameras [32]. It can be used in a wide range of UAV types because of its lightweight design (577 g) and small size (11 cm × 8 cm × 6.9 cm). It has five sensors with resolution of 3.2 megapixels (2064 × 1544 pixels) in the five spectral regions of blue (475–500 nm), green (550–565 nm), red (665–675 nm), red edge (715–725 nm), and NIR (825–860 nm). The sensor acquires all five bands at a ground sample distance (GSD) of 120 m at a speed of up to 2 captures per second with a 50° horizontal field of view (HFOV) and a 38° vertical field of view (VFOV). It also captures ultra-high-resolution panchromatic images and has a thermal sensor at the resolution of 12.4 megapixels (4112 × 3008 pixels) and 0.1 megapixels (320 × 256 pixels),

respectively, for data output. A downwelling light sensor (DLS), mounted upward on the UAV, measures incident light and enables radiometric calibration of these 5 multispectral bands during image capture.

Meanwhile, the DJI Inspire 2 is a powerful, high-tech drone that weighs approximately 3.44 kg and is capable of transmitting video in both 1080p and 720p at a maximum distance of 7 km. It travels at an impressive 94 km/h, which is quite fast. The UAV measures 42.7 cm in length, 31.7 cm in height, and 42.5 cm in width. There are 150 to 390 RAW images for each flight mission. The GPS coordinates on each photo help with 3D reconstruction.

In this study, the flight altitude was set at 70 metres above ground. Orthomosaics with a 5.28 cm spatial resolution were taken at a speed of up to two captures per second and with 80% longitudinal and 75% lateral overlap. The Pix4Dmapper software version 4.13.1 (1) (Pix4D SA, Lausanne, Switzerland) was used to execute the flight missions autonomously. All the images taken were stored in an SD card.

Image mosaicking was performed in Agisoft Metashape Professional software (Agisoft LLC., St. Petersburg, Russia), which generates a multispectral orthomosaic which includes each band imagery (Blue, Green, Red, Red-edge, NIR and thermal). Agisoft was the most widely used software due to its advantages of excluding low-quality images and its standardised workflow [33]. The process of mosaicking was started by importing all the images into the software. The primary channel was set to panchromatic for a higher-resolution panchromatic band during alignment. The MicaSense Calibrated Reflectance Panel, which was captured prior to the flight, was then used to radiometrically calibrate all the images. The primary goal is to adjust the various radiometric resolutions between the UAV camera and the sensing periods. Then, the images were aligned, and a dense point cloud model of the objects was built from the numerous collected images while also fine-tuning the camera positions of each image. After the orthomosaic imagery was generated, all the imagery of each infestation level was then exported for further analysis.

All the images of four infestation levels were loaded in the QGIS, an open-source GIS software version 3.28.2 for data extraction. Reflectance values generated by the multispectral bands corresponding to blue (475–500 nm), green (550–565 nm), red (665–675 nm), red edge (715–725 nm), and NIR (825–860 nm) were used to calculate the vegetation indices. Six vegetation indices were derived, namely, the normalized difference vegetation index (NDVI), the green normalized difference vegetation index (GNDVI), the normalized difference red edge (NDRE), the simple ratio (SR), the green Chlorophyll Index (GCI), and the red edge Chlorophyll Index (RECI) (Table 5).

Table 5. List of vegetation indices (VIs) with formulas.

No.	Vegetation Index	Formula	Reference
1	NDVI	$\frac{NIR - Red}{NIR + Red}$	[34]
2	GNDVI	$\frac{NIR - Green}{NIR + Green}$	[35]
3	NDRE	$\frac{NIR - Rededge}{NIR + Rededge}$	[36]
4	SR	$\frac{NIR}{Red}$	[37]
5	GCI	$\frac{NIR}{Green} - 1$	[38]
6	RECI	$\frac{NIR}{Rededge} - 1$	

Five points were randomly selected from each canopy of palm tree, as illustrated in Figure 4. These sampling techniques were implemented for each sample of a tree in each category of infestation. Vegetation indices of these points were then averaged to represent the vegetation indices of each tree.

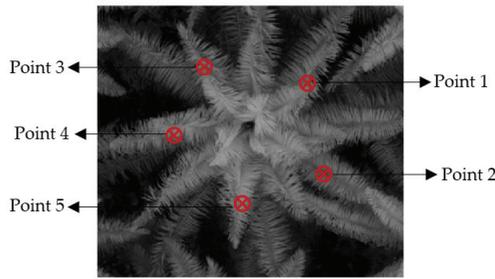


Figure 4. Illustration of vegetation index extraction.

2.4. Classification Model

A significant level of vegetation indices was identified using Analysis of Variance statistical analysis (ANOVA) in SPSS software (IBM SPSS Statistics 25, IBM, New York, NY, USA) based on a value of $p < 0.05$. Seventy percent of the total for each infestation level, totalling 11,970 datasets, served as the ANOVA's input parameters and was later used for model development, while the other 30% (5130) was used for testing. Only three significant vegetation indices with a standard error lower than 0.002 were selected as datasets to develop classification models using the classification learner apps available in the machine learning toolbox from MATLAB (2021b, The Mathworks Inc., Natick, MA, USA).

A K-fold cross-validation function in MATLAB was used to conduct a cross-validation process to assess the performance of the model. It was one of the most popular methods for classifier model selection and error estimation [39]. It divides each sample into a predetermined number of groups (N), of which N-1 groups are used to fit a model while the remaining sample is used for validation. Each group served as the validation group during the 'N' times this fitting and validation process was carried out. The model performances were described using the averaged values of the evaluating metrics. In this study, the N is set to 5, as it was randomly partitioned into 5 sub datasets of equivalent size. Figure 5 illustrates the 5-fold cross validation process.

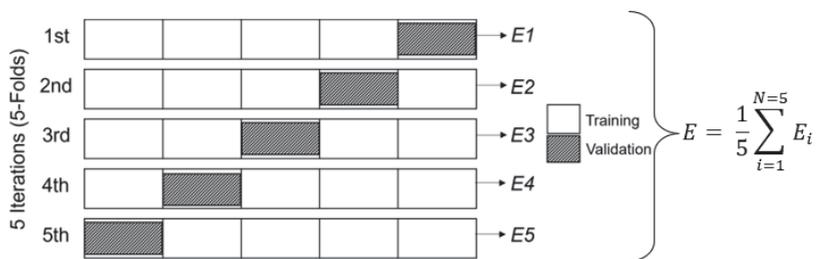


Figure 5. The illustration of 5-fold cross validation.

Figure 6 provides a summary of the 5 machine learning classifiers used in this study from the default setting by the classification learner apps in MATLAB, including Decision Tree (DT), Discriminant Analysis (DA), Naïve Bayes (NB), Support Vector Machine (SVM), and K-nearest Neighbour (KNN). The classification models were developed separately using three different combinations of vegetation indices: (a) NDVI and NDRE, (b) NDVI and GNDVI, and (c) NDRE and GNDVI. The best classification model was determined based on the highest F1 score mean value. The use of only 2 combinations of vegetation indices serves to create a straightforward and more cost-effective tool for future hardware design.

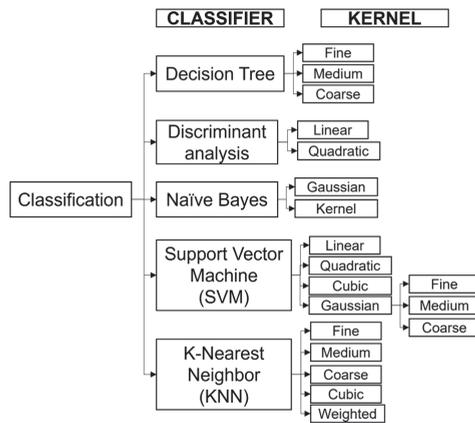


Figure 6. Type of classifier and kernel used.

2.5. Performance Evaluation

The classification for each infestation category was displayed using the multiclass confusion matrix, which shows the accuracy for each class while exposing specific misclassifications. From the confusion matrix, true positive (TP), false positive (FP), true negative (TN) and false negative (FN) can be calculated to assess the performance of the model such as accuracy, precision, recall, specificity, and F1 score. Accuracy is the proportion of correctly classified dataset over all datasets. The proportion of correctly predicted positive observations among all predicted positive observations is known as precision (Equation (2)). The proportion of correctly predicted positive observations to all the actual observations in a class is known as recall (Equation (3)). The F1 score (Equation (4)) is the harmonic mean of the precision and recall, which provides a measurement for the number of errors made by the algorithm, with 0 being the worst possible value and 1 being the best possible value. A high F1 score denotes both a high precision and recall. By comparing all the performance metrics, the F1 score seems more reliable when it comes to unbalanced data. A macro average was used to determine the results, which involved calculating each performance separately and averaging them. Additionally, the percentage difference (Equation (5)) of the F1 score between training (*i*) and testing (*j*) was then calculated to identify the pattern of the model either overfitting or underfitting.

$$Precision, P = \frac{TP}{(TP + FP)} \tag{2}$$

$$Recall, R = \frac{TP}{(TP + FN)} \tag{3}$$

$$F1 - score = 2 \times \frac{(P \times R)}{(P + R)} \tag{4}$$

$$Percentage\ difference = \frac{|i - j|}{i} \times 100 \tag{5}$$

3. Results

3.1. Imagery Acquisition

Figure 7 displays the canopy image in red, green, and blue (RGB) colour format, for all infestation levels. As shown in Figure 7a, the canopy was completely covered with green frond leaves, indicating that the canopy is unharmed. Figure 7b shows a canopy with low infestation that is beginning to change, particularly at the bottom of the canopy where the frond leaves have begun to dry out. Then, the foliar damage is increasing and starting to

strip most of the fronds at the bottom canopy area of Figure 7c, indicating mild infestation. Meanwhile, the severely infested canopy in Figure 7d is completely stripped, with no frond leaves remaining at the bottom of the canopy.

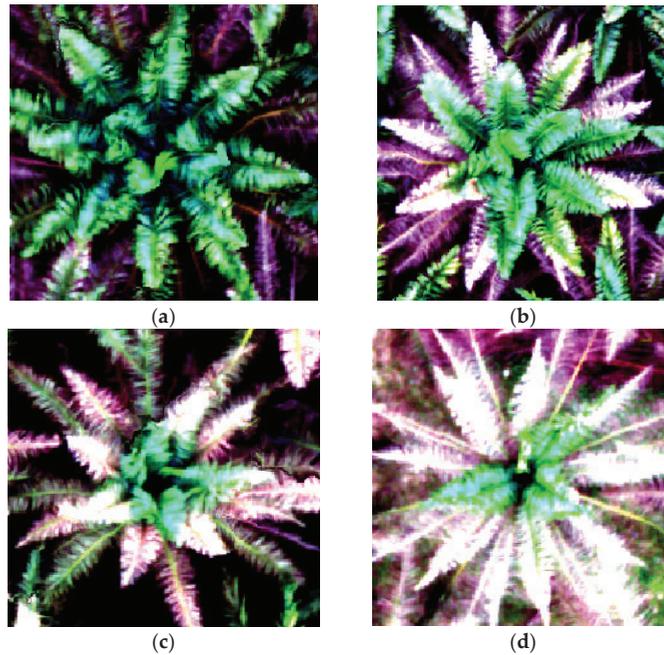


Figure 7. Condition of canopy image: (a) Healthy, (b) Low infestation, (c) Mild infestation, (d) Severe infestation.

3.2. Vegetation Indices Analysis

All these vegetation indices were then subjected to a statistical analysis to determine the variance across the means of infestation level. Results of mean (\pm standard error) comparison of the vegetation indices for each infestation category using a Tukey's HSD are tabulated in Table 6. Values that are not connected by the same letter are significantly different. According to Table 6, all the vegetation indices show consistent results where all infestation categories differ significantly. Figure 8 provides an illustration of the histogram mean comparison of each vegetation index according to the degree of infestation. As the infestation grows, it is evident that all values decrease.

Table 6. Tukey's HSD mean comparison for all vegetation indices based on infestation categories.

Infestation	NDVI	NDRE	GNDVI	SR	GCI	RECI
Healthy (n = 750)	0.9469 \pm 0.00034 ^a	0.6853 \pm 0.00130 ^a	0.8874 \pm 0.00069 ^a	38.7001 \pm 0.23590 ^a	16.6700 \pm 0.10268 ^a	4.4696 \pm 0.02413 ^a
Low (n = 800)	0.8892 \pm 0.00082 ^b	0.4908 \pm 0.00147 ^b	0.7717 \pm 0.00095 ^b	19.1967 \pm 0.16968 ^b	7.1558 \pm 0.04566 ^b	1.9789 \pm 0.01162 ^b
Mild (n = 300)	0.7816 \pm 0.00042 ^c	0.3319 \pm 0.00109 ^c	0.6380 \pm 0.00090 ^c	8.2434 \pm 0.01708 ^c	3.6012 \pm 0.01341 ^c	1.0106 \pm 0.00513 ^c
Severe (n = 1000)	0.5958 \pm 0.00180 ^d	0.1524 \pm 0.00088 ^d	0.5213 \pm 0.00149 ^d	4.1870 \pm 0.02339 ^d	2.2669 \pm 0.01316 ^d	0.3643 \pm 0.00245 ^d

Data represents the mean (\pm standard error). Different letters within the same column indicate statistical difference by the Tukey's HSD test at $p < 0.05$.

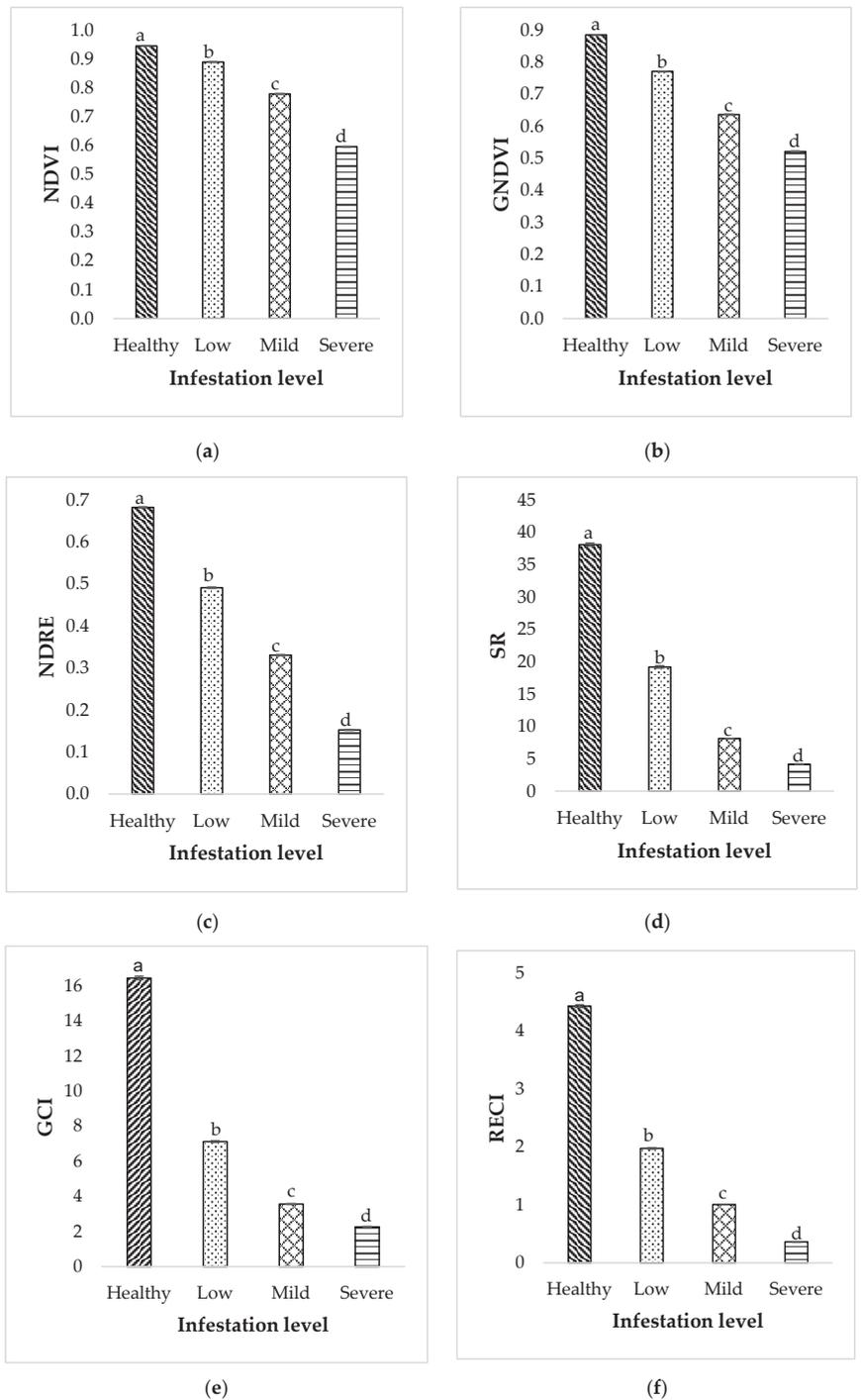


Figure 8. Average infestation level based on each vegetation index. (a) NDVI, (b) GNDVI, (c) NDRE, (d) SR, (e) GCI, and (f) RECI. Different letters within the bar chart indicate statistical difference by the Tukey’s HSD test at $p < 0.05$.

To provide a more cost-effective solution, only three significant vegetation indices with standard errors lower than 0.002 were selected, namely NDVI, NDRE, and GNDVI for model development. The combination of each selected index was then created: NDVI and NDRE, NDVI and GNDVI, and NDRE and GNDVI. As a result, these three combinations were included for model development.

3.3. Classification Model Analysis

The performance of the model was evaluated based on the F1 score value. The percentage difference between training and testing was calculated to identify the underlier of the model performance. It was determined based on the model with the smallest difference between training and testing. The tabulated results are shown in Table 7. According to Table 7, the F1 scores of all the models during training and testing were mostly high, at more than 90%. The percentage difference, meanwhile, was lower and below 1%, indicating that there was no overfitting in any of the models.

Table 7. The performance of all classifiers in all combinations, F1 score.

Classifier	Kernel	NDVI and NDRE			NDVI and GNDVI			NDRE and GNDVI		
		Train	Test	Difference (%)	Train	Test	Difference (%)	Train	Test	Difference (%)
Tree	Fine	99.65	99.89	0.24	99.32	100.00 *	0.69	99.07	99.35	0.29
	Medium	99.65	99.89	0.24	99.32	100.00 *	0.69	99.07	99.35	0.29
	Coarse	99.70	99.78	0.09	99.22	100.00 *	0.78	99.20	99.35	0.15
Discriminant	Linear	98.56	98.34	0.23	99.18	100.00 *	0.82	98.90	97.56	1.36
	Quadratic	99.61	99.14	0.47	99.50	99.08	0.43	99.52	98.62	0.90
Naïve Bayes	Gaussian	99.64	99.25	0.40	99.18	98.76	0.43	99.09	99.14	0.05
	Kernel	99.78	99.46	0.32	99.60	99.16	0.44	99.19	99.08	0.10
SVM	Linear	99.78	99.78	0.00	99.46	99.78	0.33	99.39	100.00 *	0.61
	Quadratic	99.83	100.00 *	0.17	99.47	99.78	0.31	99.39	99.78	0.40
	Cubic	99.78	99.78	0.00	99.50	100.00 *	0.50	99.39	99.35	0.04
	Fine Gaussian	99.69	100.00 *	0.31	99.51	99.82	0.31	99.39	99.46	0.07
	Medium Gaussian	99.78	99.68	0.10	99.55	99.78	0.24	99.34	100.00 *	0.66
	Coarse Gaussian	99.78	99.46	0.32	99.41	100.00 *	0.59	99.30	99.15	0.15
KNN	Fine	99.78	99.89	0.11	99.56	99.89	0.33	99.19	99.28	0.09
	Medium	99.73	99.68	0.06	99.60	100.00 *	0.41	99.26	100.00 *	0.74
	Coarse	99.65	99.25	0.41	99.55	100.00 *	0.45	99.26	99.33	0.06
	Cosine	98.22	97.34	0.89	97.89	98.40	0.53	76.64	77.03	0.51
	Cubic	99.73	99.57	0.16	99.60	100.00 *	0.41	99.34	100.00 *	0.66
	Weighted	99.73	99.89	0.16	99.61	99.89	0.29	99.39	100.00 *	0.61

* indicates the highest F1 score during testing achieved by the models in each combination.

In the combination of NDVI and NDRE, Quadratic SVM and Fine Gaussian SVM, both achieved a 100% F1 score during testing, indicating that the models successfully classify the infestation level correctly. It was then followed by a 99.98% F1 score, achieved by Fine tree, Medium tree, Fine KNN and Weighted KNN. According to their confusion matrix, the slight difference was due to ‘healthy’ being misclassified as ‘low’ with an error rate of 0.44%. The same issue was faced by another 10 models that gained an F1 score between 99.14% and 99.79% (i.e., Coarse tree, Quadratic discriminant, Gaussian Naïve Bayes, Linear SVM, Cubic SVM, Medium Gaussian SVM, Coarse Gaussian SVM, Medium KNN, Coarse KNN, and Cubic KNN) with error rates ranging between 0.89% and 4.00%. The Cosine KNN model achieved the lowest F1 score, at 97.34%, due to misclassification of the model in distinguishing between healthy and low, as well as mild being detected as severe, with error rates of 4% and 10%, respectively. Additionally, the Linear discriminant model also had a low F1 score (98.34%) because it misclassified low as healthy (2.22%) and misclassified low as mild with an error rate of 2.5%. The Kernel Naïve Bayes model obtained a 99.46% F1

score. However, it has two issues: misclassifying healthy as low and low as healthy, with error rates of 1.78% and 0.42%, respectively.

In the combination of NDVI and GNDVI, nine models, including Fine tree, Medium tree, Coarse tree, Linear discriminant, Cubic SVM, Coarse Gaussian SVM, Medium KNN, Coarse KNN, and Cubic KNN, achieved 100% F1 scores, indicating perfect classifications with zero error rates. There were two models that had an F1 score less than 99%, i.e., Gaussian Naïve Bayes (98.76%) and Cosine KNN (98.41%). Both models deal with the same issue, which is misclassification of healthy as low (error rate < 3%), as well as mild being misclassified as severe (error rate < 6%). Cosine KNN also deals with another issue, which is misclassification of low as healthy, with an error rate of 0.83%. The same problems with error rates between 0.42% and 0.83% were addressed by six additional models, namely Kernel Naïve Bayes, Linear SVM, Quadratic SVM, Medium Gaussian SVM, Fine KNN, and Weighted KNN, which all achieved F1 score ranges of 99.16% to 99.89%. Furthermore, Kernel Naïve Bayes also misclassified mild as severe with an error rate of 4.44%. Quadratic discriminant and Fine Gaussian SVM achieved F1 score ranges of 99.08% and 99.82%, respectively; however, they misclassified mild as severe with an error rate of 5.56% and 1.11%, respectively.

In the combination of NDRE and GNDVI, five models successfully achieved a 100% F1 score during testing, namely, Linear SVM, Medium Gaussian SVM, Medium KNN, Cubic KNN and Weighted KNN. On the other hand, three models achieved an F1 score lower than 99%, i.e., Quadratic discriminant (98.62%), Linear discriminant (97.56%) and Cosine KNN (77.03%). The performance of Cosine KNN in this combination was the worst because it was obviously unable to differentiate between all infestation levels, with an average error rate of 25.19%. Meanwhile, the Quadratic discriminant and Linear discriminant models had trouble, classifying healthy as low (error rate 3.11%) and low as mild (error rate 2.92%). The Quadratic discriminant model also misclassified severe as mild with an error rate of 0.67%. The other models with F1 scores between 99.00% and 99.79% essentially have the standard issue of being unable to distinguish between healthy and low, such as Fine tree, Medium tree, Coarse tree, Gaussian Naïve Bayes, Quadratic SVM, Cubic SVM, Fine Gaussian SVM, Coarse Gaussian SVM, Fine KNN, and Coarse KNN, with an average error rate of 1.64%. However, some of these models also had another issue. For example, Fine and Medium trees had error rates of 0.42% due to incorrectly classifying low as healthy. The Kernel Naïve Bayes, Coarse Gaussian SVM, and Fine KNN incorrectly misclassified severe as mild with an average error rate of 0.44%. Misclassification of low, which was predicted as mild, was also faced by Kernel Naïve Bayes, Coarse Gaussian, Gaussian Naïve Bayes and Coarse KNN, with an average error rate of 1.36%.

Overall, the performance of each model varied depending on the combination. The combination of NDVI and GNDVI was found to be the most successful in terms of perfect classification, with a 100% F1 score and zero error rate, due to nine models that accurately classified every level of infestation. It was then followed by a combination of NDRE and GNDVI with five models, and two models from NDVI and NDRE combinations. Nonetheless, all the models performed well, with an F1 score of more than 97.00% in every combination, except for Cosine KNN in the combination of NDRE and GNDVI, which performed the worst and gained an F1 score of 77.03%.

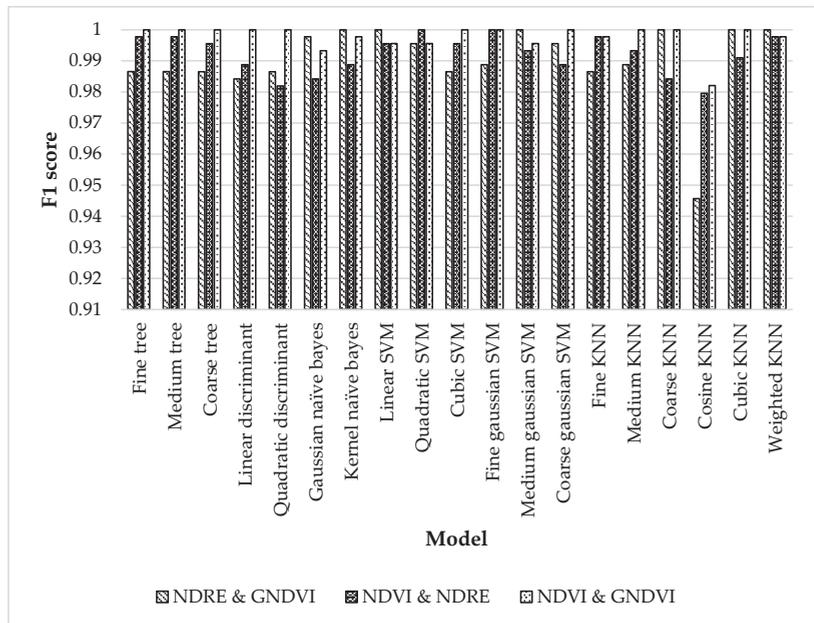
3.4. Effect of Combination of Vegetation Indices

Figure 9 shows the performance of each model in classifying the infestation level based on the combination of vegetation indices. In general, all the models performed well across all combinations. Out of 19 models, 14 models had a great performance, with an F1 score of more than 98% in classifying all infestation levels for all VI combinations, especially in classifying mild and severe levels, where all the models performed perfectly and achieved 100% F1 scores, i.e., Fine tree, Medium tree, Coarse tree, Linear SVM, Quadratic SVM, Cubic SVM, Fine Gaussian SVM, Medium Gaussian SVM, Coarse Gaussian SVM, Fine KNN, Medium KNN, Coarse KNN, Cubic KNN and Weighted KNN. The best models out

of these models were the Weighted KNN and Cubic KNN models, which worked well in all combinations and accurately classified infestation levels. The main distinction between Cubic KNN and Weighted KNN was that, when NDVI and GNDVI were combined, Cubic KNN achieved a 100% F1 score in classifying healthy and low, whereas Weighted KNN achieved 99.78% and 99.79%, respectively. Nevertheless, Weighted KNN outperformed Cubic KNN, which achieved 99.70% in classifying low and healthy in the combination of NDVI and NDRE.

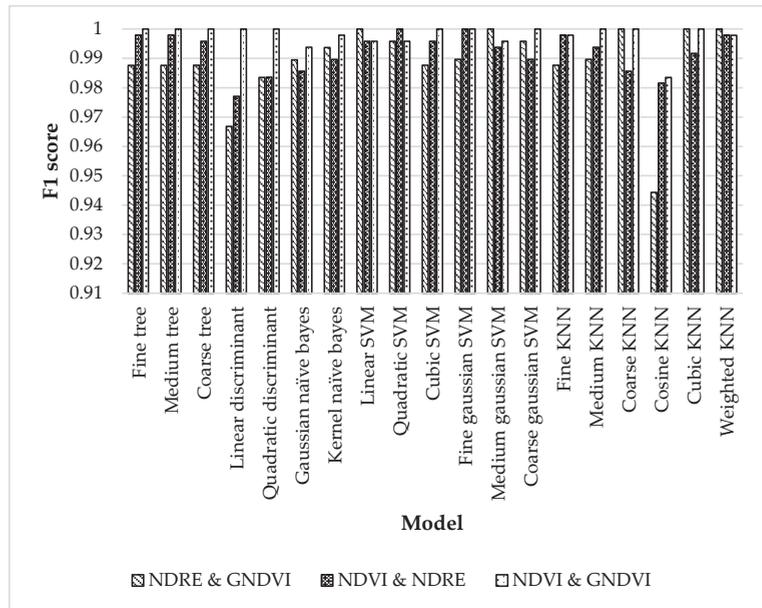
Out of all the models, Cosine KNN performed the least well, especially when combined with NDRE and GNDVI, which performed the least well at classifying all infestation levels. For instance, it gained a 32% F1 score in classifying mild, followed by 87% (severe), and 94% (healthy and low). Nevertheless, it did well in classifying a mild level in the other combinations, with an F1 score of more than 94%. Additionally, the Cosine KNN performed well when NDVI and GNDVI were combined, where all infestation levels were accurately identified and an F1 score range of 97% to 99% was obtained.

The performance of the remaining four models, which included the Linear discriminant, Quadratic discriminant, Gaussian Naïve Bayes, and Kernel Naïve Bayes, were varied, with F1 scores ranging from 95% to 100%. Generally, they successfully classified severe level in all combinations and received a perfect F1 score. In addition, they also performed well in classifying healthy and low, especially in the combination of NDVI and GNDVI, and obtained F1 scores between 96% and 100%. However, they had difficulty in categorising mild levels (F1 scores ranged between 95% and 97%), except for the NDRE and NDVI combination, where they successfully achieved a 100% F1 score.

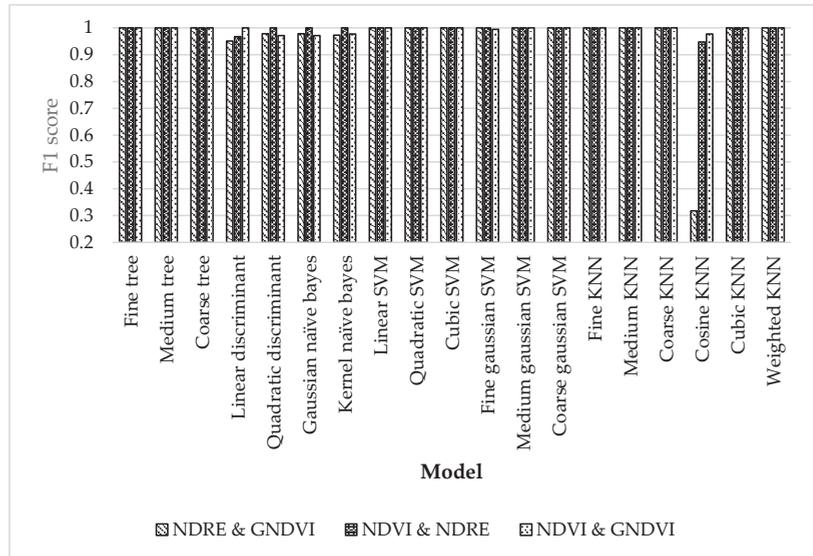


(a)

Figure 9. Cont.

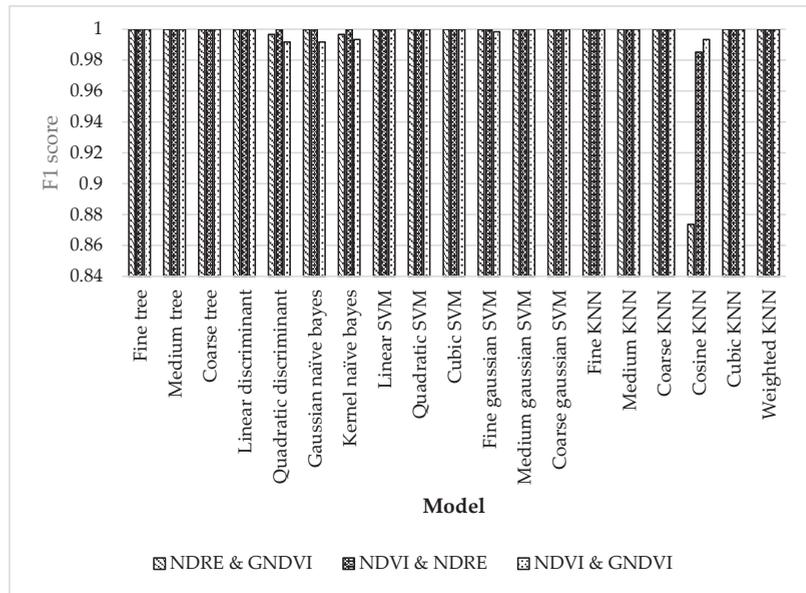


(b)



(c)

Figure 9. Cont.



(d)

Figure 9. The performance of each model in distinguishing each infestation level based on combinations of vegetation indices. (a) healthy, (b) low, (c) mild, and (d) severe.

In general, the best combination for the model to perform well in classifying the healthy and low levels of infestation and achieving a 100% F1 score was the combination of NDVI and GNDVI. For instance, 11 out of 19 models achieved a 100% F1 score in healthy level (Figure 9a), followed by the combination of NDRE and GNDVI, where only six models gained a 100% F1 score. A similar outcome was present at the low level, where most of the models obtained a 100% F1 score (11 models) when NDVI and GNDVI were combined, followed by the combination of NDRE and GNDVI (four models). For the combination of NDVI and GNDVI, it appears to be impossible for the models to classify at the healthy and low levels and achieve a 100% F1 score; instead, it is the most effective for classifying at the mild and severe levels, where 17 and 18 models out of 19 models perfectly performed well at the mild and severe levels, respectively.

The findings of this study were logical, as foliar damage in a severe condition was evidently present and gave off a brown appearance. The same is true of the mild level, where all the foliar damage began to become apparent. The crucial factor was therefore the healthy and low conditions, where the foliar damage was not readily apparent and recognised. As a result, all the models in this study successfully classified healthy and low infestation levels, particularly when NDVI and GNDVI were combined.

4. Discussion

In this study, UAV-based multispectral images were used in detecting different severity levels of bagworm infestation in oil palm plantations. This study employs UAV-acquired images and machine learning techniques to locate the bagworm *Metisa plana* infestation area. It focuses exclusively on the ability of the machine learning to categorise the severity level of infestation as healthy, low infestation, mild infestation, and severe infestation using vegetation indices extracted from UAV images. Out of five vegetation indices, three were selected and formed three combinations: NDVI and NDRE, NDVI and GNDVI, and NDRE and GNDVI. A total of 19 models were used to determine the effectiveness of the combination dataset to classify each infestation level.

Weighted KNN was chosen out of all the models used due to its highly consistent performance and the great classification of all infestation levels (F1 score greater than 99.70%). It was then followed by Cubic KNN, which had an F1 score of over 99.10%. Meanwhile, the Cosine KNN model was chosen as having the least effective performance among the others with an F1 score range between 32% and 94%. It was clearly demonstrated that the same classifier, using a different kernel, produced the best and worst performance. KNN typically works by using the distance function to determine how far new data entry is from values provided in datasets with different classes based on its closeness in the given range (k) of neighbours. In this study, the k -neighbour was constant and set at 10, indicating medium distinctions between classes. Meanwhile, the distance function was based on the kernel type of the KNN. For instance, Weighted KNN uses the distance-weighting concept, where the weighing is calculated using Euclidean distances. Cosine KNN and Cubic KNN use cosine distance and cubic distances, respectively. This study clearly demonstrated that the weighted kernel provided an excellent result due to the addition of weights to the Euclidean distance, which enhances classification performance. The same verdict was obtained by Mohd Johari et al. [29], in differentiating the four larval instar stages with an accuracy of 91% to 95%. In addition, Rathore et al. [40] also found that KNN, using weighted kernel, achieved a high accuracy of 90% compared to other kernels in distinguishing between various type of insects and between adult and larvae insect sounds.

Furthermore, the best combination of vegetation indices was determined to be NDVI and GNDVI, as most models could successfully classify the level of infestation and achieved a 100% F1 score, especially in healthy and low levels. The same outcome was obtained by Mangewa et al. [41], where the NDVI and GNDVI were determined to be the most effective vegetation indices for detecting and monitoring ecological changes in wildlife habitat condition classes (i.e., very good, good, poor and very poor). Generally, NDVI is most useful when used to assess vegetation density over large areas and to assess crop health; meanwhile, GNDVI is based on the greenness level, which is determined by the radiance of the leaf surface and is a significant indicator in distinguishing between healthy and infested leaves. Moreover, the combination of NDVI and NDRE was found to be the most suitable combination for the models in classifying mild and severe levels and achieved an F1 score of 100%. A comparable finding was presented by Boiarskii and Hasegawa [42], who used NDVI and NDRE to identify the poorly growing vegetation area and demonstrated that NDRE was sensitive to chlorophyll content, indicating nitrogen limitation in the leaves. Hence, it can be inferred that the results obtained from this research, which showcased the effectiveness of utilising the combination of NDVI and NDRE, as well as NDVI and GNDVI, in achieving optimal classification performance for identifying low and severe infestation, as well as healthy and low infestation, respectively, are deemed satisfactory.

In terms of spectral bands, NIR makes up all the vegetation indices used in this study. Following that, the NIR band was tested against other bands such as red, green, and red edge using the same arithmetic operation (subtraction, division, and addition). The combination of NIR and red band formed NDVI; NIR and green band formed GNDVI; and NIR and red edge formed NDRE. The combination of NDVI and NDRE obtained the best results in classifying between mild and severe infestation, as well as a combination of NDVI and GNDVI obtained the best results in classifying healthy and low. In this study, NDVI was recognised as a crucial vegetation index because it aids in classifying all levels of infestation. The red band was regarded as a crucial band in addition to NIR because it also affects the performance of all classifications. Thus, the performance of all classification models is clearly boosted by the combination of NIR and red bands.

Nevertheless, a severe condition seems easy to identify, as the foliar damage is obvious. As mentioned by Corley and Tinker [5], the lower and central crown appeared greyish brown as a result of the severely damaged leaves. This is consistent with the findings of this study, which showed that most of the models were correctly classified as severe levels and had 100% F1 scores in all combinations. In this case, healthy and low infestation levels

play a crucial role, as it is difficult to detect the starting point of the infestation. As a result, this suggested method has an excellent chance of identifying healthy and low infestation levels even when the foliar damage is unseen, and there are no colour changes of the frond. Therefore, decision-making models may be able to tell farmers when to start pest control measures to stop the spread of the pest, especially for early infection predictions.

5. Conclusions

In this study, *Metisa plana* infestation levels were classified using UAV images and a machine learning approach. To enhance the classification performance of each model in classifying the level of infestation, three types of combinations among chosen vegetation indices were developed, namely NDVI and NDRE, NDVI and GNDVI, and NDRE and GNDVI. According to the results, the best combination for classifying healthy and low levels was found to be NDVI and GNDVI, empowering the model to classify all infestation levels with a 100% F1 score. In addition, the combination of NDVI and NDRE was found to be the best combination for classifying mild and severe levels. The most important vegetation index that could detect every level of infestation was NDVI. The classification of the infestation level is made clearer and more accurate by combining it with other vegetation indices. In addition, Weighted KNN became the best model, which constantly gave the best performance in classifying all the infestation levels (F1 score > 99.70%) in all combinations.

Early detection of a bagworm infestation is crucial for effective management and early control measures. Therefore, this suggested method is essential for the early phase of severity level detection, considering that the infestation level can be automatically identified, allowing the planning and management of the control measure to be planned more quickly. Furthermore, the outcomes of this study demonstrated the enormous potential of UAV synergy for the detection of pest infestations using machine learning. Transfer learning methodology could be used in future studies to provide more automatic classification.

Author Contributions: Conceptualization, S.N.A.M.J. and S.K.-B.; methodology, S.N.A.M.J. and S.K.-B.; software, S.K.-B.; formal analysis, S.N.A.M.J. and S.K.-B.; investigation, S.N.A.M.J.; resources, M.M.M.M. and N.K.; data curation, S.N.A.M.J.; writing—original draft preparation, S.N.A.M.J.; writing—review and editing, S.K.-B.; supervision, S.K.-B., A.R.M.S., N.A.H. and M.M.M.M.; project administration, S.K.-B.; funding acquisition, S.K.-B. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Ministry of Higher Education Malaysia (MOHE) under Fundamental Research Grants Scheme (FRGS) (Project number: FRGS/1/2018/TK04/UPM/02/4) and the Graduate Study and Research in Agriculture (SEARCA) under Professorial Chair Award 2022–2023 in the field of Imaging Technology and Remote Sensing.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to restrictions.

Acknowledgments: The authors would like to thank the MPOB for providing sample data.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Norman, K.; Basri, M.W. Status of common oil palm insect pests in relation to technology adoption. *Planter* **2007**, *83*, 371–385.
2. Cheong, Y.; Tey, C.C. Environmental Factors which Influence Bagworm outbreak. In Proceedings of the 5th MPOB-IOPRI International Seminar, Kuala Lumpur, Malaysia, 22–23 November 2013; Volume 2013.
3. Bagworm Infestation in District Causing Palm Oil Production to Drop. Available online: <https://www.thestar.com.my/news/community/2012/11/21/bagworm-infestation-in-district-causing-palm-oil-production-to-drop> (accessed on 12 June 2022).
4. Chung, G.F. Effect of Pests and Diseases on Oil Palm Yield. In *Palm Oil*; AOCS Press: Boulder Urbana, IL, USA; Elsevier Inc.: Amsterdam, The Netherlands, 2012; pp. 163–210.
5. Corley, R.H.V.; Tinker, P.B. Pests of the Oil Palm. In *The Oil Palm*; John Wiley & Sons: Hoboken, NJ, USA, 2019; pp. 437–459. [CrossRef]

6. Wood, B.J.; Kamarudin, N. Bagworm (Lepidoptera: Psychidae) infestation in the centennial of the Malaysian oil palm industry—A review of causes and control. *J. Oil Palm Res.* **2019**, *31*, 364–380. [CrossRef]
7. Tuck, H.C.; Ibrahim, Y.; Chong, K.K. Infestations by the bagworms *metisa plana* and *pteroma pendula* for the period 1986–2000 in major oil palm estates managed by golden hope plantation berhad in peninsular malaysia. *J. Oil Palm Res.* **2011**, *23*, 1040–1050.
8. Aziz, N.A.; Omar, W.; Kassim, R.; Kamarudin, N. Remote Sensing Measurement for Detection of Bagworm Infestation in Oil Palm Plantation. *MPOB Inf. Ser.* **2012**, *502*, 589–597.
9. Standard Operating Procedures (SOP) Guidelines for Bagworm Control. Malaysian Palm Oil Board (MPOB): Bangi, Malaysia, 2016.
10. Kamarudin, N.; Ahmad Ali, S.R.; Mohd Masri, M.M.; Ahmad, M.N.; Che Manan, C.A.H.; Kamarudin, N. Controlling *Metisa plana* Walker (Lepidoptera: Psychidae) outbreak using *Bacillus thuringiensis* at an oil palm plantation. *J. Oil Palm Res.* **2019**, *29*, 47–54. [CrossRef]
11. Kim, J.; Kim, S.; Ju, C.; Il Son, H. Unmanned aerial vehicles in agriculture: A review of perspective of platform, control, and applications. *IEEE Access* **2019**, *7*, 105100–105115. [CrossRef]
12. Candiago, S.; Remondino, F.; De Giglio, M.; Dubbini, M.; Gattelli, M. Evaluating multispectral images and vegetation indices for precision farming applications from UAV images. *Remote Sens.* **2015**, *7*, 4026–4047. [CrossRef]
13. Xiang, H.; Tian, L. Development of a low-cost agricultural remote sensing system based on an autonomous unmanned aerial vehicle (UAV). *Biosyst. Eng.* **2011**, *108*, 174–190. [CrossRef]
14. Guimarães, N.; Pádua, L.; Marques, P.; Silva, N.; Peres, E.; Sousa, J.J. Forestry remote sensing from unmanned aerial vehicles: A review focusing on the data, processing and potentialities. *Remote Sens.* **2020**, *12*, 1046. [CrossRef]
15. Gutierrez-Rodriguez, M.; Escalante-Estrada, J.A.; Rodriguez-Gonzalez, M.T. Canopy Reflectance, Stomatal Conductance, and Yield of *Phaseolus vulgaris* L. and *Phaseolus coccineus* L. Under Saline Field Conditions. *Int. J. Agric. Biol.* **2005**, *7*, 491–494.
16. Klouček, T.; Komárek, J.; Surový, P.; Hrach, K.; Janata, P.; Vašíček, B. The use of UAV mounted sensors for precise detection of bark beetle infestation. *Remote Sens.* **2019**, *11*, 1561. [CrossRef]
17. Minařík, R.; Langhammer, J.; Lenzioch, T. Automatic tree crown extraction from uas multispectral imagery for the detection of bark beetle disturbance in mixed forests. *Remote Sens.* **2020**, *12*, 4081. [CrossRef]
18. Tsouros, D.C.; Bibi, S.; Sarigiannidis, P.G. A review on UAV-based applications for precision agriculture. *Information* **2019**, *10*, 349. [CrossRef]
19. Marin, D.B.; Ferraz, G.A.S.; Santama, L.S.; Barbosa, B.D.S.; Barata, R.A.P.; Osco, L.P.; Ramos, A.P.M.; Guimaraes, P.H.S. Detecting coffee leaf rust with UAV-based vegetation indices and decision tree machine learning models. *Comput. Electron. Agric.* **2021**, *190*, 106476. [CrossRef]
20. Su, J.; Liu, C.; Coombes, M.; Hu, X.; Wang, C.; Xu, C.; Li, Q.; Guo, L.; Chen, W.H. Wheat yellow rust monitoring by learning from multispectral UAV aerial imagery. *Comput. Electron. Agric.* **2018**, *155*, 157–166. [CrossRef]
21. DadrasJavan, F.; Samadzadegan, F.; Seyed Pourazar, S.H.; Fazeli, H. UAV-based multispectral imagery for fast Citrus Greening detection. *J. Plant Dis. Prot.* **2019**, *126*, 307–318. [CrossRef]
22. Wang, T.; Thomasson, J.A.; Yang, C.; Isakeit, T.; Nichols, R.L. Automatic classification of cotton root rot disease based on UAV remote sensing. *Remote Sens.* **2020**, *12*, 1310. [CrossRef]
23. Yu, R.; Luo, Y.; Zhou, Q.; Zhang, X.; Wu, D.; Ren, L. A machine learning algorithm to detect pine wilt disease using UAV-based hyperspectral imagery and LiDAR data at the tree level. *Int. J. Appl. Earth Obs. Geoinf.* **2021**, *101*, 102363. [CrossRef]
24. Haw, Y.H.; Lai, K.W.; Chuah, J.H.; Khairunniza-Bejo, S.; Husin, N.A.; Hum, Y.C.; Yee, P.L.; Tee, C.A.T.H.; Ye, X.; Wu, X. Classification of basal stem rot using deep learning: A review of digital data collection and palm disease classification methods. *PeerJ. Comput. Sci.* **2023**, *9*, e1325. [CrossRef]
25. Khosrokhani, M.; Khairunniza-Bejo, S.; Pradhan, B. Geospatial technologies for detection and monitoring of *Ganoderma* basal stem rot infestation in oil palm plantation: A review on sensors and techniques. *Geocarto Int.* **2018**, *33*, 260–276. [CrossRef]
26. Viera-Torres, M.; Sinde-González, I.; Gil-Docampo, M.; Bravo-Yandún, V.; Toulkeridis, T. Generating the baseline in the early detection of bud rot and red ring disease in oil palms by geospatial technologies. *Remote Sens.* **2020**, *12*, 3229. [CrossRef]
27. Anuar, I.M.; Arof, H.; Nor, N.M.; Hashim, Z.; Seman, I.A.; Masri, M.M.M.; Ibrahim, M.S.; Tat, E.H.; Toh, C.T. Remote Sensing for Detection of *Ganoderma* Disease and Bagworm Infestation in Oil Palm. *Adv. Agric. Food Res. J.* **2021**, *2*, 1–14. [CrossRef]
28. Ahmad, M.N.; Mohamed Shariff, A.R.; Aris, I.; Abdul Halin, I.; Moslim, R. Identification and determination of the spectral reflectance properties of live and dead bagworms, *Metisa plana* Walker (Lepidoptera: Psychidae) using Vis/NIR spectroscopy. *J. Oil Palm Res.* **2020**, *33*, 425–435. [CrossRef]
29. Mohd Johari, S.N.A.; Khairunniza-Bejo, S.; Shariff, A.R.M.; Husin, N.A.; Masri, M.M.M.; Kamarudin, N. Identification of bagworm (*Metisa plana*) instar stages using hyperspectral imaging and machine learning techniques. *Comput. Electron. Agric.* **2022**, *194*, 106739. [CrossRef]
30. Mohd Johari, S.N.A.; Khairunniza-Bejo, S.; Shariff, A.R.M.; Husin, N.A.; Masri, M.M.M.; Kamarudin, N. Automatic Classification of Bagworm, *Metisa plana* (Walker) Instar Stages Using a Transfer Learning-Based Framework. *Agriculture* **2023**, *13*, 442. [CrossRef]
31. Thaer, S.; Kassim, F.A.; Hasbullah, N.A.; Al-obaidi, J.R. Evaluation of bagworm, *Metisa plana* (Lepidoptera: Psychidae) infestation and beneficial parasitoid in an oil palm plantation, Perak, Malaysia. *J. Sci. Math. Lett.* **2021**, *9*, 19–35.

32. Duarte, A.; Borrvalho, N.; Cabral, P.; Caetano, M. Recent Advances in Forest Insect Pests and Diseases Monitoring Using UAV-Based Data: A Systematic Review. *Forests* **2022**, *13*, 911. [CrossRef]
33. Manfreda, S.; McCabe, M.F.; Miller, P.E.; Lucas, R.; Madrigal, V.P.; Mallinis, G.; Dor, E.B.; Helman, D.; Estes, L.; Ciraolo, G.; et al. On the use of unmanned aerial systems for environmental monitoring. *Remote Sens.* **2018**, *10*, 641. [CrossRef]
34. Rouse, J.W.; Haas, R.H.; Schell, J.A.; Deering, D.W. Monitoring vegetation system in the great plains with erts. *J. Agric. Food Chem.* **1976**, *24*, 24–26. [CrossRef]
35. Gitelson, A.A.; Kaufman, Y.J.; Merzlyak, M.N. Use of a green channel in remote sensing of global vegetation from EOS- MODIS. *Remote Sens. Environ.* **1996**, *58*, 289–298. [CrossRef]
36. Gitelson, A.A.; Merzlyak, M.N. Quantitative estimation of chlorophyll-a using reflectance spectra: Experiments with autumn chestnut and maple leaves. *J. Photochem. Photobiol. B Biol.* **1994**, *22*, 247–252. [CrossRef]
37. Baret, F.; Guyot, G. Potentials and limits of vegetation indices for LAI and APAR assessment. *Remote Sens. Environ.* **1991**, *35*, 161–173. [CrossRef]
38. Gitelson, A.A.; Viña, A.; Ciganda, V.; Rundquist, D.C.; Arkebauer, T.J. Remote estimation of canopy chlorophyll content in crops. *Geophys. Res. Lett.* **2005**, *32*, 1–4. [CrossRef]
39. Arlot, S.; Celisse, A. A survey of cross-validation procedures for model selection. *Stat. Surv.* **2010**, *4*, 40–79. [CrossRef]
40. Rathore, D.S.; Ram, B.; Pal, B.L.; Malviya, S. Analysis of Classification Algorithms for Insect Detection using MATLAB. In Proceedings of the 2nd International Conference on Advanced Computing and Software Engineering (ICACSE) 2019, Sultanpur, India, 8–9 February, 2019; Volume 2019, pp. 299–304. [CrossRef]
41. Mangewa, L.J.; Ndakidemi, P.A.; Alward, R.D.; Kija, H.K.; Bukombe, J.K.; Nasolwa, E.R.; Munishi, L.K. Comparative Assessment of UAV and Sentinel-2 NDVI and GNDVI for Preliminary Diagnosis of Habitat Conditions in Burunge Wildlife Management Area Tanzania. *Earth* **2022**, *3*, 769–787. [CrossRef]
42. Boiarskii, B.; Hasegawa, H. Comparison of NDVI and NDRE Indices to Detect Differences in Vegetation and Chlorophyll Content. *J. Mech. Contin. Math. Sci.* **2019**, *4*, 20–29. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.



Article

SpikoPoniC: A Low-Cost Spiking Neuromorphic Computer for Smart Aquaponics

Ali Siddique ^{1,2,*}, Jingqi Sun ³, Kung Jui Hou ^{4,5}, Mang I. Vai ^{1,2,5}, Sio Hang Pun ¹
and Muhammad Azhar Iqbal ⁶

¹ State Key Lab of Analog and Mixed Signal VLSI (AMSV), University of Macau, Macau 999078, China; fstmiv@um.edu.mo (M.I.V.); lodgepun@um.edu.mo (S.H.P.)

² Department of Electrical and Computer Engineering, Faculty of Science and Technology, University of Macau, Macau 999078, China

³ Department of Computer Science, Beijing Jiaotong University, Weihai 264003, China; jingqisun@bjtu.edu.cn

⁴ Lingyange Semiconductor Inc., Zhuhai 519031, China; ray.hou@lyg-semi.com

⁵ ZUMRI-LYG Joint Lab, Zhuhai UM Science and Technology Research Institute, Zhuhai 519031, China

⁶ School of Computing, Faculty of Engineering and Physical Sciences, University of Leeds, Leeds LS2 9JT, UK; m.a.iqbal1@leeds.ac.uk

* Correspondence: yb97482@umac.mo

Abstract: Aquaponics is an emerging area of agricultural sciences that combines aquaculture and hydroponics in a symbiotic way to enhance crop production. A stable smart aquaponic system requires estimating the fish size in real time. Though deep learning has shown promise in the context of smart aquaponics, most smart systems are extremely slow and costly and cannot be deployed on a large scale. Therefore, we design and present a novel neuromorphic computer that uses spiking neural networks (SNNs) for estimating not only the length but also the weight of the fish. To train the SNN, we present a novel hybrid scheme in which some of the neural layers are trained using direct SNN backpropagation, while others are trained using standard backpropagation. By doing this, a blend of high hardware efficiency and accuracy can be achieved. The proposed computer *SpikoPoniC* can classify more than 84 million fish samples in a second, achieving a speedup of at least 3369× over traditional general-purpose computers. The *SpikoPoniC* consumes less than 1100 slice registers on Virtex 6 and is much cheaper than most SNN-based hardware systems. To the best of our knowledge, this is the first SNN-based neuromorphic system that performs smart real-time aquaponic monitoring.

Keywords: artificial intelligence; deep learning; digital agriculture; Internet of Things (IoT); neuromorphic chips; on-chip learning; precision agriculture; smart aquaponics; smart farming; spiking neural networks

Citation: Siddique, A.; Sun, J.; Hou, K.J.; Vai, M.I.; Pun, S.H.; Iqbal, M.A. *SpikoPoniC: A Low-Cost Spiking Neuromorphic Computer for Smart Aquaponics*. *Agriculture* **2023**, *13*, 2057. <https://doi.org/10.3390/agriculture13112057>

Academic Editors: Gniewko Niedbala, Sebastian Kujawa, Magdalena Piekutowska and Tomasz Wojciechowski

Received: 5 September 2023

Revised: 8 October 2023

Accepted: 10 October 2023

Published: 27 October 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The world is witnessing a rapid increase in population and a decrease in the amount of available food. The food sources are depleting faster than ever before. This is why it is crucial to develop advanced agricultural techniques. Aquaponics is a recently developed agricultural technique that combines aquaculture and hydroponics in a symbiotic way to increase the overall food production and quality. In an aquaponic system, fish excrete their waste, which is transformed into nutrients by nitrifying bacteria, which in turn are readily absorbed by plants. This symbiotic relationship is shown in Figures 1 and 2. Smart aquaponic systems (SASs) first take input from sensors that collect data about the environment and then use that information for making critical decisions such as controlling the level of oxygen and nutrients in the ecosystem. The parameters may include temperature, pH level, and oxygen level. In Figure 2, the symbol N represents the number of input features, and p is a symbol for the p th hidden neuron.

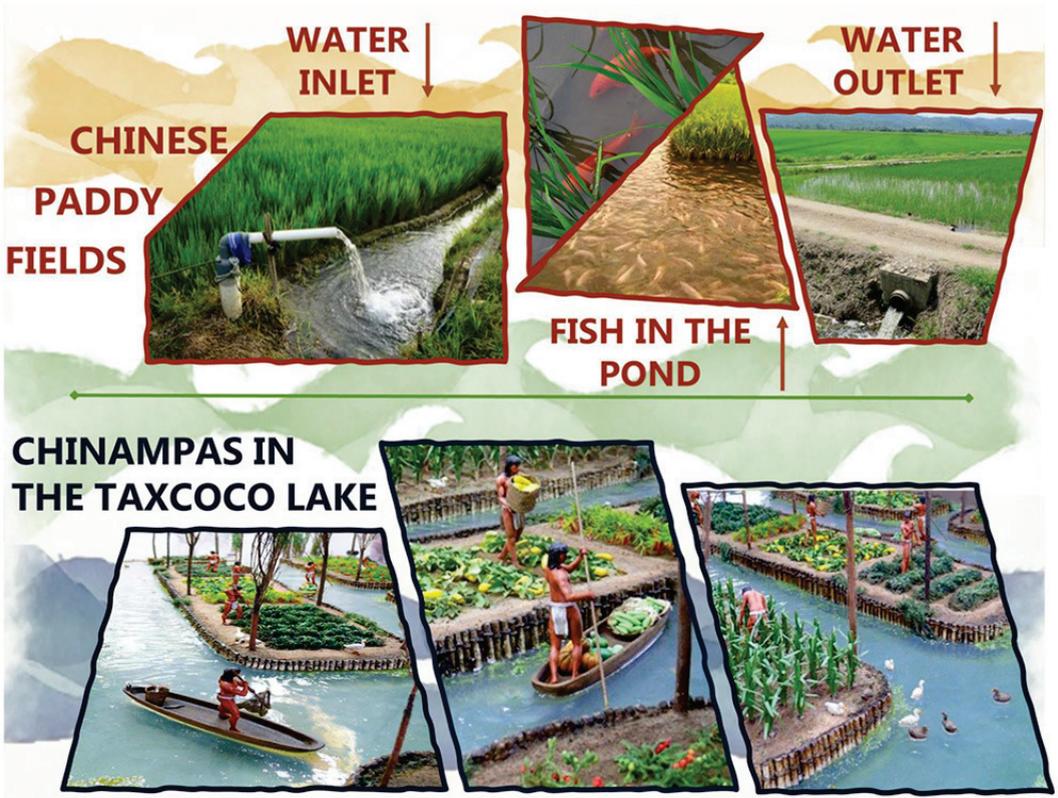


Figure 1. A real-world aquaponic system [1].

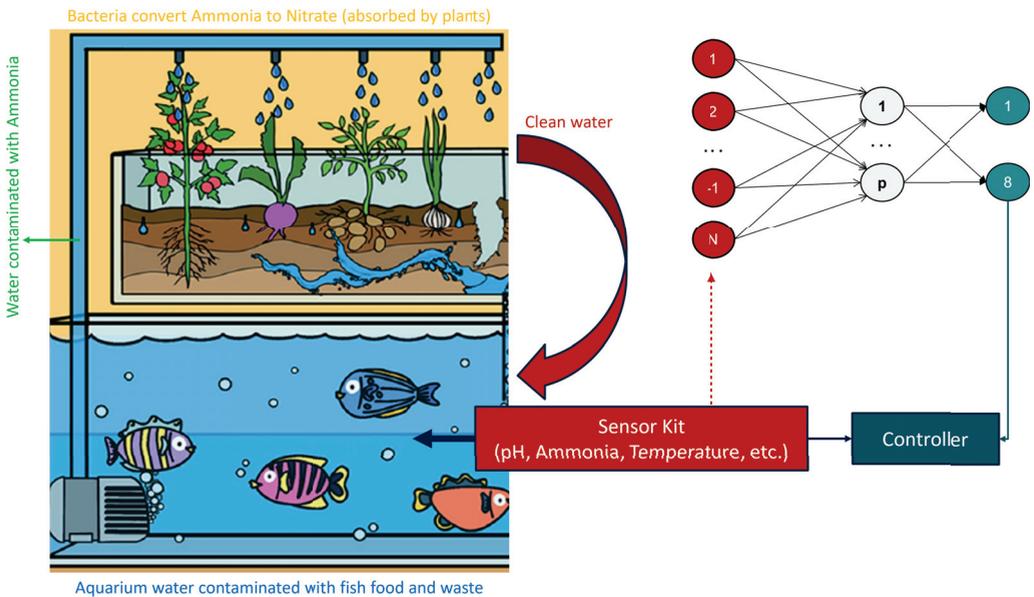


Figure 2. A typical smart aquaponic system using deep learning for feedback.

Aquaponics is safer for environment than the traditional farming techniques because it limits the use of harmful chemicals [2]. The use of aquaponics can greatly resolve the water scarcity crisis as the water consumed by aquaponic systems is only 2–10% of what is used by the traditional agricultural systems. Aquaponics also plays a good role in promoting soil-less culture [2,3]. Despite all these advantages, only 31% of the aquaponic solutions have been found to be commercially viable due to poor management and inexperienced handlers [2]. This is where the role of deep learning (DL) comes into play. Deep learning is a proven tool to monitor and control nutrients in an aquaponic system. DL can help in the early prediction of fish size and water quality, which can in turn help in the automatic adjustment and upgradation of system parameters. For example, if a fish is smaller than the expected size, it can be provided with more food and healthy nutrition. In this context, various deep learning models have been proposed to predict water quality parameters, fish classification, fish size estimation, feeding decisions, etc. [2,4–7].

Since large-scale aquaponic systems have to deal with a huge number of parameters and require extensive monitoring, it is quite hard for humans to do all the manual work. The problem becomes more severe when there is a shortage of experienced manpower. In order to resolve these issues, automation techniques and artificial-intelligence-based systems can be extremely helpful. Such techniques and systems reduce the need for huge manpower and allow for better management at a commercial scale. Smart Systems (SSs), in the context of aquaponic systems, refer to small but intelligent electronic devices capable of performing a huge number of complex operations such as sensing, monitoring, and control in a minimal amount of time [2].

Although the literature is filled with works describing the development of smart DL systems for aquaponics, no intelligent DL system is based on dedicated hardware (HW). Here, the term *dedicated hardware* refers to field programmable gate arrays (FPGAs) and application-specific integrated circuits (ASICs). All the systems are implemented using microcontrollers (uCs) and/or general-purpose computers (GPCs) that are inherently slow. This is because uCs and GPCs follow a sequential model of execution. As a result, a high degree of parallelization in such systems is almost impossible. Even if there is a certain degree of parallelization in any such system, it can never match the level of parallelization offered by dedicated HW systems. Due to their slow speed, the use of uCs and GPCs in large-scale commercial aquaponic systems—where millions of parameters have to be handled in real time—can become quite problematic. There is another problem associated with the use of GPCs: they have a large area and require expensive software tools for proper operation. This certainly increases their cost and footprint and limits their usage in practical systems. FPGAs and ASICs, on the other hand, have smaller footprints, offer a higher speed, and have lower power requirements. This has been demonstrated even in this article: the use of the FPGA-based DL system is about 3369 times faster than traditional general-purpose computers. Figure 3 shows the relationship between various computing platforms in terms of efficiency and flexibility. As can be seen in Figure 3, ASICs and FPGAs are quite difficult to develop but have higher efficiency, i.e., lower power consumption, higher speed, and smaller area. Dedicated HW systems have a huge number of parallel computational elements, which makes them suitable for high speed applications. This is the basic advantage of using neuromorphic systems/accelerators (NMSs/NMAs). NMAs are accelerators that use dedicated hardware devices (FPGAs/ASICs) to implement neural networks. By doing so, a lot of improvement in system speed, energy consumption, and area can be obtained [8,9]. Section 1.2 discusses various modern neuromorphic accelerators.

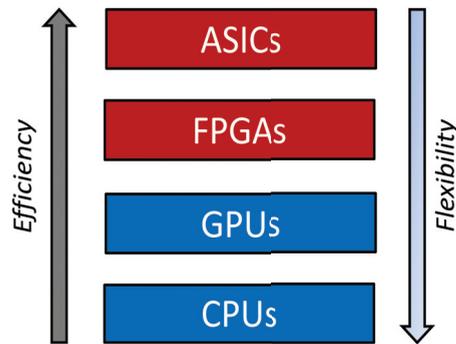


Figure 3. Flexibility–efficiency relationship between various computing platforms.

There are some other shortcomings as well in the currently available works. One such shortcoming is that all these works (aquaponic systems) use artificial neural networks (ANNs) that are quite expensive and compute-intensive, especially when implemented on dedicated hardware devices such as ASICs. Though ANNs have exhibited excellent performance in the context of both classification and regression, they consume a lot of system energy and are extremely complex [10]. Spiking neural networks (SNNs), on the other hand, are more efficient than ANNs when it comes to hardware efficiency [8] since SNNs are event driven and consume lesser energy than ANNs. The primary difference between an ANN and an SNN is that the former uses continuous analog values, while the latter uses spikes for neuronal communication. These spikes are emitted by a neuron only when an important event takes place, otherwise nothing is emitted and the subsequent units remain silent [8]. This behavior is similar to the biological neurons [11,12]. In fact, the asynchronous behavior of biological neurons is the reason why the brain consumes only a few watts of energy despite having a very small area [11]. Therefore, SNNs are more bioplausible than ANNs [12,13]. In fact, since SNNs use simple spikes, they might have a very small hardware footprint, as in [9,14]. Moreover, SNNs can achieve almost the same level of accuracy as ANNs, as can be seen in [14–16]. This is why it is the need of the hour to develop SNN-based aquaponic systems. Another big problem in most modern studies is the lack of available data. Most datasets have a few hundred samples for system evaluation, which is insufficient to obtain reliable results. For example, the work in [3] uses only 211 samples for performance evaluation. SNN training is also a very serious issue since spikes are non-differentiable and backpropagation is not easy to apply [15,17]. Moreover, ANN-SNN conversion requires a lot of costly operations such as weight-threshold balancing [18].

Keeping all these issues in mind, we propose a novel, high-speed and small-footprint smart aquaponic system based on a spiking neural network for fish size estimation. The system can monitor and predict the size of fishes in real time. The system can classify a given input sample into one of the eight classes defined with respect to weight or length. The system can estimate both the expected length and weight of fishes based on input data. The throughput of the system is 84.23 million samples per second, i.e., the system can classify 84.23 million samples in a second. Moreover, the proposed SNN system consumes 45% fewer hardware resources than its ANN counterpart for the same level of accuracy. In fact, the proposed system is $2\times$ – $3\times$ cheaper than most contemporary systems.

Main Contributions

This paper presents a novel, hardware-efficient, SNN-based neuromorphic engine for fish size estimation, which finds application in aquaponic monitoring. The system has been modeled in Verilog language at the register-transfer level (RTL). The main contributions of this work are mentioned below. A summary of these contributions is given in Figure 4.

1. To provide a complete methodology to develop an aquaponic monitoring system that uses spiking neural networks to predict fish size. The system is capable of predicting both the length and weight of a fish, unlike other systems that can predict either length or weight, not both. In the proposed system, this is done using two *switching buffers*: one for predicting the weight, and the other one for predicting length.
2. A proposal of a novel hybrid training scheme that uses both ANN and SNN layers to achieve a blend of high accuracy and hardware efficiency. The system uses direct training for SNNs and standard backpropagation for ANN layers. The proposed implementation is much more hardware-efficient not only than a typical, fully ANN implementation but other SNN implementations too, without any loss of accuracy. The system can estimate the range of length with more than 98.03% accuracy, and the range of weight with 99.67% accuracy.
3. The system does not use any complex weight-threshold balancing mechanism, unlike various SNN training schemes [18,19], since it uses direct SNN backpropagation.
4. An SNN-based neuromorphic system implemented on a field programmable gate array (FPGA) for real-time aquaponic monitoring. It is an edge computer capable of predicting fish size (length and weight) on the basis of input parameters. The proposed edge computer can predict 8 different fish size categories based on the given data. The system can operate in the ‘fully parallel’ mode and can estimate 84.23 million samples in a second. The throughput is about 3369 times higher than a typical CPU-based software system, making it suitable for large-scale commercial use. While other systems use only a few hundred samples for testing purposes, the proposed system has been trained/tested on 175,000 samples, which proves that the obtained results are more reliable than others’.
5. The proposal of a hardware-efficient surrogate gradient that is as efficient as sigmoid but has higher flexibility. The mean-squared error between the sigmoidal derivative and the proposed derivative is 0.013%. The learning technique is suitable for developing on-chip learning (OCL) systems since the proposed surrogate gradient requires far fewer hardware resources than most gradients proposed in the literature while being extremely accurate.

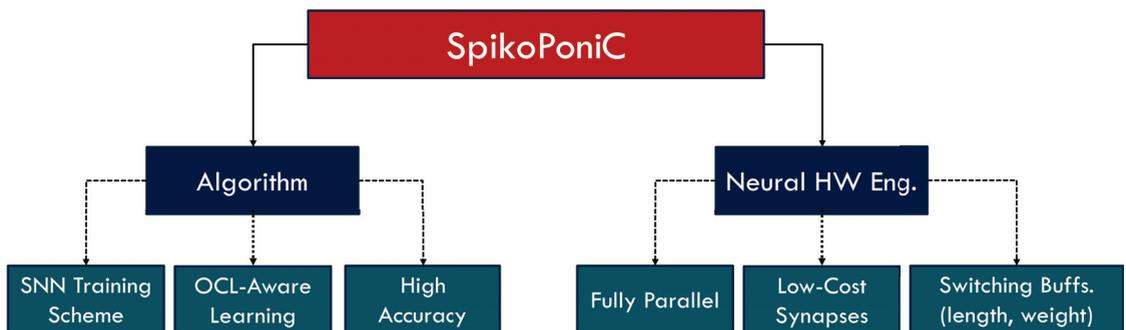


Figure 4. Proposed system: features and components.

The rest of this paper is organized as follows. The coming parts of this section present a review of various modern smart aquaponic systems and neuromorphic accelerators. It also presents the problem definition. Section 2 presents the training methodology for fish size estimation. Section 2 also presents the operation and details of SpikoPoniC, the proposed neuromorphic computer. The results are presented in Section 3. Finally, Section 4 concludes the work.

Related Work and Problem Definition

Smart aquaponic systems (SASs) are generally built using GPCs and microcontrollers (MCs) [2]. We could not find any SAS built using FPGAs/ASICs in the literature. This is where the problem lies. Modern SASs are quite complex, and MCs or GPCs are not fast enough to operate at the required speed [20,21]. Moreover, MCs and GPCs consume a lot of power and can be extremely expensive when manufactured in bulk quantities. Therefore, we chose to implement the proposed design on an FPGA. The same design can be implemented on an ASIC as well, and higher hardware efficiency can be achieved. Furthermore, we focus not only on the algorithm but also on the hardware design. This approach has been proven to be extremely effective at developing high-performance systems [9,22] since hardware efficiency is directly related to algorithmic efficiency.

Since the proposed scheme SpikoPoniC deals with both algorithm and architecture, we divide this section into two major parts. The first part presents a comprehensive review of modern smart aquaponic systems, and the second part deals with the neuromorphic accelerators. Based on the literature review, we conclude that SNNs have not yet been used for building SASs. In fact, none of the NM accelerators have been designed specifically for aquaponics.

1.1. Smart Aquaponics: Algorithms and Monitoring Systems

In the context of smart aquaponics, deep learning finds its applications, among others, in the following areas: fish size estimation, water quality prediction, plant disease diagnosis, and intelligent feeding decisions. The estimation of fish length and weight is important in order to properly manage aquaponic systems and to model stock trends. The dynamics of fish length distribution represent a key input for understanding the fish population dynamics and taking informed management decisions on exploited stocks. Nevertheless, in most fisheries the length of landed fish is still made by hand. As a result, length estimation is precise at a fish level, but due to the inherent high costs of manual sampling, the sample size tends to be small.

In [4], the authors present a scheme for fish length estimation using convolutional neural networks (CNNs). Their scheme is 93% accurate. A comparison between various popular CNN topologies for estimating the mass of Pintado Real fingerlings is given in [23]. The estimation of fish length using videos can also be very useful for developing a stable SAS. In this context, the authors in [5] present a method for underwater fish detection using videos. The system uses a ResNet-50 CNN and is 95.47% accurate. To operate a stable SAS, the early diagnosis of fish diseases can also be very helpful. In case a fish is found to be unhealthy, it can be removed from the pond; otherwise, the whole ecosystem can get disturbed. The researchers in [6] propose a scheme that can differentiate red spots on a fish from the white ones with 94.44% accuracy. Deep learning can be used to predict and control various nutrients and chemicals (moving in and out of the system) as well. For example, the authors in [24] present a scheme that can predict the concentration of oxygen in SASs. If a system contains less than the optimal level of oxygen, it can be supplied with more oxygen. Neural networks can be used to monitor other SAS parameters also, such as pH level, ammonia level, and temperature. They can also be used to make appropriate feeding decisions for plants and/or fishes [7].

Let us now discuss spiking neural networks. SNNs are the *next-generation* NNs whose potential has been demonstrated for a variety of applications, such as low-power image classification [8]. However, there are some major problems facing SNNs, especially in the context of SAS design.

- Firstly, no SAS-specific SNN system is available in the literature. All the smart aquaponic systems presented in the literature use artificial neural networks for parameter prediction and other tasks.
- Secondly, most SNN systems presented in the literature yield very low accuracy, even for digit classification tasks. Only a few SNN schemes achieve high accuracy. This is because spikes are non-differentiable in nature and direct backpropagation is quite

tricky to apply [15]. The non-differentiable nature of spikes is shown in Figure 5. Therefore, most researchers typically use an unsupervised algorithm *spike-timing-dependent plasticity* (STDP) for SNN training.

- Many backpropagation-based SNN training schemes [16,18,19] use complex mechanisms such as weight-threshold balancing.
- SNNs, unlike ANNs, require multiple time steps to process an input. This is why SNNs, sometimes, can consume a lot of time and hardware energy.

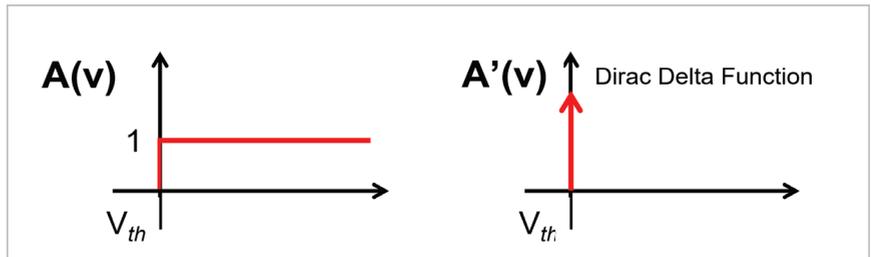


Figure 5. Spiking activation function and its non-differentiable derivative.

For example, the SNN scheme in [25] achieves 95% accuracy on MNIST dataset, even with more than five million synapses and hundreds of time steps. This is why it is crucial to develop SNN learning schemes that can yield high accuracy with a minimal network size. For instance, the work in [17] uses SNN backpropagation to achieve 98.7% accuracy with only 268,800 synapses and 8 time steps. The work in [17] is one of the rare works that do not require a long time to train SNNs. The other example is the work in [9] that requires only 10 time steps and a little over 150k synapses to reach 97.5% on MNIST. Most works, especially the ones based on STDP, require hundreds of time steps and millions of synapses to achieve reasonable accuracy, even on MNIST. The work in [26] is a good example of this. The scheme requires 700 time steps to achieve 92.63% accuracy, which is extremely low. Even backpropagation can be difficult to apply on SNNs. A good example is [27], where a 38-layer network with a complex batch form of normalization is applied to train the SNN. Though the network achieves good accuracy—equal to 92.8%—on CIFAR-10, the scheme is extremely complex to apply. Other such examples can be seen in [28–30].

The system in [31] achieves only 97.20% accuracy on MNIST in 1200 epochs, where each epoch contains tens of time steps. The scheme in [32] requires multiple time steps and 100 s of iterations to achieve 98.6% accuracy. To do this, multiple convolutional layers are followed by spiking layers equipped with backpropagation and STDP to achieve better accuracy. Though the scheme achieves good accuracy, the overall system is extremely complex and unsuitable for hardware implementations. Moreover, the scheme has not been tested on any aquaponics data.

1.2. Neuromorphic Accelerators (NMAs)

As mentioned earlier, though a lot of NMAs have been presented in the literature, none of them have been specifically designed for aquaponics. Moreover, the available NMAs or the schemes they use have been designed keeping only the inference in mind. Only a few hardware systems can train SNNs [9]. Moreover, systems capable of online learning are quite complex and do not yield high accuracy.

The authors in [33], for example, present a hardware neural system that can detect epileptic seizures with 95.14% accuracy. They implement fully parallel sigmoidal neurons using look-up tables on an FPGA. A generic hardware NN system that has been tested on a digit classification dataset is presented in [22]. The system in [34] has been tested on a cancer detection dataset. The system can predict the type of cancer with 98.23% accuracy.

Again, there is no system that can perform inference in a single time step. Almost all systems require tens of time steps for inference. Systems that require fewer than 10 time

steps are quite rare. The system presented in [17] is one such example; the system achieves convergence in eight time steps. The basic problem with [17] is that it has been designed specifically for MNIST dataset and has not been applied to aquaponics. The system in [35] uses 100 s of time steps to achieve 89% accuracy on a binary pattern recognition dataset having six samples. Since the dataset is extremely small, the efficiency of the scheme cannot be trusted. Moreover, the efficiency of the scheme has not been evaluated on any aquaponics data.

The system in [36] uses linear STDP to simplify the proposed hardware structure. Multiple subnetworks are used instead of a grand network to achieve better accuracy. The system in [37] presents a low-power SNN system capable of online learning. The system is better than many contemporary systems but achieves only 85% accuracy on MNIST. Though the systems in [36,37] are better than many other systems in terms of hardware efficiency and accuracy, their efficiency has not been demonstrated for aquaponics. The work in [38] presents a system that is capable of training SNNs by itself. The system uses the Tempotron learning rule for SNN training. However, the Tempotron algorithm is not suitable for multiclass classification and may not be suitable for aquaponics where multiple output classes are present. The system in [39] is another example of the online SNN learning system. It uses the Tempotron learning rule for 3-class classification and uses look-up tables for post-synaptic potential (PSP) kernel computations. It uses the system for pixel classification that takes around 100 time steps.

Other examples of systems requiring tens of time steps can be seen in [40–43]. Based on all these examples, we may safely conclude that no dedicated hardware system deals with smart aquaponics. Moreover, almost all systems require tens of time steps to achieve decent accuracy and are quite costly.

Problem Definition

Fish size estimation is an important part of smart aquaponics. Such systems are generally deployed using general-purpose computers that use artificial neural networks. Since GPCs are extremely slow and ANNs are quite costly, most SASs are expensive and slow. Though spiking neural networks are more hardware-efficient than ANNs, their training process is quite complex, tricky, and time consuming. Moreover, most online learning engines based on SNNs yield low accuracy.

Therefore, the first goal is to develop a hardware-efficient SNN learning scheme for smart aquaponics. The second goal is to design a hardware engine that can support the SNN-based ‘intelligent’ part of the SAS. The proposed system can estimate the fish weight category with 99.67% accuracy and the length category with 98.03% accuracy. The hardware engine is much cheaper than most contemporary designs. This intelligent module can automate the nutrient control process, which will in turn help in maintaining a stable SAS.

2. Materials and Methods

We develop and synthesize a neural network for estimating fish length and weight. The fish size estimation results can then be sent to a controller that makes feeding decisions. The proposed network structure is shown in Figure 6. The network has 4 layers: one input layer, two hidden layers, and an output layer. The input layer has been normalized according to the procedure described in [34]. In Figure 6, the symbol N represents the number of input features, and H is a symbol for the H^{th} hidden neuron.

Forward NN Pass

The complete network has four layers: one for inputs, one for outputs, and two hidden layers for spiking computations. In this article, the i^{th} synaptic weight and input are denoted by W_i and X_i , respectively. The subscript j is for the j^{th} postsynaptic neuron. The letter C_{th} represents the threshold controller, and the letter V represents the postsynaptic voltage. The postsynaptic voltage at a neuron j for the time step t is represented in Equation (1) by V_j . In Equation (1), C_{th}^i acts as a bias for ANN layers and as a dynamic threshold controller

(DTC) for spiking layers. This dynamic threshold controller is learnable. The λ is the leak factor. The purpose of this leak is to allow a steady decay of the membrane potential with respect to time. This leak can enhance the level of accuracy if the noise in a system has high frequency components [44]. However, since the focus of this work is to perform all computations in a single time step, this *leak* term is ignored. This neuronal model is similar to the one in [17].

$$v_j[t] = \sum_i (W_i \cdot X_i) + C_{th}^j + \lambda v[t - 1] \tag{1}$$

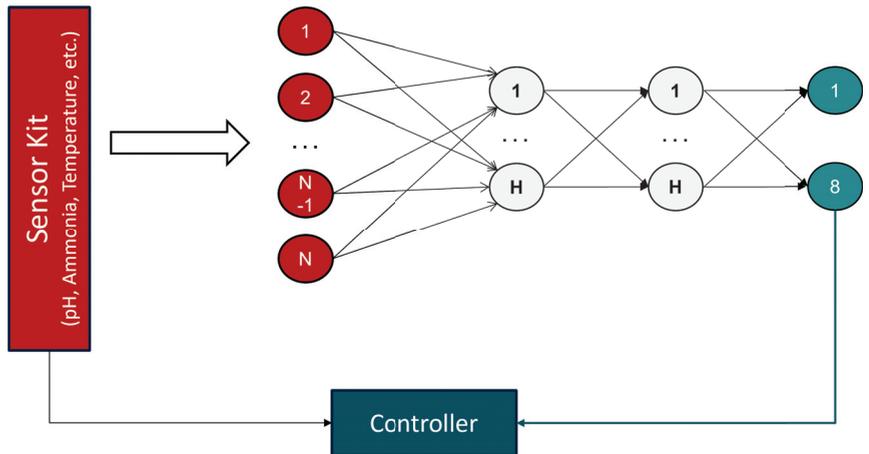


Figure 6. Structure of the proposed NN.

The inputs are normalized, as mentioned earlier. Unlike most other works on SNN such as [14,15,17], we do not convert input data into spikes. This is because the conversion to spikes is never lossless, and this can result in a severe degradation of accuracy [19]. Therefore, full-resolution inputs are applied to the network and then multiplied with corresponding synaptic efficacies, according to Equation (2). The learnable C_1 term is added as a neuronal offset for better learning. The final postsynaptic voltage is then compared against a threshold. If the given voltage is greater than this threshold, a spike is produced at the neuronal output; otherwise, nothing happens. This is described mathematically in Equation (3).

$$v_1 = \sum_i (W_i \cdot X_i) + C_{th}^1 \tag{2}$$

$$A_1 = \begin{cases} 1 & v_1 \geq V_{th} \\ 0 & v_1 < 0 \end{cases} \tag{3}$$

The Layer 1 activation vector is then passed as input to the Hidden Layer 2 in order to obtain the postsynaptic voltage V_2 , as represented in Equation (4).

$$v_2 = \sum_i (W_i \cdot A_{1i}) + C_{th}^2 \tag{4}$$

The Hidden Layer 1 spiking operation is repeated for the Hidden Layer 2 (HL2) as well. The spikes coming from the HL2 are applied to the output layer. However, the output layer is slightly different from the hidden layers. At the output layer, we do not perform any thresholding and use full-resolution outputs for prediction/classification. Again, the purpose is to preserve data integrity in order to produce accurate results. The thresholding operation can greatly reduce the network performance. As shown in the coming sections in detail, the full-resolution outputs do not reduce hardware efficiency at all. In fact, it can be slightly more hardware-efficient than the case where the spiking operation has to

be performed in hardware since the spiking operation requires an additional comparator. At the output layer, the activated values are the same as the incoming voltage values. Therefore, the neuronal voltages are compared directly. The neuron with the maximum voltage corresponds to the predicted class, as shown in Equation (5).

$$A_3 = \sum_i (W_i \cdot A_{2i}) + C_{th}^3 \tag{5}$$

Backward NN Pass

As mentioned in the previous sections, the gradient of the spiking function is the *direct delta* function that cannot be backpropagated. In order to resolve this issue, scientists have come up with the so-called surrogate gradients that are used as a replacement of the true gradient [15,17]. Some of the famous surrogate gradients are the sigmoidal gradient, the rectangular function, the polynomial function, and the Gaussian function. Among all these functions, the rectangular function is the most hardware-efficient. However, this sometimes results in poor accuracy, as shown in Section 3. The sigmoid gradient is quite smooth; it is given in Equation (6).

$$d(A_j) = \frac{1}{1 + e^{-v_j}} \cdot \left(1 - \frac{1}{1 + e^{-v_j}}\right) \tag{6}$$

Though the sigmoid derivative works pretty well as a surrogate gradient, it is difficult to implement in hardware since it is quite complex, as shown in Table 1. Therefore, various researchers have come up with functions that yield the same performance as the original sigmoid function while being hardware-efficient. One of the best examples that we could find in the literature is ‘Zhang–Sigmoid’ (ZS), given in [45]. The derivative of the Zhang–Sigmoid in [45] has a mean-squared error (MSD) of only 0.013% with the original sigmoidal derivative. The ZS derivative is compared with the original derivative in visual form in Figure 7. The mathematical expression is given in Equation (7).

$$d(A_j) = \begin{cases} \frac{4 + v_j}{16} & -4 \leq v_j < 0 \\ \frac{4 - v_j}{16} & 0 \leq v_j < 4 \\ 0 & otherwise \end{cases} \tag{7}$$

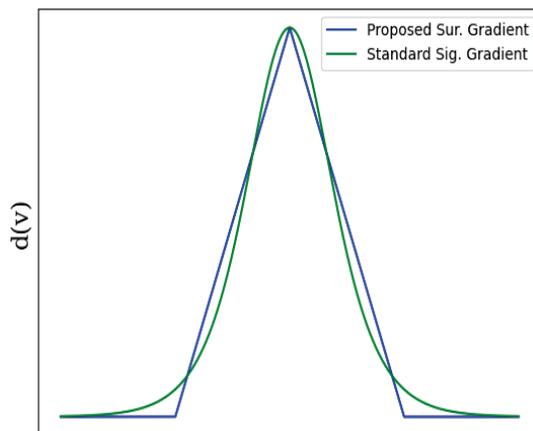


Figure 7. A comparison between original Sigmoid derivative and ZS derivative.

Proposed Surrogate Gradient

Though the original Zhang derivative/gradient apparently seems nearly perfect, a basic problem surrounding it is the lack of flexibility. The original ZS derivative cannot move and can work only when the spiking threshold voltage (V_{th}) is equal to zero, i.e., when $V_{th} = 0$. Therefore, we proposed a modification of the original derivative so that it can work with any threshold. The proposed, generalized form of ZS derivative is given in Equation (8). By manipulating the coefficients a and V_{th} , it is possible to train an SNN for any threshold value. This flexibility can yield great results since the zero threshold may not work for all datasets. The choice of threshold and other such parameters can greatly affect accuracy. The value of a can be adjusted in a way that it can be implemented without using any multiplier. The impact of the changing threshold on the shape of the proposed derivative is visually represented in Figure 8. Moreover, a comparison between the proposed surrogate gradient and other modern gradients (using the number of operations as a metric) is given in Table 1. In Table 1, μ represents the Gaussian mean, σ represents the standard deviation, and fxd is a short form for ‘fixed’.

$$d(A_j) = \begin{cases} \frac{a(4 + v_j - V_{th})}{16} & \frac{-4}{a} + V_{th} \leq v_j < V_{th} \\ \frac{a(4 - v_j - V_{th})}{16} & V_{th} \leq v_j < \frac{4}{a} + V_{th} \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

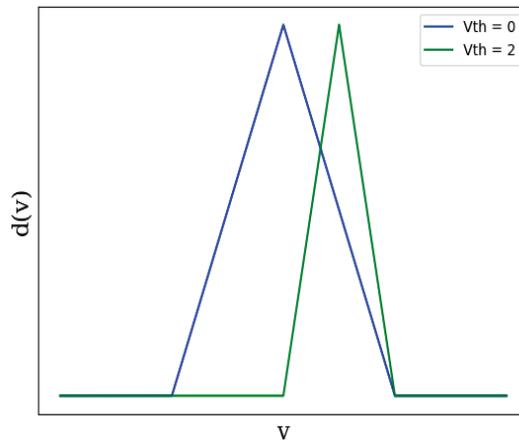


Figure 8. Impact of changing threshold on the shape of the proposed surrogate gradient.

Table 1. Hardware efficiency comparison between various SG implementations.

Parameters	#Adds.	#Mults.	#Div.	#Exp.	#Cmp.	Flexibility	Sig-MSE	
Rectangular	0	0	0	0	1	None	High (cond.)	
Sigmoid'	2	1	1	1	0	None	0	
Gaussian	$\mu = 0, \sigma = 1$	0	2	0	1	None	N.A.	
LogSQNL' [46]		2	0	0	0	1	None	0.41%
Zhang Sigmoid'	$a = 1, V_{th} = fxd$	2	0	0	0	1	None	0.013%
Proposed	General	3	0	0	0	1	Absolute	$\leq 0.013\%$

Experimental and Mathematical Proofs

As discussed and proven in this segment, the proposed Spikoponic derivative (given in Equation (8)) is valid for SNN backpropagation.

- **Experimental Proof:** The proposed Spikoponic derivative is used for backpropagation to train a network that classifies fish on the basis of their weight and length. Extensive experiments have been carried out using the proposed Spikoponic derivative. The details of the dataset are given in Section 3.1; the parameter values are given in Table 2. The results are given in Section 3.2. As shown in results, the proposed derivative works perfectly and can train an SNN for fish size estimation.
- **Mathematical Proof:** In order to perform backpropagation, the activation function must have a finite derivative [9,15,47]. The proposed spikoponic derivative, given in Equation (8), is *finite*. The derivative holds valid values since it is not always equal to zero or infinity.

Moreover, if the parameter a (in the Spikoponic derivative function) is equal to ∞ , the derivative converges to the dirac delta function, shown in Figure 5. This behavior clearly shows that the Spikoponic derivative is a valid function for the backward pass if step function is used in the forward pass. The mathematical expression for this behavior is given in Equation (9).

$$d(A_j) = \begin{cases} \infty & v_j = V_{th} \\ 0 & otherwise \end{cases} \tag{9}$$

The weights and DTCs are updated according to gradient descent rules, where network layers are iteratively updated based on an error function. Though all these processes are integrated into modern Python packages and we do not have to code everything in detail, we give a brief overview just to enhance readers’ understanding. The two basic parameter update rules are given in (10) and (11).

$$W^{(l)+} \leftarrow W^l - \eta \frac{\partial L}{\partial W^l} \tag{10}$$

$$C_{th}^{(l)+} \leftarrow C_{th}^l - \eta \frac{\partial L}{\partial C_{th}^l} \tag{11}$$

In the above equations, W^l represents ‘weight vector’ and C_{th}^l represents the DTC vector at layer l . Here, η represents the learning rate, the parameter that determines the speed at which the network updates weights in a training iteration. The term $\frac{\partial L}{\partial W^l}$ describes the changes in loss function with respect to weights at layer l . Both these terms are calculated using the chain rule, as in [9,15,17].

Since there are multiple layers in the proposed network, it would be unnecessary to derive mathematical expressions for all the layers. Therefore, we derive expressions only for one layer as a reference, just to give an idea of how the system works. Expressions for other layers can be derived using the same principle.

We mathematically establish the dependence of loss functions (L) on Layer 2 synaptic strengths in Equation (12), and on Layer 2 DTC (C_{th}^2) in Equation (13). To make the analysis understandable, the mean squared error (MSE) function has been used for reference. In the following equations, A_3 is the obtained output value at Layer 3, and y is the label voltage. The Spikoponic derivative function is already given in Equation (8). In order to keep mathematics simple, we do not incorporate terms associated with the optimization methods such as ADAM [48] in the presented mathematical expressions. Equations (12) and (13) do not incorporate the temporal dependence of the network

parameters and have been derived for one time step, which is one of the main goals of this work. To ignore temporal dependence, we make λ equal to zero.

$$\frac{\partial L}{\partial W_2} = \begin{cases} \frac{A_1 W_3 (A_3 - y) (a(4 + v_2 - V_{th}))}{16} & \frac{-4}{a} + V_{th} \leq v_2 < V_{th} \\ \frac{A_1 W_3 (A_3 - y) (a(4 - v_2 - V_{th}))}{16} & V_{th} \leq v_2 < \frac{4}{a} + V_{th} \\ 0 & otherwise \end{cases} \quad (12)$$

$$\frac{\partial L}{\partial C_{th}^2} = \begin{cases} \frac{C_{th}^2 W_3 (A_3 - y) (a(4 + v_2 - V_{th}))}{16} & \frac{-4}{a} + V_{th} \leq v_2 < V_{th} \\ \frac{C_{th}^2 W_3 (A_3 - y) (a(4 - v_2 - V_{th}))}{16} & V_{th} \leq v_2 < \frac{4}{a} + V_{th} \\ 0 & otherwise \end{cases} \quad (13)$$

Table 2. Hyper-parameter values and test conditions.

Parameter	Value
#TimeSteps	1
Learning Rate (η)	Default (0.001)
Batch Size	Default (32)
Optimizer	Adam
Loss Function	Cross Entropy
Leak (λ)	0
Output Coding	One Hot
Test Samples	20%
#Epochs	47

Proposed SpikoPoniC Hardware Engine

The proposed hardware system takes input from eight different sensors (systems), responsible for computing the following parameters: pH, temperature, date of creation, turbidity, dissolved oxygen, ammonia, nitrate, and population size. The system is fully parallel and can predict eight levels of fish weight and length (based on the input data) in a single clock cycle. The system consists of an input layer, two event-driven SNN layers, and an output layer containing a predictor. There are six different memory buffers, two for each layer. The top level diagram of the complete system is shown in Figure 9. The details of each of the components are mentioned in the coming subsections.

2.1. Switching Buffers

Since the system can estimate both length and weight categories, there are two types of memory buffers available for every network layer: one for storing synaptic efficacies corresponding to fish weight and the other one for storing efficacies corresponding to length. Which memory buffers are to be activated depends on the user. If length is to be predicted, the length memory buffers (LMBs) are enabled, weight memory buffers (WMBs) are disabled, and vice versa. Upon selection, all the synaptic efficacies are fetched in parallel to obtain high throughput.

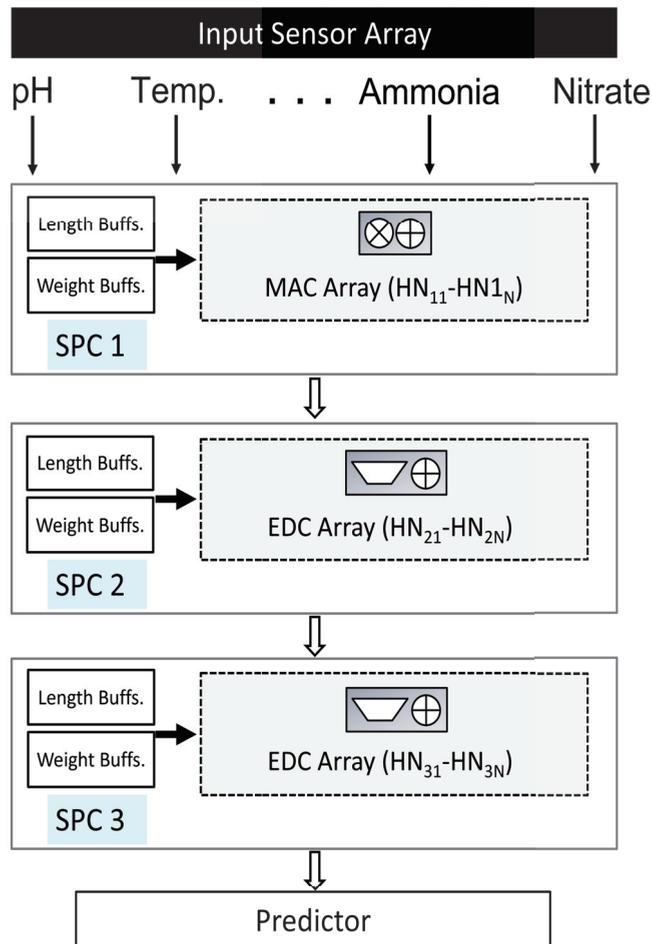


Figure 9. SpikoPoniC: top-level view.

2.2. Event-Driven Spiking Computers (EDCs)—Hidden Layer 1

As mentioned earlier, the first layer receives normalized inputs from sensors. These inputs are multiplied by the corresponding synaptic efficacies (SEs) in order to obtain *voltage*. The multiplications can be carried out using DSP48 elements on the FPGA. The voltage is obtained using a pipelined adder tree; the purpose of pipelining is to increase throughput. The first layer synaptic operations are, therefore, typical multiply accumulate (MAC) operations.

If the voltage is greater than a pre-defined threshold, a 1-bit spike is emitted; otherwise, nothing is passed. This is done using a comparator (CMP). Finally, these 1-bit values are stored in a pipeline register in order to increase throughput. Since spikes are quite small in size (1 bit), the pipeline register has a very small footprint. The spikes are then sent to the subsequent layer EDCs. The structure of a Layer 1 EDC is shown in Figure 10.

2.3. EDCs—Hidden Layer 2

The spikes coming from the HL1 are provided as input to the HL2. The HL2 actuators are also event-driven in nature, i.e., they are activated only when there is a valid spike coming from the previous layer. If there is no spike, no processing takes place by the processing elements and *zero voltage* is passed on to the subsequent unit. If there is a

spike, the respective synaptic efficacy gets added up to other valid synaptic efficacies by a pipelined adder tree.

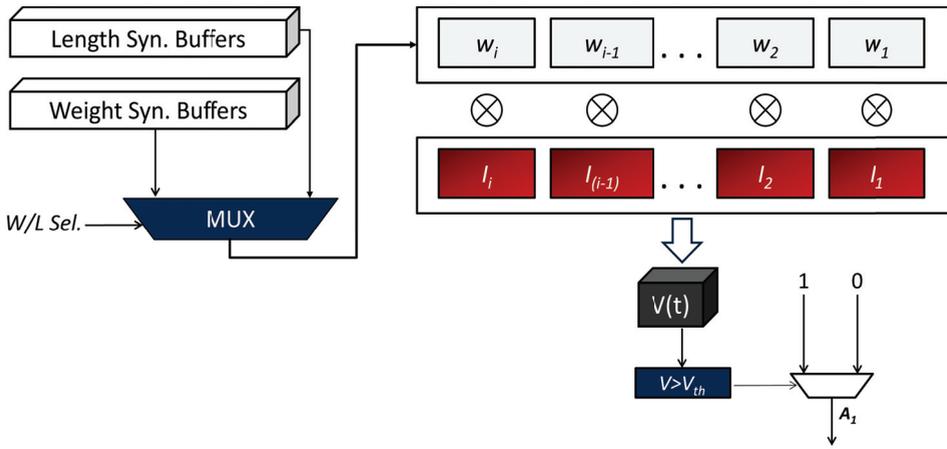


Figure 10. Internal Structure of an HL1 EDC.

The structure of the HL2 EDCs is shown in Figure 11. Like HL1, both the length and weight buffers are present. Once all the valid synaptic efficacies have been added up by the voltage computer (adder tree), a comparison operation is performed on the final voltage. If the voltage is greater than a predefined threshold, a spike is produced; otherwise, nothing happens. These spikes are then stored in the HL2 spike register that acts like a pipeline register. The outputs emanating from the spike register are then applied as input to the output layer.

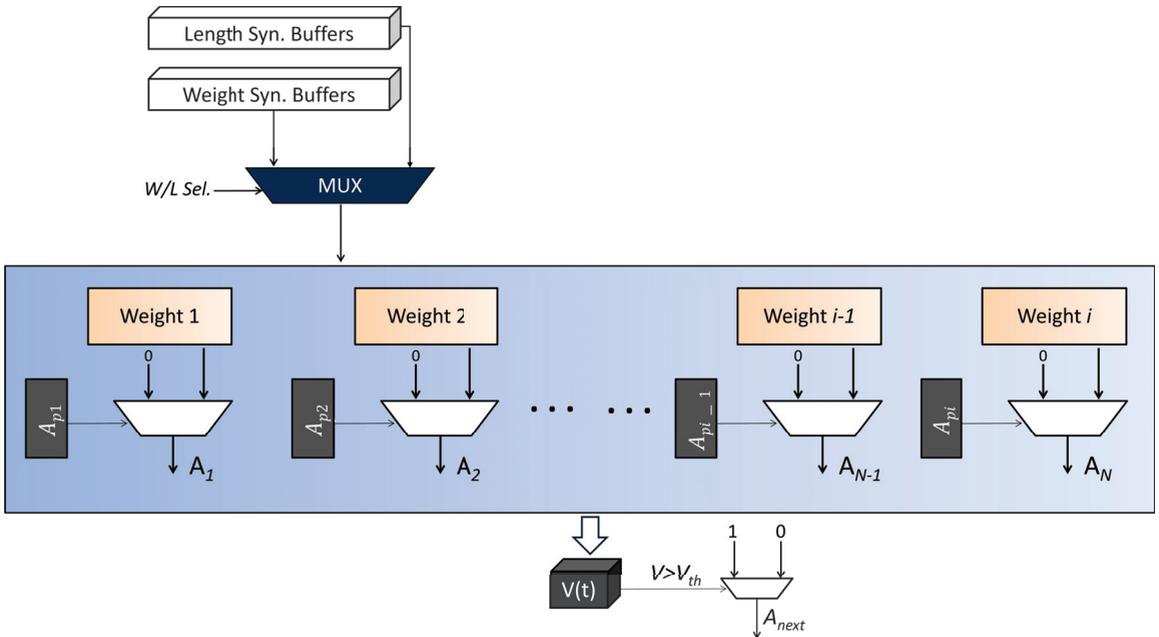


Figure 11. Internal structure of the HL2 EDCs.

2.4. Output-Layer EDCs

The structure of output-layer NACs is shown in Figure 12. Since softmax is a costly function and is used only when cross entropy loss is to be visualized, we use *logits* to perform prediction at the output [22]. The maximum voltage neuron corresponds to the predicted class. No spikes are used at the output for two reasons: the use of spikes results in a degradation of accuracy; an extra comparator is required. Therefore, the predictor directly compares logits and selects the one with the maximum value.

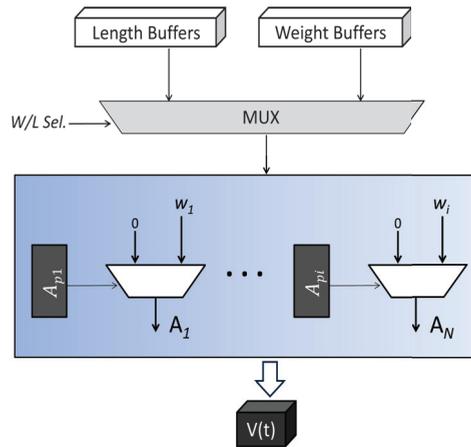


Figure 12. Internal structure of the output-layer EDCs.

3. Results and Discussion

This section presents the algorithmic and hardware efficiency of SpikoPoniC. The SpikoPoniC is compared with both software- and hardware-based works under the predefined test conditions. The FPGA-based SpikoPoniC is first compared with the CPU based aquaponic neural system in terms of speed for the same level of accuracy and then with other FPGA-based works in terms of cost and throughput.

3.1. Benchmarks and Test Conditions

The hardware inference system SikoPoniC—implemented on Virtex 6 (xc6vlx75t-ff784)—uses 7-bit synaptic efficacies and dynamic threshold controllers. The training has been performed on a GPC for the following NN topology: eight inputs; two hidden layers, each having 16 neurons; and eight output neurons. Python has been used for algorithmic evaluations, and Verilog has been used for hardware modeling. Since the platform is Virtex 6, the maximum temperature at which the device can operate is 85 degrees celsius [49]. Each 64 bit-wide block RAM can generate, store, and utilize eight additional Hamming-code bits and perform single-bit error correction and double-bit error detection (ECC) during the read process. As a special option, the bitstream can be AES-encrypted to prevent unauthorized copying of the design [49].

The dataset used is available from [50]. Since the directory contains a lot of files, only one file, which contains 175,000 samples, is sufficient for reliable performance evaluation. About 150k samples are used for training, and 25k are used for testing. There are 32 samples in a batch. The learning rate is 0.001, by default. The scheme uses the ADAM optimizer [48]. Backpropagation is performed using the cross entropy loss function. The system converges in about 47 epochs. Table 2 presents the test conditions and hyper-parameter values used for evaluating SpikoPoniC. For hyper-parameter tuning, we use *grid search* [51].

3.2. Algorithmic Efficiency Evaluation and Comparisons

The amount of CPU time that a training epoch consumes in the proposed scheme is shown in Figure 13. The average time per epoch is around 5 s. Once the network has been trained, the CPU uses approximately 1 s to infer all ($\approx 25,000$) the test samples.

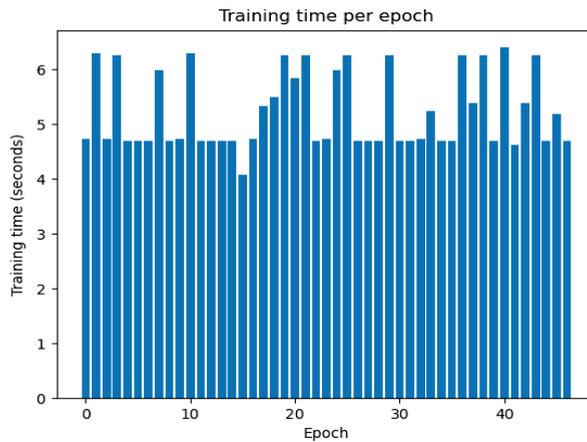


Figure 13. Per-epoch CPU processing time.

The proposed system can estimate the range of fish weight with 99.67% accuracy and the range of fish length with 98.03% accuracy. The average fish size estimation accuracy, therefore, is 98.85% accurate. The accuracy as a function of the number of epochs is shown in Figure 16; this figure is shown in the later part of this subsection along with all the necessary details. A comparison of SpikoPoniC with various modern DL-based aquaponic monitoring schemes in terms of accuracy is given in Table 3.

For fish weight estimation, the maximum precision and recall values are 94.80% and 99.88%, respectively. For fish length estimation, the maximum precision and recall values are 88.39% and 99.30%, respectively. On average, an F1 score of 93.701% can easily be obtained when it comes to fish size (which includes both length and weight) estimation. It is not possible to compare our results with other works in terms of precision, recall, or F1 score. This is because many research works on aquaponics do not use these metrics and rely on accuracy for performance evaluation.

Table 3. Accuracy comparisons—smart aquaponic systems.

	Accuracy	Application
[6]	94.44%	Fish disease detection
[23]	67.08%	Fingerling weight estimation
[7]	95%	Feeding intensity estimation
[52]	96.50%	Plant detection
[53]	97.80%	Fish length estimation
[54]	92.60%	Plant detection
[54]	98.70%	Plant detection
[55]	87%	Fish size estimation
Prop.	99.67%	Fish weight estimation
Prop.	98.03%	Fish length estimation

A visual comparison in the form of bar graphs between various algorithms for fish weight and length estimation is given in Figure 14. The following algorithms are compared in Figure 14. All the algorithms are applied for just one time step.

- *Direct SNN Training (DST)* [15,17].
All the layers use spikes in the forward pass and surrogate gradient (sigmoidal gradient [15]) in the backward pass. The network achieves very low accuracy since there is only one time step for which we have to train the network.
- *ReLU-SNN Conversion (ReLU-SNN)* [19].
The network is trained using ReLU function, and the trained network is then converted into an SNN. No weight-threshold normalization is applied since the purpose is to devise an algorithm that is hardware-efficient if on-chip learning is required. Normalization processes can never be efficient for on-chip learning [9]. For better accuracy, the inputs are not converted into spikes since this results in a loss of accuracy [19].
- *Rectangular Straight-Through Estimator (Rect-STE)* [17].
The network uses spikes in the forward pass, and the rectangular-shaped surrogate gradient in the backward pass, as in [17]. For achieving high accuracy, full-resolution inputs are used and no conversion to spikes takes place.
- *Proposed Algorithm (Normalized Inputs, Spiking Outputs)*.
The proposed algorithm, as mentioned in Section 2, is applied with full-resolution inputs but spiking outputs.
- *Proposed Algorithm (Normalized Inputs, Full-Resolution Outputs)*.
The proposed algorithm, as described in Section 2, is applied with full-resolution inputs and outputs (logits).

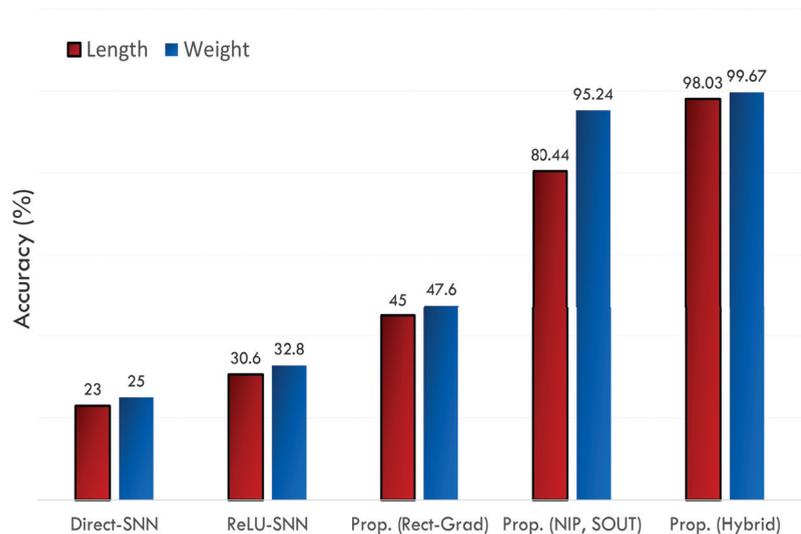


Figure 14. Accuracy comparison between various algorithms.

The results are presented in Figure 14. All schemes other than the proposed one yield poor results. Let us discuss the underlying causes. The DST, ReLU-SNN, and Rect-STE are strongly dependent on a large time period since they all use spikes at both the input and the output layers. Though the use of spikes at the input layer is quite hardware-efficient, it results in a degradation of accuracy due to the loss incurred in the conversion process. As a result, good accuracy cannot be obtained in a single time step. The ReLU-SNN scheme depends strongly on weight-threshold balancing too [18]. For the proposed scheme, the use of spikes at the output layer results in a lower accuracy than the case where logits (voltage as such) are used at the output layer. The simple reason is that the conversion of voltage into 1-bit spikes even at the output layer reduces the dynamic range and precision, resulting in visible loss of accuracy. Moreover, as explained in the coming sections, the use

of raw voltage instead of converted spikes at the output layer does not decrease hardware efficiency at all.

As mentioned earlier, the use of spikes at both the input and output layers for a single time step does not produce good results. This is evident even from Figure 15a, which shows the evolution of training accuracy with the number of training epochs: the fluctuation is very high and there is no such thing as convergence. Similarly, the use of rectangular, flat surrogate gradient results in a very poor accuracy. In fact, no convergence is achieved since the gradient is completely flat and can result in gradient vanishing/explosion. The rectangular gradient, on the other hand, does change its shape around the *threshold* and results in better learning.

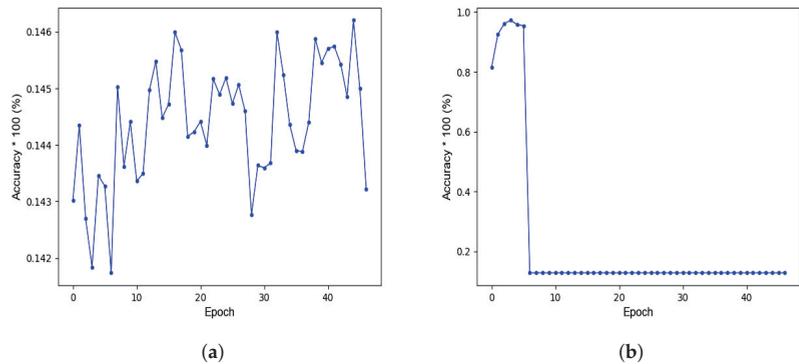


Figure 15. Accuracy results for some hardware-efficient schemes. (a) Fully spiking network. (b) Direct SNN training using the rectangular surrogate gradient.

The use of full-resolution inputs and outputs, instead of spikes, preserves the data content and produces excellent results. The results obtained using the proposed algorithm for fish length and weight estimation are shown in Figure 16c and Figure 16d, respectively. As shown in the figures, the network learning is quite smooth. However, if spikes are used even at the output layer, the accuracy degrades and network accuracy keeps fluctuating. This is shown in Figure 16a,b.

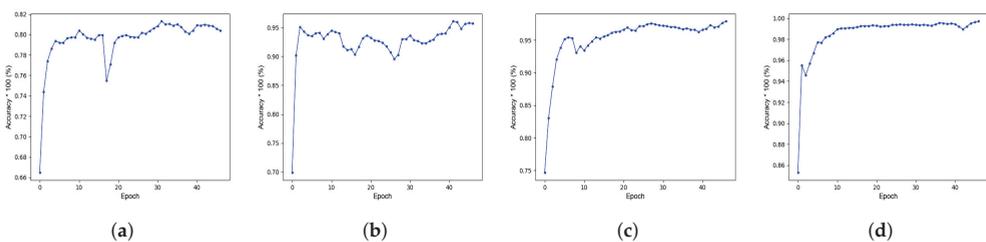


Figure 16. Accuracy as a function of epochs, obtained for various conditions and scenarios. (a) Normalized inputs (NIP), spiking outputs (SOUT) for fish length estimation. (b) NIP, SOUT for fish weight estimation (c) Proposed scheme for length estimation. (d) Proposed scheme for weight estimation.

3.3. Hardware Efficiency Evaluation and Comparisons

SpikoPoniC is fully parallel and can process a single sample in a single clock cycle. The maximum attainable clock frequency is 84.23 MHz to 117.33 MHz, depending on the type of implementation and hardware synthesizer settings, as will be discussed later. SpikoPoniC can process more than 84 million samples in a second. A comparison between the FPGA and CPU implementations is given in Figure 17, which shows that the FPGA system

is at least 3369 times faster than the CPU implementation. In Figure 17, the throughput is given in terms of *millions of samples per second* (MSPS). This shows the potential of neuromorphic systems and provides a great incentive to SAS developers. For the SpikoPoniC, the results have been obtained with and without using DSP48 elements. At the given frequencies, the DSP implementation consumes about 1.975 watts, of which the leakage power is around 1.577 watts; the non-DSP implementation consumes about 2.302 watts, of which the leakage power is around 1.582 watts.

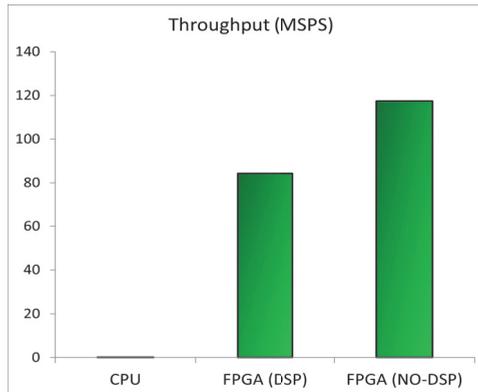


Figure 17. Hardware efficiency comparison between the CPU and the FPGA implementation.

A comparison between the fully ANN implementation and the proposed SNN (hybrid) implementation for the same test topology (30-5-5-2) under the same test conditions is shown in Figure 18. As shown in Figure 18, the proposed implementation is far better than the ANN implementation in terms of latency (speed) and cost (amount of FPGA resources occupied). Compared with the proposed technique, the fully ANN implementation consumes approximately 27% more registers, 45% look-up tables, and 28% more time. The disparity will grow as the network becomes deeper and larger. The purpose of this figure is to give a glimpse into the disparity between the proposed implementation and the fully ANN implementation. The network accuracy, however, remains (more or less) the same. For example, in the finalized network, discussed in Table 4, the proposed scheme incurs less than 1% loss, compared to the actual ANN scheme using rectified linear units (ReLU) at all the layers in the context of fish length estimation. For fish weight estimation, there is no difference at all between the proposed scheme and the ANN-based scheme. There is a reason why we chose 30-5-5-2 topology for ANN-SNN hardware efficiency comparison: the available hardware platform is a low-end model of Virtex 6 (xc6vlx75t-ff784) which has approximately 93,000 slice registers and about 46,000 look-up tables only. A large, fully parallel ANN design cannot fit on such a small platform, given the limited amount of resources. The ANN implementation for a large topology, say 8-16-16-8, is unable to fit on the given platform, though an SNN implementation for the same topology is possible. This simply proves that the SNN implementation is more efficient on hardware than the ANN one.

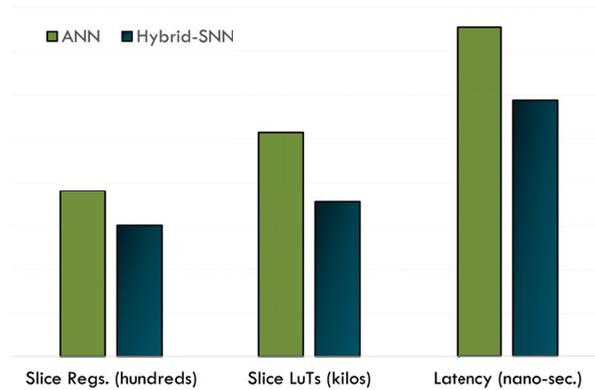


Figure 18. Speed and cost disparity between the traditional ANN and the proposed SNN implementation.

Figure 19 shows the number of adders and multipliers required by the proposed implementation. The cost and throughput comparisons are given in Table 4.

# MACs	: 112
11x7-to-21-bit MAC	: 112
# Multipliers	: 16
11x7-bit registered multiplier	: 16
# Adders/Subtractors	: 40
12-bit adder	: 24
22-bit adder	: 16
# Adder Trees	: 120
11-bit / 4-inputs adder tree	: 24
9-bit / 4-inputs adder tree	: 96

Figure 19. Advanced hardware description language (HDL) synthesis.

For the SpikoPoniC, the results have been obtained with and without using DSP48 elements. In both the cases, SpikoPoniC shows superior performance. Though the systems presented in Table 4 are quite hardware-efficient, they suffer from one or both of the following problems: the systems are not made specifically for aquaponics, or the systems are quite costly because they are based on ANNs. For example, the system in [33] has been developed for diagnosing epilepsy. Moreover, the system uses parallel sigmoidal neurons, which are quite costly to be used on hardware. The system predicts epilepsy with 95.14% accuracy. The hardware system [34] uses hardware-aware sigmoidal and swish neurons to predict cancer with high accuracy. The system in [56] implements a network with radial basis functions just to demonstrate the efficiency of slope-based Gaussian approximation; no dataset is used to evaluate performance. The system in [22] uses ReLU at all the network layers to improve classification accuracy. However, it uses extremely small weights and biases, which might not be enough for obtaining a reasonable accuracy on aquaponics data. Furthermore, the system does not use any aquaponics data for system evaluation. Therefore, how it will perform on aquaponics data remains dubious.

Though the inference engine in [9] achieves a very high throughput and is cost effective, it uses spikes at all the layers. Therefore, it cannot be used for the SAS under consideration, as mentioned earlier. The data under consideration requires full resolution inputs and outputs; otherwise, it produces a very low accuracy. The work in [35] uses a small (toy) dataset with 25 binary input pixels and one neuron for binary (X and O) classification; two samples are used for training. The authors do not mention the system throughput

explicitly. However, it is safe to assume that the maximum TP is far less than 1.9×10^6 samples per second. This is because the maximum operating frequency of the system is around 189 MHz, and the time period requires one to compute a sample is 100 ms. Moreover, the discretization step is 0.001. There is another metric used to compare these works: processing time per second (PTPS), represented in microseconds (μ s). PTPS is the amount of time required to infer/process a given input sample. The smaller the PTPS value, the faster the system.

Table 4. Hardware cost and throughput comparisons.

System	Application	Topology	Accur.	Regs.	LuTs	DSPs	Platform	TP ($\times 10^6$)	PTPS (μ s)
[33]	Epilepsy Det.	5-12-3	95.14%	114	12,960	116	Cyclone IV	50	0.02
[34]	Cancer Det.	30-5-2	98.23%	983	2654	234	Virtex 6	63.5	0.0157
[22]	Digit Class.	64-20-10	94.28%	4677	30,654	0	Virtex 6	93.2	0.0107
[35]	Bin. Class.	25-5-1	89%	1023	11,339	-	Virtex 6	$< < 1.89$	> 0.53
[57]	None	5-5-2	-	1898	3124	154	Virtex 5	-	-
[56]	None	-	-	790	1195	14	Spartan 3	10	0.1
Prop._{DSP}	Aquaponics	8-16-16-8	98.85%	1091	3749	128	Virtex 6	84.23	0.012
Prop._{NO-DSP}	Aquaponics	8-16-16-8	98.85%	4259	18,283	0	Virtex 6	117.33	0.008

4. Conclusions

This article presents a novel training methodology to train and implement a spiking neural neural network on a neuromorphic system for smart aquaponics. The article also presents a novel surrogate gradient for SNN training that promises both flexibility and hardware efficiency.

The purpose of developing this system is the low-cost, real-time estimation of fish size, which in turn will help devise efficient smart aquaponic systems and make appropriate feeding decisions. The hardware inference engine is capable of classifying more than 84 million samples in a second. The system is trained using 150,000 samples and can predict fish length with 98.03% accuracy and fish weight with 99.67% accuracy. The design occupies only 1100 slice registers and 3749 look-up tables. The engine is about 3369 times faster than a typical GPC and is far less costly than the implementation that uses an ANN for the same job. The ANN implementation is 45% costlier than the proposed implementation with less than 1% degradation in accuracy. The system consumes about XYZ milliwatts at the maximum possible frequency, i.e., 84.23 MHz. All these results have been obtained on a low-end Virtex 6 FPGA. The maximum temperature at which the device can operate is 85 degrees celsius.

Therefore, it may safely be concluded that the proposed scheme is suitable for DL-based smart aquaponics. The scheme gives a roadmap on how to use SNNs for Aquaponics 4.0 industrial applications. In future, the same technique can be modified to be extended to spiking convolutional neural networks for better classification. Another possible improvement is to use a higher number of input features to obtain better results. A larger number of sensors can be used to collect data for application at the input side.

Author Contributions: Conceptualization, A.S. and M.A.I.; methodology, A.S.; software, A.S. and J.S.; validation, A.S. and J.S.; formal analysis, A.S.; investigation, A.S.; resources, M.A.I.; writing—original draft preparation, A.S.; writing—review and editing, A.S., M.I.V. and M.A.I.; visualization, A.S. and M.A.I.; supervision, M.I.V., M.A.I. and S.H.P.; project administration, M.I.V., M.A.I. and K.J.H.; and funding acquisition, K.J.H. and M.I.V. All authors have read and agreed to the published version of the manuscript.

Funding: The authors would like to acknowledge the financial support from the following organizations: Zuhai UM Science and Technology Research Institute, via the projects of the ZUMRI-Lingyange

Semiconductor Joint Lab (CP-031-2022); the Lingyang Semiconductor Incorporated, Zhuhai (CP-017-2022); and the Blue Ocean Smart System (Nanjing) Limited (CP-003-2023).

Institutional Review Board Statement: This study does not involve humans or animals. There are no ethical concerns regarding this study.

Data Availability Statement: All the datasets used for the evaluation of the proposed scheme are available publicly. The relevant references have already been mentioned in the bibliography.

Conflicts of Interest: The authors declare no competing interests.

Abbreviations

The following abbreviations are used in this manuscript:

ANN	Artificial neural network
ASIC	Application-specific integrated circuit
DL	Deep learning
FPGA	Field programmable gate array
GPC	General-purpose computer
SAS	Smart aquaponic system
SNN	Spiking neural network

References

- Calone, R.; Orsini, F. Aquaponics: A Promising Tool for Environmentally Friendly Farming. *Front. Young Minds* **2022**, *10*, 707801. [CrossRef]
- Taha, M.F.; ElMasry, G.; Gouda, M.; Zhou, L.; Liang, N.; Abdalla, A.; Rousseau, D.; Qiu, Z. Recent Advances of Smart Systems and Internet of Things (IoT) for Aquaponics Automation: A Comprehensive Overview. *Chemosensors* **2022**, *10*, 303. [CrossRef]
- Dhal, S.B.; Jungbluth, K.; Lin, R.; Sabahi, S.P.; Bagavathiannan, M.; Braga-Neto, U.; Kalafatis, S. A machine-learning-based IoT system for optimizing nutrient supply in commercial aquaponic operations. *Sensors* **2022**, *22*, 3510. [CrossRef]
- Lu, H.; Ma, X. Hybrid decision tree-based machine learning models for short-term water quality prediction. *Chemosphere* **2020**, *249*, 126169. [CrossRef]
- Jalal, A.; Salman, A.; Mian, A.; Shortis, M.; Shafait, F. Fish detection and species classification in underwater environments using deep learning with temporal information. *Ecol. Inform.* **2020**, *57*, 101088. [CrossRef]
- Hasan, N.; Ibrahim, S.; Aqilah Azlan, A. Fish diseases detection using convolutional neural network (CNN). *Int. J. Nonlinear Anal. Appl.* **2022**, *13*, 1977–1984.
- Ubina, N.; Cheng, S.C.; Chang, C.C.; Chen, H.Y. Evaluating fish feeding intensity in aquaculture with convolutional neural networks. *Aquac. Eng.* **2021**, *94*, 102178. [CrossRef]
- Merolla, P.A.; Arthur, J.V.; Alvarez-Icaza, R.; Cassidy, A.S.; Sawada, J.; Akopyan, F.; Jackson, B.L.; Imam, N.; Guo, C.; Nakamura, Y.; et al. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science* **2014**, *345*, 668–673. [CrossRef]
- Siddique, A.; Vai, M.I.; Pun, S.H. A low cost neuromorphic learning engine based on a high performance supervised SNN learning algorithm. *Sci. Rep.* **2023**, *13*, 6280. [CrossRef]
- Kim, S.; Park, S.; Na, B.; Yoon, S. Spiking-YOLO: Spiking neural network for energy-efficient object detection. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 11270–11277.
- Maass, W.; Papadimitriou, C.H.; Vempala, S.; Legenstein, R. Brain computation: A computer science perspective. *Comput. Softw. Sci.* **2019**, *10000*, 184–199.
- Izhikevich, E.M. Which model to use for cortical spiking neurons? *IEEE Trans. Neural Netw.* **2004**, *15*, 1063–1070. [CrossRef]
- Izhikevich, Eugene M. Simple model of spiking neurons. *IEEE Trans. Neural Netw.* **2003**, *14*, 1569–1572. [CrossRef]
- Han, J.; Li, Z.; Zheng, W.; Zhang, Y. Hardware implementation of spiking neural networks on FPGA. *Tsinghua Sci. Technol.* **2020**, *25*, 479–486. [CrossRef]
- Wu, Y.; Deng, L.; Li, G.; Zhu, J.; Shi, L. Spatio-temporal backpropagation for training high-performance spiking neural networks. *Front. Neurosci.* **2018**, *12*, 331. [CrossRef]
- Qiao, G.; Hu, S.; Chen, T.; Rong, L.; Ning, N.; Yu, Q.; Liu, Y. STBNN: Hardware-friendly spatio-temporal binary neural network with high pattern recognition accuracy. *Neurocomputing* **2020**, *409*, 351–360. [CrossRef]
- Yin, S.; Venkataramanaiah, S.K.; Chen, G.K.; Krishnamurthy, R.; Cao, Y.; Chakrabarti, C.; Seo, J. Algorithm and hardware design of discrete-time spiking neural networks based on back propagation with binary activations. In Proceedings of the 2017 IEEE Biomedical Circuits and Systems Conference (BioCAS), Torino, Italy, 19–21 October 2017; pp. 1–5. [CrossRef]
- Diehl, P.U.; Neil, D.; Binas, J.; Cook, M.; Liu, S.C.; Pfeiffer, M. Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing. In Proceedings of the 2015 International Joint Conference on Neural Networks (IJCNN), Killarney, Ireland, 12–17 July 2015; pp. 1–8.

19. Rueckauer, B.; Lungu, I.A.; Hu, Y.; Pfeiffer, M.; Liu, S.C. Conversion of continuous-valued deep networks to efficient event-driven networks for image classification. *Front. Neurosci.* **2017**, *11*, 682. [CrossRef]
20. Azghadi, M.R.; Lammie, C.; Eshraghian, J.K.; Payvand, M.; Donati, E.; Linares-Barranco, B.; Indiveri, G. Hardware implementation of deep network accelerators towards healthcare and biomedical applications. *IEEE Trans. Biomed. Circuits Syst.* **2020**, *14*, 1138–1159. [CrossRef]
21. Ortega-Zamorano, F.; Jerez, J.M.; Urda Muñoz, D.; Luque-Baena, R.M.; Franco, L. Efficient Implementation of the Backpropagation Algorithm in FPGAs and Microcontrollers. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**, *27*, 1840–1850. [CrossRef]
22. Siddique, A.; Iqbal, M.A.; Aleem, M.; Islam, M.A. A 218 GOPS neural network accelerator based on a novel cost-efficient surrogate gradient scheme for pattern classification. *Microprocess. Microsyst.* **2023**, *99*, 104831. [CrossRef]
23. Junior, A.d.S.O.; Sant’Ana, D.A.; Pache, M.C.B.; Garcia, V.; de Moares Weber, V.A.; Astolfi, G.; de Lima Weber, F.; Menezes, G.V.; Menezes, G.K.; Albuquerque, P.L.F.; et al. Fingerlings mass estimation: A comparison between deep and shallow learning algorithms. *Smart Agric. Technol.* **2021**, *1*, 100020. [CrossRef]
24. Ren, Q.; Zhang, L.; Wei, Y.; Li, D. A method for predicting dissolved oxygen in aquaculture water in an aquaponics system. *Comput. Electron. Agric.* **2018**, *151*, 384–391. [CrossRef]
25. Diehl, P.U.; Cook, M. Unsupervised learning of digit recognition using spike-timing-dependent plasticity. *Front. Comput. Neurosci.* **2015**, *9*, 99. [CrossRef] [PubMed]
26. Saunders, D.J.; Siegelmann, H.T.; Kozma, R.; et al. STDP learning of image patches with convolutional spiking neural networks. In Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1–7.
27. Vicente-Sola, A.; Manna, D.L.; Kirkland, P.; Di Caterina, G.; Bihl, T. Keys to accurate feature extraction using residual spiking neural networks. *Neuromorphic Comput. Eng.* **2022**, *2*, 044001. [CrossRef]
28. Deng, S.; Gu, S. Optimal conversion of conventional artificial neural networks to spiking neural networks. *arXiv* **2021**, arXiv:2103.00476.
29. Fang, W.; Yu, Z.; Chen, Y.; Huang, T.; Masquelier, T.; Tian, Y. Deep residual learning in spiking neural networks. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 21056–21069.
30. Zhang, W.; Li, P. Temporal spike sequence learning via backpropagation for deep spiking neural networks. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 12022–12033.
31. Tavanaei, A.; Maida, A. BP-STDP: Approximating backpropagation using spike timing dependent plasticity. *Neurocomputing* **2019**, *330*, 39–47. [CrossRef]
32. Tavanaei, A.; Kirby, Z.; Maida, A.S. Training spiking convnets by stdp and gradient descent. In Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1–8.
33. Sarić, R.; Jokić, D.; Beganović, N.; Pokvić, L.G.; Badnjević, A. FPGA-based real-time epileptic seizure classification using Artificial Neural Network. *Biomed. Signal Process. Control* **2020**, *62*, 102106. [CrossRef]
34. Siddique, A.; Iqbal, M.A.; Aleem, M.; Lin, J.C.W. A high-performance, hardware-based deep learning system for disease diagnosis. *PeerJ Comput. Sci.* **2022**, *8*, e1034. [CrossRef]
35. Farsa, E.Z.; Ahmadi, A.; Maleki, M.A.; Gholami, M.; Rad, H.N. A low-cost high-speed neuromorphic hardware based on spiking neural network. *IEEE Trans. Circuits Syst. II Express Briefs* **2019**, *66*, 1582–1586. [CrossRef]
36. Sun, C.; Sun, H.; Xu, J.; Han, J.; Wang, X.; Wang, X.; Chen, Q.; Fu, Y.; Li, L. An Energy Efficient STDP-Based SNN Architecture With On-Chip Learning. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2022**, *69*, 5147–5158. [CrossRef]
37. Li, S.; Zhang, Z.; Mao, R.; Xiao, J.; Chang, L.; Zhou, J. A Fast and Energy-Efficient SNN Processor With Adaptive Clock/Event-Driven Computation Scheme and Online Learning. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2021**, *68*, 1543–1552. [CrossRef]
38. Siddique, A.; Vai, M.I.; Pun, S.H. A low-cost, high-throughput neuromorphic computer for online SNN learning. *Clust. Comput.* **2023**, 1–18. [CrossRef]
39. Zhang, G.; Li, B.; Wu, J.; Wang, R.; Lan, Y.; Sun, L.; Lei, S.; Li, H.; Chen, Y. A low-cost and high-speed hardware implementation of spiking neural network. *Neurocomputing* **2020**, *382*, 106–115. [CrossRef]
40. Heidarpur, M.; Ahmadi, A.; Ahmadi, M.; Azghadi, M.R. CODIC-SNN: On-FPGA STDP learning with izhikevich neurons. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2019**, *66*, 2651–2661. [CrossRef]
41. Lammie, C.; Hamilton, T.; Azghadi, M.R. Unsupervised character recognition with a simplified FPGA neuromorphic system. In Proceedings of the 2018 IEEE International Symposium on Circuits and Systems (ISCAS), Florence, Italy, 27–30 May 2018; pp. 1–5.
42. Ma, D.; Shen, J.; Gu, Z.; Zhang, M.; Zhu, X.; Xu, X.; Xu, Q.; Shen, Y.; Pan, G. Darwin: A neuromorphic hardware co-processor based on spiking neural networks. *J. Syst. Archit.* **2017**, *77*, 43–51. [CrossRef]
43. Neil, D.; Liu, S.C. Minitaur, an event-driven FPGA-based spiking network accelerator. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2014**, *22*, 2621–2628. [CrossRef]
44. Chowdhury, S.S.; Lee, C.; Roy, K. Towards Understanding the Effect of Leak in Spiking Neural Networks. *arXiv* **2020**, arXiv:2006.08761.
45. Zhang, M.; Vassiliadis, S.; Delgado-Frias, J.G. Sigmoid generators for neural computing using piecewise approximations. *IEEE Trans. Comput.* **1996**, *45*, 1045–1049. [CrossRef]
46. Wuraola, A.; Patel, N.; Nguang, S.K. Efficient activation functions for embedded inference engines. *Neurocomputing* **2021**, *442*, 73–88. [CrossRef]

47. Esser, S.K.; Appuswamy, R.; Merolla, P.; Arthur, J.V.; Modha, D.S. Backpropagation for energy-efficient neuromorphic computing. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 1117–1125.
48. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
49. Xilinx. Virtex-6 Family Overview. Available online: <https://www.digikey.com/htmldatasheets/production/738648/0/0/1/virtex-6-fpga-familyoverview.html> (accessed on 9 October 2023).
50. Collins Udakor. Sensor Based Aquaponics Fish Pond Datasets. Available online: <https://www.kaggle.com/datasets/ogbuokiriblessing/sensor-based-aquaponics-fish-pond-datasets?resource=download> (accessed on 5 May 2023).
51. Zheng, A. Evaluating Machine Learning Models: A Beginner’s Guide to Key Concepts and Pitfalls. 2015. Available online: <https://www.oreilly.com/content/evaluating-machine-learning-models/> (accessed on 9 October 2023).
52. Taha, M.F.; Abdalla, A.; ElMasry, G.; Gouda, M.; Zhou, L.; Zhao, N.; Liang, N.; Niu, Z.; Hassanein, A.; Al-Rejaie, S.; et al. Using deep convolutional neural network for image-based diagnosis of nutrient deficiencies in plants grown in aquaponics. *Chemosensors* **2022**, *10*, 45. [CrossRef]
53. Monkman, G.G.; Hyder, K.; Kaiser, M.J.; Vidal, F.P. Using machine vision to estimate fish length from images using regional convolutional neural networks. *Methods Ecol. Evol.* **2019**, *10*, 2045–2056. [CrossRef]
54. Yadav, A.; Thakur, U.; Saxena, R.; Pal, V.; Bhateja, V.; Lin, J.C.W. AFD-Net: Apple Foliar Disease multi classification using deep learning on plant pathology dataset. *Plant Soil* **2022**, *477*, 595–611. [CrossRef]
55. Álvarez-Ellacuría, A.; Palmer, M.; Catalán, I.A.; Lisani, J.L. Image-based, unsupervised estimation of fish size from commercial landings using deep learning. *ICES J. Mar. Sci.* **2020**, *77*, 1330–1339. [CrossRef]
56. Shymkovich, V.; Telenyk, S.; Kravets, P. Hardware implementation of radial-basis neural networks with Gaussian activation functions on FPGA. *Neural Comput. Appl.* **2021**, *33*, 9467–9479. [CrossRef]
57. Tiwari, V.; Khare, N. Hardware implementation of neural network with Sigmoidal activation functions using CORDIC. *Microprocess. Microsyst.* **2015**, *39*, 373–381. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.



Article

MushR: A Smart, Automated, and Scalable Indoor Harvesting System for Gourmet Mushrooms

Anant Sujatanagarjuna [†], Shohreh Kia [†], Dominique Fabio Briechle [†] and Benjamin Leiding ^{*,†}

Institute for Software and Systems Engineering, Clausthal University of Technology, 38678 Clausthal-Zellerfeld, Germany; anant.sujatanagarjuna@tu-clausthal.de (A.S.); shohreh.kia@tu-clausthal.de (S.K.); dominique.fabio.briechle@tu-clausthal.de (D.F.B.)

* Correspondence: benjamin.leiding@tu-clausthal.de

[†] These authors contributed equally to this work.

Abstract: Gourmet mushrooms are foraged from the wild or grown indoors in controlled environments. Indoor mushroom farms with controlled growth environments allow for all-year-round growing. However, it remains a labor-intensive process. We propose MushR as a modular and scalable gourmet mushroom growing and harvesting system that goes beyond the state of the art, which merely monitors and controls the growing environment, by introducing an image recognition system that determines when and which mushrooms are ready to be harvested in conjunction with a proof of concept of an automated mushroom harvesting mechanism for harvesting the mushrooms without human interaction. The image recognition setup monitors the growing status of the mushrooms and guides the harvesting process. We present a Mask R-CNN model for the detection of oyster mushroom maturity with a 91.7% training accuracy and a semiautomated harvesting system, integrating a Raspberry Pi for control, an electrical switch, an air compressor, and a pneumatic cylinder with a cutting knife to facilitate timely mushroom harvesting. The modularity and scalability of the system allow for industry-level usage and can be scaled according to the required mushroom-growing systems within the facility. The AI model, its underlying dataset, a digital twin for mushroom production, the setup of our growth and control chambers, and additional information are all made available under an open-source license.

Keywords: gourmet mushroom; digital twin; AI; Mask R-CNN; IoT; automation; sustainability

Citation: Sujatanagarjuna, A.; Kia, S.; Briechle, D.F.; Leiding, B. MushR: A Smart, Automated, and Scalable Indoor Harvesting System for Gourmet Mushrooms. *Agriculture* **2023**, *13*, 1533. <https://doi.org/10.3390/agriculture13081533>

Academic Editor: Gniewko Niedbala, Sebastian Kujawa, Tomasz Wojciechowski and Magdalena Piekutowska

Received: 1 July 2023

Revised: 24 July 2023

Accepted: 27 July 2023

Published: 1 August 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Gourmet mushrooms such as shiitake, oyster, and enoki are harvested in the wild or grown indoors in controlled environments. Harvesting mushrooms outside is a seasonal activity and thus is limited to a few months per year. Moreover, it is a labor-intensive process, and the changing outdoor conditions result in volatile harvests. Furthermore, climate change further limits outdoor mushroom growing and harvesting opportunities [1,2]. Indoor mushroom farms with controlled growth environments allow for an all-year-round growing and harvesting of mushrooms in sensor-controlled grow rooms and grow tents. Additionally, some approaches towards automated production of button mushrooms exist, e.g., [3].

However, those do not apply to gourmet mushrooms, and thus, it remains a labor-intensive process that requires skilled workers [4]. Moreover, gourmet mushroom production often relies on one-time-use plastic bags to hold the substrate. The plastic bag is cut open to start the mushroom fruiting process, which renders the plastic bag useless once the mushrooms have been harvested, thereby creating large amounts of plastic waste. Some production facilities have moved to reusable plastic containers or jars. Whether those are more sustainable remains an open question [5,6].

MushR aims to fill this gap by introducing a modular and scalable gourmet mushroom growing and harvesting system that extends the state of the art—which only monitors

and controls the growing environment—by introducing an image recognition system that detects when and which mushrooms are ready to be harvested in combination with a proof of concept of an automated mushroom harvesting mechanism for harvesting the mushrooms without human interaction. The image recognition setup monitors the growing status of the mushrooms and guides the robot arm during the harvesting process. The modularity and scalability of the system allow for industry-level usage and can be scaled according to the required mushroom-growing systems within the facility. As a result, MushR drastically reduces the necessity of manual labor for gourmet mushroom growing/harvesting and allows for further industrial-scale automation and increased yields and quality of mushrooms.

Specifically, this work addresses the following research questions:

RQ How to automate/digitize/enable sustainable indoor cultivation and harvesting of gourmet mushrooms?

RQ1 How to create a digital twin for the gourmet mushroom production process?

RQ2 How to monitor gourmet mushroom growth and detect which mushrooms are ready for harvest?

RQ3 How to automate harvesting of gourmet mushrooms?

Note that the production of oyster mushrooms (*Pleurotus ostreatus*) serves as a running case of this research as oyster mushrooms are among the most common gourmet mushrooms. Results obtained from the running case can be abstracted to other types of gourmet mushrooms due to their strong similarities of the production processes and parameters.

The paper is structured as follows: Section 2 presents supplementary literature, related works, and sustainability in mushroom production. In contrast, Section 3 describes an approach towards a digital twin for (gourmet) mushroom production. Section 4 details an AI model used to monitor the growth stages of gourmet mushrooms and the underlying dataset used to train the model that we created as part of this research. Section 5 outlines automated harvesting mechanisms for gourmet mushrooms. The results of our evaluation are presented in Section 6. Finally, in Section 7, we give our conclusions and outline possible directions for future research.

2. Supplementary Literature and Related Work

This section provides background information and supplementary literature and introduces related works. Section 2.1 briefly outlines the general process of gourmet mushroom production, while Section 2.2 focuses on related works. Finally, Section 2.3 details a more sustainable approach based on reusable mushroom pods.

2.1. Gourmet Mushroom Production

Indoor mushroom farms allow for an all-year-round growing and harvesting of mushrooms in sensor-controlled grow rooms and grow tents. The mushrooms are grown on substrate blocks, e.g., straw or wood chips, in one-time-use plastic bags. After a prepared substrate block is inoculated with a sample of mycelium (mycelium is a rootlike structure of a fungus consisting of a mass of branching, threadlike hyphae), it is kept in a dark, sterile environment for an incubation period. During incubation, the mycelium completely colonizes the substrate block, after which the plastic bags are cut open and placed in an environment with high humidity to initiate the fruiting of the mushrooms. They are typically harvested in cycles—so-called flushes—with idling periods in between flushes for the mushrooms (or, more precisely, the mushroom mycelium) to recover before triggering the next flush and starting the fruiting of mushrooms again. The substrate is depleted after about three flushes. While fruiting, the mushroom body grows fast and needs to be constantly monitored to be harvested at the right time. Additionally, the growing environment must be in perfect conditions, e.g., humidity, temperature, and air quality.

Figure 1 depicts oyster mushrooms in four different growth stages that we considered for the subsequent training of our AI model: First, in stage I, the fruiting process begins,

and the mushrooms start pinning. Next, in stage II, the mushroom body grows. Third, the mushroom matures and is ready to be harvested. In stage IV, the mushroom has passed the optimal harvesting time window and is now considered overdue.

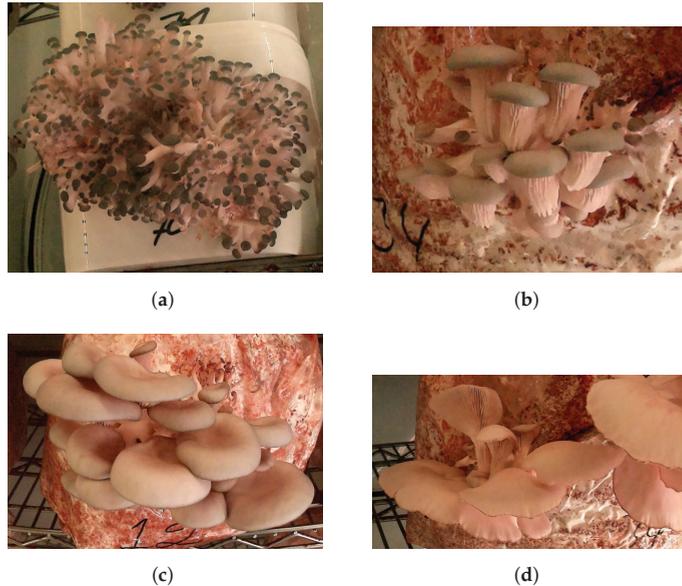


Figure 1. Progressing growth stages of *Pleurotus ostreatus*; (a) Stage I; (b) Stage II; (c) Stage III; (d) Stage IV.

Learning when and how to harvest mushrooms properly requires training and practice. Especially since harvesting is about more than just selecting the largest mushrooms. On the one hand, they tend to grow in dense clusters, and if they are not adequately thinned out (by hand), the mushrooms might damage each other. Moreover, it prevents the healthy, sustainable growth of the mushroom block, resulting in a decreased yield. Damaged or deformed mushrooms have to be sold at a discount. On the other hand, a particular cluster density is required; otherwise, the yield drops as well. Therefore, even though indoor mushroom growing facilities allow for an all-year-round harvest and optimized yield via controlled growing environments, it remains a labor-intensive process that requires skilled workers [4].

2.2. Related Work

Some level of automation can be achieved for button mushrooms (also known as chestnut mushrooms). Button mushrooms are grown on large soil beds instead of substrate blocks and are subsequently harvested by hand. Ref. [3] presented a robot arm-controlled suction cup for harvesting button mushrooms with a maturity recognition accuracy of 70.93 percent. Moreover, approaches towards sorting button mushrooms have been explored [7]. In contrast to button mushrooms, most gourmet mushrooms, such as the king oyster or the oyster mushroom, do not have a suction-cup-compatible surface. Furthermore, they are more sensitive than button mushrooms and thus require special care in their growing environment and during harvesting. Thus far, they are grown and harvested manually with little process automation; i.e., only the indoor growing environment is controlled and managed using a sensor-actuator setup for variables, such as temperature, humidity, and air quality [8,9]. Some rudimentary (semi-)automated harvesting mechanisms for mushrooms exist, e.g., [10–12], but they do not recognize the mushroom growth

status and optimal harvest time. They do not consider mushroom damage caused by the harvesting process.

Other works focus solely on detecting the growth stage of oyster and button mushrooms using different versions of the YOLO (You Only Look Once) object detection algorithm that have been proposed in the past, e.g., [13–15]. However, neither the model nor the underlying datasets have been published. Moreover, the combination of growth stage detection and automated harvesting was neglected.

Finally, some works consider reusable jars/bottles for mushroom production, e.g., [5,6], and some industry entities started using reusable plastic buckets instead of one-time-use-only mushroom bags [16,17]. However, an analysis of the environmental benefit of using reusable mushroom pods is missing.

2.3. Reusable Mushroom Pods

Aiming to reduce, if not eliminate, the plastic waste created using one-time-use plastic (polypropylene) bags as substrate containers, we experimented with an alternative approach: reusable plastic buckets.

In our experimental setup, we use plastic (polypropylene) buckets, which, weighing at 90 g, can contain up to 3 L of substrate. As shown in Figure 2, we drill 3.5 cm holes on five sides of the bucket, which function as fruiting and ventilation holes. These holes are sealed with a microporous tape during the incubation phase, which is removed for fruiting. Unlike one-time-use plastic bags, the plastic buckets, which we call mushroom pods, are never permanently damaged during the lifetime of the substrate contained therein.

We compute the environmental benefit of using reusable mushroom pods (in terms of materials) over one-time-use plastic bags by running life-cycle impact assessment calculations using OpenLCA [18]. To this end, we define the reference flow and functional unit for both container types in terms of “amount of polypropylene (g)” and “colonizable volume of the container in litres (L)”. We further quantify the relationship between the defined reference flow and the functional unit for both container types. Through experimentation, we estimate the colonizable volume of a 5 L one-time-use plastic bag weighing 30 g to be 3 L (on average). This decrease is because the bags do not have a built-in method to seal the contained substrate. A substantial amount of the bag is used for this purpose, accomplished by folding the opening over itself several times. Unlike the bags, our mushroom pods are equipped with a sealable lid. As a consequence, the estimated colonizable volume of our 3 L mushroom pods is still 3 L.

Using these calculations, we create a reference process for both these container types in OpenLCA. We source the life-cycle inventory (LCI) data pertaining to the manufacturing processes of polypropylene bags and buckets from the Agribalyse dataset [19]. The following are the life-cycle impact assessment (LCIA) results computed using the BEES+ impact assessment method:

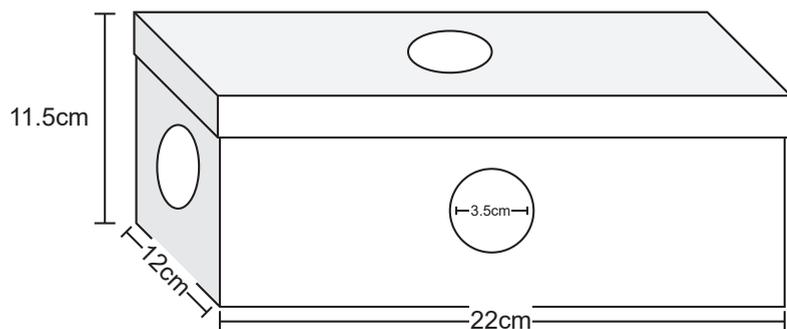


Figure 2. Graphical representation of the MushR mushroom pods.

From Table 1, we can infer the benefit of our reusable mushroom substrate pods over the non-reusable substrate bags with respect to the impact categories. Using “global warming” as an example, a 5 L (3 L usable) one-time-use bag has an impact of 102.46 g CO₂ eq, compared with a 3 L reusable pod with 217.20 g CO₂ eq. From this, we can infer that one reusable pod will have less impact in this category than three non-reusable bags. This means that simply reusing our mushroom pods three times will result in them having a lower impact in this category than the non-reusable counterparts. Similar comparisons can be made with other impact categories. The complete LCIA calculations and results are given in Appendix A. The above figures indicate that the reusable mushroom pods will have a lower environmental impact over time. These figures, however, are highly dependent on the specific materials used for producing the containers. These figures also might not scale linearly with the container size for mass-producing mushrooms.

Table 1. LCIA results comparison of 3 L colonizable volume of one-time-use bags and reusable pods.

Impact Category	Reference Unit	One-Time-Use Bag	MushR Reusable Pods
Acidification	H+ mmole eq	11.21	40.15
Ecotoxicity	g 2,4-D eq	2.07	0.72
Eutrophication	g N eq	0.43	0.34
Global warming	g CO ₂ eq	102.46	217.20
Habitat alteration	T&E count	1.88×10^{16}	-7.03×10^{17}
HH cancer	g C ₆ H ₆ eq	0.15	0.48
HH criteria air pollutants	microDALYs	0.004	0.015
HH noncancer	g C ₇ H ₇ eq	423.69	1124.07
Indoor air quality	g TVOC eq	0	0
Natural resource depletion	MJ surplus	0.26	0.92
Ozone depletion	g CFC-11 eq	2.64×10^7	1.49×10^6
Smog	g NO _x eq	0.14	0.47
Water intake	liters	0.95	2.3

3. Digital Twins for Gourmet Mushrooms

In the food industry, digital twins can provide better food quality, predictive maintenance, energy-use minimization, and higher transparency of the production processes [20].

The digital twin implementation developed for this project models the state of the key assets throughout various stages of the mushroom production process. Since the state of these assets is tightly linked with the state(s) of other inter-related assets, we chose to model this information using a graph database, Neo4j [21]. The various assets are modeled as nodes, and their temporal and nontemporal relationships are modeled using relations.

Figure 3 shows a class diagram of the graph database schema created for the database. Figure 3a shows a class hierarchy of the various node classes stored in the database. It further shows the possibility of relations that can be created between two node classes, including the directionality. These relations themselves have a class hierarchy, shown in Figure 3b.

Neo4j is incapable of enforcing rigid class hierarchy or data types of their attributes of nodes or relations. For this reason, we implement the class hierarchy using the Neomodel Object Graph Mapper (OGM) for Neo4j, which is then exposed using a REST-API, the MushR DigitalTwin API. All interactions with the data, including the creation of new assets, are handled through this API. Since the API itself does not store any of the digital twin data, the scalability of Neo4j’s various enterprise solutions can be leveraged to handle increasing amounts of data without compromising the API’s performance.

The defined schema allows the retrieval of the state of the network at any previous instance in time. For example, the graph stores the state of a substrate container and contains information on where it is located, which can be either a storage location or a grow chamber. The relation between an instance of a substrate with its substrate container implicitly gives its location history. Substrates are inoculated using samples of an existing mycelium spawn or newly procured mushroom strains.

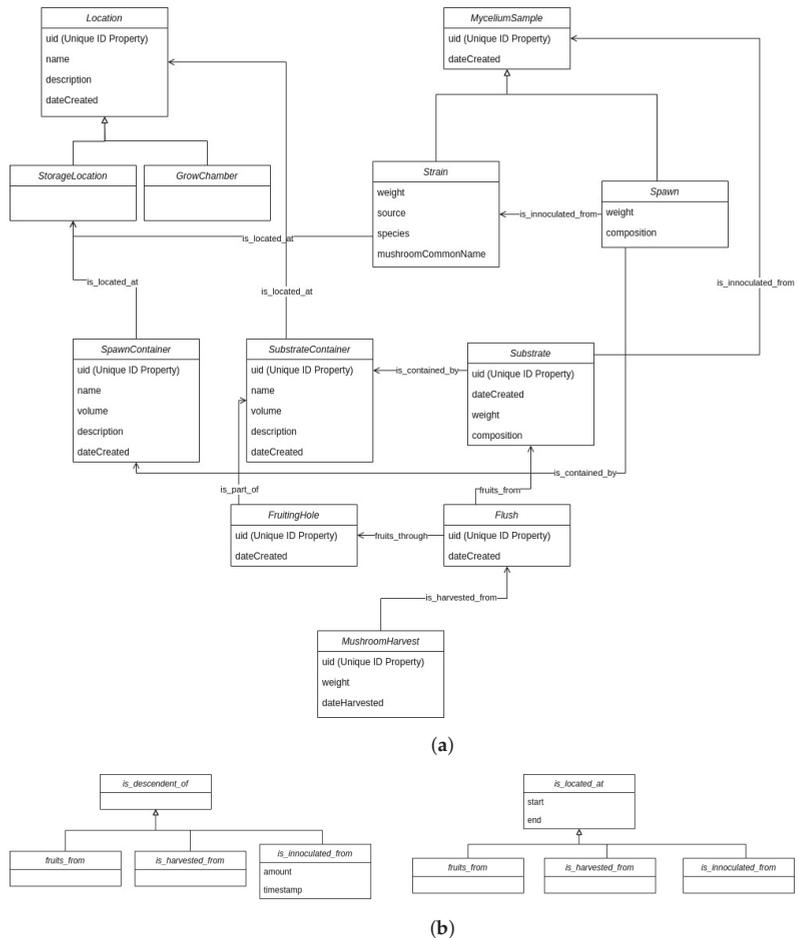


Figure 3. Digital twin database class diagram; (a) Node Classes; (b) Relation Classes.

The provenance of a specific mushroom harvest can be traced using the digital twin by recursively traversing the relations. Figure 4 shows a graph visualization of provenance based on data recorded over 8 months in 2022. The data are restricted to 300 nodes and relations pertaining to inoculation, fruiting, and harvests. The digital twin database also stores location-related information so that the location of any of the assets can be accessed at any instance of time.

More information about the MushR digital twin implementation can be found in our GitHub repository (Appendix A).

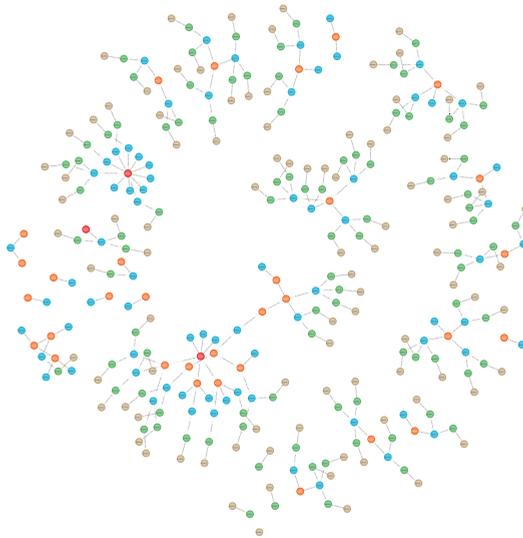


Figure 4. Digital twin graph visualization showing provenance of mushrooms (brown) harvested from flushes (green) fruiting from substrate (blue) inoculated from spawn (orange) or strain (red).

4. Detection and Monitoring of Gourmet Mushroom Growth Stages

Subsequent Section 4.1 provides details on our open-source dataset, followed by Section 4.2, which focuses on training our AI model and its predictions for gourmet mushroom growth stages.

4.1. Dataset

The dataset created for this project focuses on capturing images of the mushroom-growing environment from three different perspectives within each of our two growth tents for mushroom production. Instead of providing images of every individual bucket and mushroom, we capture the overall scene and its variations. The images from each perspective are captured simultaneously and automatically hourly. This approach allows for monitoring the development and maturity of the oyster mushrooms over time. We captured and accumulated 34,400 images using six Raspberry Pi HQ Cameras [22] equipped with wide-angle lenses (120° vertical field of view) over a period of 10 months to create a comprehensive dataset.

In the dataset preparation phase, we clean data to ensure that the focus of our project, which is detecting the maturity of oyster mushrooms, is maintained. As part of this process, we remove images that are foggy and have a significant amount of noisy areas or are contaminated. The selected dataset only included images where the mushrooms were clearly visible, allowing for training the model specifically for maturity detection. However, it is important to note that we have preserved the original raw dataset, including the foggy and black images on Kaggle (Appendix A). This is performed to provide a comprehensive dataset for other developers working on related projects or requiring access to diverse images. Researchers and enthusiasts in the field of agriculture automation can access and utilize the dataset for further analysis and experimentation. Since the images are not only from buckets and mushrooms, they enable the development of new algorithms and approaches for mushroom maturity detection and other related works. In addition to the raw images, we provide annotations for a subset of the dataset. Annotations were carefully created to mark the regions of mushrooms within the images. The annotation approach involves manual labeling of mushroom maturity regions based on visual cues, such as the mushrooms' color, shape, and size. The annotation provides a valuable resource for training and validating the Mask R-CNN model.

4.2. Training and Prediction

The machine learning model employed for the maturity detection of oyster mushrooms in this project is Mask R-CNN (region-based convolutional neural network) [23]. The model belongs to the family of instance segmentation models, which detect objects within an image and delineate their precise boundaries through pixel-level segmentation. This model is well suited for our task as it enables the accurate identification and localization of mature mushrooms within the captured images. The Mask R-CNN model architecture combines two key components, a region proposal network (RPN) [24] and a fully convolutional network (FCN) [25]. RPN generates a set of potential object proposals, while FCN performs pixelwise segmentation and classification for each proposed region. This two-stage approach allows the model to achieve robust performance by leveraging local and global contextual information effectively. To train this model, we annotated a subset of the custom dataset using CVAT [26]. We annotated *individual* mushroom fruiting bodies into three classes (based on three of the four stages outlined in Section 2.1): “not-ready”, “ready”, and “overdue”, indicating whether the mushroom can be harvested. The annotated subset of images serves as the ground truth for training and evaluating the model.

To train the model, we rely on the Detectron2 [27] library, a popular computer vision framework, for instance, segmentation that provides a comprehensive set of tools and predefined architectures for object detection and instance segmentation tasks. In our research, the Detectron2 library is the foundation for training and fine-tuning the Mask R-CNN model on our custom dataset. It optimizes the model’s performance specifically for oyster mushroom maturity detection. The Mask R-CNN model, integrated with the Detectron2 library, proves to be a robust and effective solution for the maturity detection of oyster mushrooms. See Table 2.

Table 2. MushR dataset overview, number of annotated instances per class.

Dataset	Not-Ready	Ready	Overdue	Total
Train	723	1344	692	2759
Test	251	198	92	541

For the training of the model, we initialize the model with weights pretrained from the COCO dataset [28]. We then train the model on our custom training dataset. We adopted the default Detectron2 hyperparameters (https://github.com/facebookresearch/detectron2/blob/main/configs/COCO-InstanceSegmentation/mask_rcnn_R_50_FPN_3x.yaml (accessed on 30 June 2023)) and used a base learning rate (post-warm-up) of 0.00015, with a batch size of 2 for 40,000 iterations. The model trained on an Nvidia GTX 4090 using 2.7 GB of memory at 10 iterations per second, and achieved a final training accuracy of 91.72%, as shown in Figure 5.

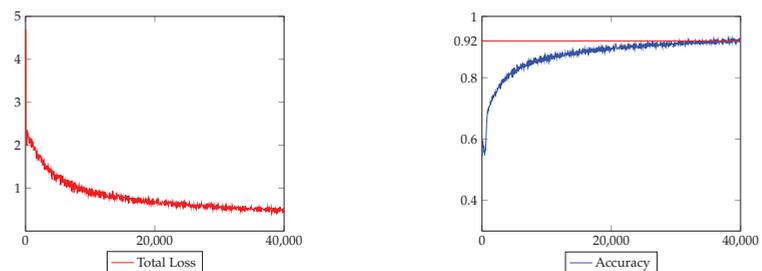


Figure 5. Mask R-CNN training curve.

Although the trained Mask R-CNN model segments and classifies *individual* mushrooms, in mushroom production, entire flushes are always harvested together rather than individual mushrooms. Therefore, an additional step is required to cluster the predicted masks and obtain a final prediction for the entire flush. This clustering process helps to consolidate the individual mushroom predictions into a cohesive prediction for the entire harvest, ensuring efficient and accurate harvesting in mushroom production.

We use the DBSCAN algorithm to cluster the predicted masks, computing the center of the masks to be used as proxy. We use $\epsilon = 2.5 \times 2 \times r$, where r is the average (approximate) radius of all the mask instance predictions in the input image. This value defines the neighborhood of a mask center point, and any masks that do not center within this neighborhood are not considered part of the same flush. Figure 6 shows example results of our maturity prediction workflow performed on a single image. In addition to this clustering, we compute each cluster's total area of the instance classes, Not-Ready, Ready, and Overdue. This computation improves the accuracy of the final maturity state prediction for the cluster over using a simple instance count.

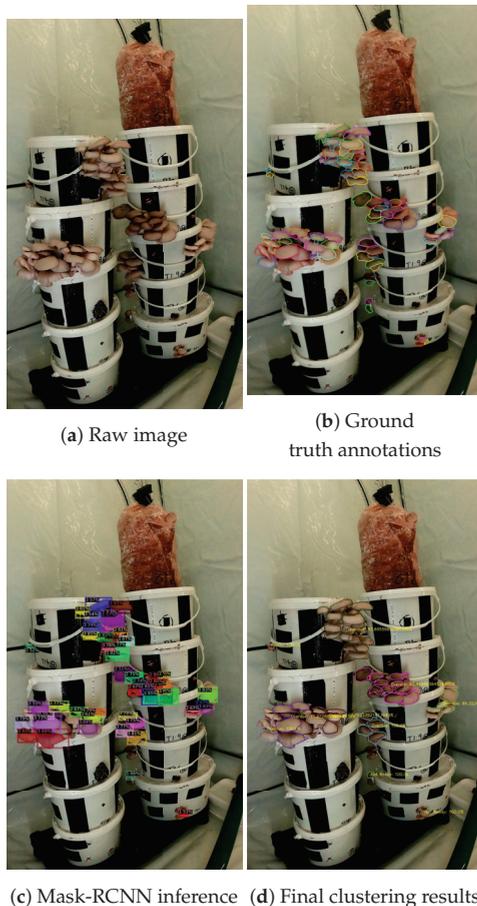


Figure 6. Example of oyster mushroom maturity prediction on a single image showing (a) raw image used for training; (b) annotated ground truth instances; (c) Mask R-CNN inference results on trained model, with each instance labeled with the predicted class; Not-Ready(1)/Ready(2)/Overdue(3) and class probability (%); and (d) final harvesting decision by clustering predicted instance masks, including class probability (%).

More information, such as the complete training configuration, train and test datasets used, and code used for clustering and visualization can be found in our GitHub repository (Appendix A).

5. Harvesting Gourmet Mushrooms

As outlined in Sections 2.1 and 2.2, it is infeasible to use suction cups to harvest gourmet mushrooms. In pursuit of the goal of a prototype harvesting system for gourmet, we followed two possible approaches: pneumatic and motorized harvesting, detailed in Sections 5.1 and 5.2, respectively.

5.1. Pneumatic Harvesting

Figure 7 shows the components used to prototype the pneumatic harvesting system. This system actuates a blade (B), which is mounted perpendicularly on a bidirectional piston rod cylinder (C). Figure 8 shows how the cylinder is set up to harvest oyster mushrooms growing from our developed mushroom pods. The pneumatic tubes T_1 and T_2 control the motion of the pneumatic cylinder. Depending on whether T_1 or T_2 is pressurized, the piston rod moves outwards or inwards from the cylinder, respectively. When both T_1 and T_2 are pressurized or depressurized, the piston is immobilized or free to move, respectively. For the purpose of harvesting, T_1 and T_2 must be pressurized and depressurized alternatively. This is accomplished by the 5/3-way solenoid valve (SV). Internally, two solenoids control the valves pressurizing T_1 and T_2 . These solenoids are activated by 12-volt DC magnetic coils, MG_1 and MG_2 . These coils are digitally controlled by 5-volt relay switches connected to a Raspberry Pi 4B [29], programmatically controlling the entire setup. More information about our experimental setup can be found in Appendix A.

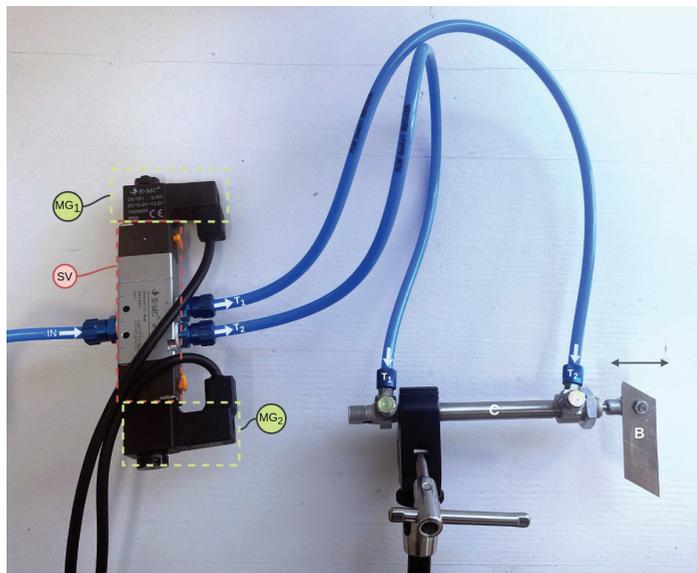


Figure 7. Pneumatic harvesting system components.



Figure 8. Pneumatic harvesting system setup to harvest mushrooms from a MushR mushroom pod.

5.2. Motorized Harvesting

Figure 9 shows the major components of the developed motorized harvesting system, which consists of a stainless steel cylinder (H) with fruiting holes suspended from the ceiling of a stool constructed using item profiles. The cylinder contains a plastic bag filled with a substrate for oyster mushroom growth. These holes on the cylinder align with the perforation in the plastic bag, allowing the mushrooms to emerge and grow. A 3D-printed lid (L), matching the diameter of the cylinder, seals the bottom to prevent substrate spillage in case of damage to the bag. A stepper motor (S) is installed underneath the suspended cylinder to drive a 3D-printed plate (B), which vertically holds the blade. The plate is specifically designed to accommodate the motor, ensuring stability during rotation.

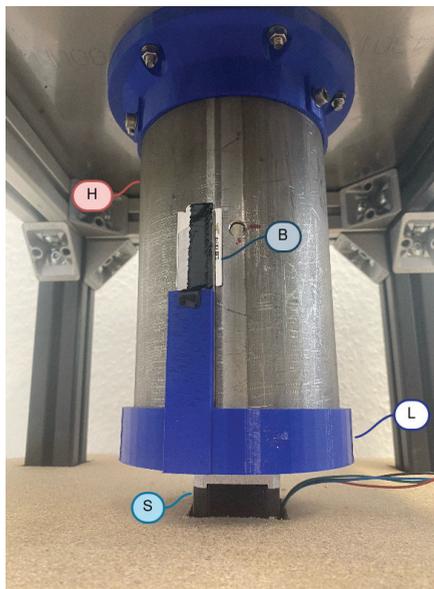


Figure 9. Motorized harvesting system.

The stepper motor initiates rotation, causing the blade to cut the mushrooms protruding through the holes in the cylinder. We program the motor to rotate in either direction to ensure successful harvesting. Adjustments, such as sharpening the knife and modifying the motor's speed, can be made to optimize the system's performance.

6. Evaluation

The developed digital twin implementation models the state of the key assets throughout various stages of the gourmet mushroom production process. While it does not provide process optimization for the gourmet mushroom production process, it is modeled to be capable of storing the temporal state of any of the essential assets, such as substrate, spawn and harvests involved in the production process, as detailed in Section 3.

Tables 3 and 4 evaluate the Mask R-CNN model described in Section 4. We evaluated the trained model in terms of average precision (AP) on our test set using the COCO evaluation metric (<https://cocodataset.org/#detection-eval> (accessed on 30 June 2023)). Table 5 provides a brief description of the evaluation metrics used.

Table 3. Mask R-CNN model evaluation using the COCO evaluation metric [30].

Criteria	AP	$AP^{IOU=0.50}$	$AP^{IOU=0.75}$	AP^{small}	AP^{medium}	AP^{large}
Bounding Box	61.876	71.016	67.920	20.792	69.788	87.207
Segmentation	49.332	70.639	60.812	9.010	54.799	80.218

Table 4. Mask R-CNN model evaluation (per class).

Task	Class	AP
Bounding Box	Not-Ready	34.386
	Ready	74.683
	Overdue	76.558
Segmentation	Not-Ready	26.502
	Ready	61.083
	Overdue	60.410

As represented in Table 4, the AP of the bounding boxes predicted by the Mask R-CNN model is generally larger than those of segmentation. The "Not-Ready" class of instances also has lower AP than the other classes. A similar trend can be seen in Table 3 with AP across scales: AP^{small} is significantly lower than AP^{medium} and AP^{large} . Since mushroom fruiting bodies that are "Not-Ready" are usually smaller in size, the resolution of the training images seems to play a role in the decrease in AP. The model has its largest AP under the evaluation criterion AP^{large} .

Table 5. Description of COCO metrics (adapted from [30]).

Metric	Description
AP	AP at IoU = 0.50:0.05:0.95
$AP^{IOU=0.50}$	AP at IoU = 0.50 (PASCAL VOC metric)
$AP^{IOU=0.75}$	AP at IoU = 0.75 (strict metric)
AP^{small}	AP for small objects: pixel area < 32 ²
AP^{medium}	AP for medium objects: 32 ² < pixel area < 96 ²
AP^{large}	AP for large objects: area > 96 ²

Sections 5.1 and 5.2 describe our prototype pneumatic and motorized harvesting systems, respectively. While they are capable of harvesting full flushes of mushrooms, the time required to harvest a single flush increases heavily on the water content and thickness of the flushes at the base of the fruiting holes. From our limited testing, the pneumatic harvesting mechanism setup, as shown in Figure 8, failed to produce any significant results

below 2.1 bar of pressure. At 7.5 bar, we could reliably harvest flushes, albeit requiring up to 5 s of continuous application to successfully harvest the thickest fruiting flushes. On the other hand, since the fruiting holes in the motorized harvesting system are significantly smaller, fruiting flushes can be harvested nearly instantaneously, given that the blade used is sufficiently sharp. The downside of using a smaller fruiting hole, however, is a decrease in the fruiting speed and yield quantity of the flushes.

Unlike the motorized harvesting mechanism, which uses a dedicated harvesting mechanism for each substrate container, the pneumatic harvesting system is an independent modular component. In an industrial-grade setup, since our mushroom pods have flat surfaces, they can be moved from their growth chambers when the mushrooms are ready to harvest (as detected by our Mask R-CNN-based maturity detection method) to the harvesting system via conveyor belts (or other industrial mobility solutions). The number of mushroom pods being used in production can hence increase faster than the number of harvesting systems, the rate of the increase being dependent on the desired throughput of the production environment, thereby allowing the system to be very scalable.

Discussion and Limitations

Our three main contributions, the digital twin, the Mask R-CNN-based maturity detection, and the automated harvesting mechanisms, are novel in their application towards gourmet mushroom production. They do, however, have several limitations that require further research to amend.

The digital twin implementation described in Section 3 did not use any process modeling to monitor or simulate the various production processes involved in gourmet mushroom production.

While our Mask R-CNN model's performance is promising, there are certain limitations to consider. Factors such as lighting conditions, oyster mushroom variations, and occlusions may affect the model's accuracy. As mentioned in Section 4, we accumulated 34,400 images of oyster mushrooms. However, due to the immensely time-consuming endeavor of annotating the mushrooms with masks, we could only annotate a small portion of those images. This subsequently limits the potential of our Mask R-CNN model to learn to segment these mushrooms.

Lastly, our developed methods have yet to be tested in an industrial gourmet mushroom production environment, and the results presented in this paper have the possibility of being only applicable to our experimental setup for oyster mushroom production.

7. Conclusions and Future Work

This work makes four contributions: First, we present a digital twin for gourmet mushroom production, which is capable of storing the temporal state of any of the important assets, such as substrate, spawn and harvests involved in the mushroom production process. We also provide a visual user interface to interact with the graph-based structure of the digital twin representation. Second, we introduce an image recognition system that determines when and which mushrooms are ready to be harvested in conjunction with a proof-of-concept of an automated mushroom harvesting mechanism for harvesting the mushrooms without human interaction. The image recognition setup monitors the growing status of the mushrooms and guides the harvesting process. Third, we present a Mask R-CNN model for the detection of oyster mushroom maturity, which has an evaluated average precision of up to 80.2 ($A^{P_{large}}$), as well as a semiautomated harvesting system that can be scaled, integrating a Raspberry Pi for control, an electrical switch, an air compressor, and a pneumatic cylinder with a cutting knife to facilitate timely mushroom harvesting as a third contribution. Fourth, we perform an analysis of the environmental benefit of using reusable mushroom pods in favor of one-time-use-only plastic bags. In our use case, reusing the mushroom pods three times yields a more eco-friendly output than previous approaches relying on plastic bags.

Finally, the AI model, its underlying dataset, a digital twin for mushroom production, the setup of our growth and control chambers, and additional information are all made available under an open-source license.

Future improvements could involve collecting a more extensive and more diverse dataset to enhance the model's robustness further, e.g., different types of gourmet mushrooms but also different settings. Additionally, fine-tuning the model with additional training iterations or exploring alternative architectures may yield even better results. Besides the automated harvesting mechanism, subsequent research, i.e., future work, may optimize the growth environment parameters to maximize mushroom growth and harvest based on collected sensor data and the digital twin/shadow data of previously harvested mushroom pods.

While this work focused on monitoring and detecting gourmet mushroom growth stages and subsequent harvesting activities, future work may also focus on other issues of the mushroom production process. Recognizing contaminated substrate or mushrooms further reduces human intervention. Our open-source dataset includes images exhibiting contamination, where mushrooms, buckets, tents, or even the camera may be contaminated. By keeping the complete dataset, we aim to support future research and enable the development of new projects beyond the scope of this specific maturity detection project.

Moreover, the monitoring system could also be coupled with the control unit of the growth environment to manipulate growth-related parameters (e.g., temperature) so that mushrooms are ready to be harvested on predetermined days.

Beyond the production of gourmet mushrooms, the solutions, e.g., growth/harvest monitoring and detection and the robot-based harvesting developed in MushR, may be generalized to other indoor farming activities in controlled environments beyond mushrooms and, in special cases, even for outdoor farming activities.

Author Contributions: Conceptualization, A.S., S.K., D.F.B. and B.L.; Methodology, A.S., S.K., D.F.B. and B.L.; Software, A.S. and S.K.; Validation, A.S. and S.K.; Investigation, S.K., D.F.B. and B.L.; Resources, A.S.; Data curation, A.S. and S.K.; Writing—original draft, A.S., S.K. and B.L.; Writing—review & editing, A.S., D.F.B. and B.L.; Visualization, A.S.; Supervision, B.L.; Project administration, A.S. and B.L.; Funding acquisition, A.S., D.F.B. and B.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Cascade Funding via the VEDLIoT Open Call. The VEDLIoT project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement no. 957197. We acknowledge the financial support from the Open Access Publishing Fund of Clausthal University of Technology.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: All data related to this project have been made available to the public and are listed in Appendix A.

Acknowledgments: The authors thank Harish Gundelli, Athira Mavoomkuttathil Sivachandran, Sepideh Sayadkouh, and Johannes Mayer for their support and contribution to the research presented in this work.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

1. MushR GitHub Repository: <https://github.com/ETCE-LAB/MushR> (accessed on 30 June 2023)
2. AI model and dataset to detect the growth stages of gourmet mushrooms (specifically oyster mushrooms): <https://github.com/ETCE-LAB/mushr-mask-r-cnn/> (accessed on 30 June 2023)
3. Raw underlying dataset for the AI model: <https://www.kaggle.com/datasets/etcelab/mushr-project-raw-image-dataset-oyster-mushrooms> (accessed on 30 June 2023)

4. Github repository of the digital twin for gourmet mushroom production: <https://github.com/ETCE-LAB/mushr-digitaltwin-api> (accessed on 30 June 2023)
5. Github repository for the growth chamber setup and control environment: <https://github.com/ETCE-LAB/MushR/blob/main/growthchamber-setup> (accessed on 30 June 2023)
6. LCA results (one-time-use plastic bags & MushR reusable mushroom pods): <https://github.com/ETCE-LAB/MushR/blob/main/lca-calculations> (accessed on 30 June 2023)

References

1. Kausserud, H.; Stige, L.C.; Vik, J.O.; Økland, R.H.; Høiland, K.; Stenseth, N.C. Mushroom Fruiting and Climate Change. *Proc. Natl. Acad. Sci. USA* **2008**, *105*, 3811–3814. [CrossRef] [PubMed]
2. Yang, X.; Luedeling, E.; Chen, G.; Hyde, K.D.; Yang, Y.; Zhou, D.; Xu, J.; Yang, Y. Climate Change Effects Fruiting of the Prize Matsutake Mushroom in China. *Fungal Divers.* **2012**, *56*, 189–198. [CrossRef]
3. Huang, M.; Jiang, X.; He, L.; Choi, D.; Pecchia, J.; Li, Y. Development of a Robotic Harvesting Mechanism for Button Mushrooms. *Trans. ASABE* **2021**, *64*, 565–575. [CrossRef]
4. Huang, M.; He, L.; Choi, D.; Pecchia, J.; Li, Y. Picking Dynamic Analysis for Robotic Harvesting of Agaricus Bisporus Mushrooms. *Comput. Electron. Agric.* **2021**, *185*, 106145. [CrossRef]
5. Mamiro, D.P.; Mamiro, P.S.; Mwatawala, M.W. Oyster Mushroom (*Pleurotus* spp.) Cultivation Technique Using Re-Usable Substrate Containers and Comparison of Mineral Contents With Common Leafy Vegetables. *J. Appl. Biosci.* **2014**, *80*, 7071–7080. [CrossRef]
6. Sánchez, C. Cultivation of *Pleurotus ostreatus* and Other Edible Mushrooms. *Appl. Microbiol. Biotechnol.* **2010**, *85*, 1321–1337. [CrossRef] [PubMed]
7. Wang, F.; Zheng, J.; Tian, X.; Wang, J.; Niu, L.; Feng, W. An Automatic Sorting System for Fresh White Button Mushrooms Based on Image Processing. *Comput. Electron. Agric.* **2018**, *151*, 416–425. [CrossRef]
8. Mohammed, M.; Azmi, A.; Zakaria, Z.; Tajuddin, M.; Isa, Z.; Azmi, S. IoT-based Monitoring and Environment Control System for Indoor Cultivation of Oyster Mushroom. *J. Phys. Conf. Ser.* **2018**, *1019*, 012053. [CrossRef]
9. Ariffin, M.A.M.; Ramli, M.I.; Amin, M.N.M.; Ismail, M.; Zainol, Z.; Ahmad, N.D.; Jamil, N. Automatic Climate Control for Mushroom Cultivation Using IoT Approach. In Proceedings of the 2020 IEEE 10th International Conference on System Engineering and Technology (ICSET), Shah Alam, Malaysia, 9 November 2020; pp. 123–128.
10. Rong, J.; Wang, P.; Yang, Q.; Huang, F. A Field-tested Harvesting Robot for Oyster Mushroom in Greenhouse. *Agronomy* **2021**, *11*, 1210. [CrossRef]
11. Yaong, Q.; Rong, J.; Wang, P.; Yang, Z.; Genc, P. Real-time Detection and Localization Using SSD Method for Oyster Mushroom Picking Robot. In Proceedings of the 2020 IEEE International Conference on Real-Time Computing and Robotics (RCAR), Asahikawa, Japan, 28–29 September 2020; pp. 158–163.
12. Huang, M.; Jiang, X.; He, L.; Choi, D.; Pecchia, J. Hand-picking Dynamic Analysis for Robotic Agaricus Mushroom Harvesting. In Proceedings of the 2020 ASABE Annual International Virtual Meeting, Virtual, 13–15 July 2020; p. 1.
13. Lu, C.P.; Liaw, J.J.; Wu, T.C.; Hung, T.F. Development of a Mushroom Growth Measurement System Applying Deep Learning for Image Recognition. *Agronomy* **2019**, *9*, 32. [CrossRef]
14. Moysiadis, V.; Kokkonis, G.; Bibi, S.; Moscholios, I.; Maropoulos, N.; Sarigiannidis, P. Monitoring Mushroom Growth with Machine Learning. *Agriculture* **2023**, *13*, 223. [CrossRef]
15. Wang, Y.; Yang, L.; Chen, H.; Hussain, A.; Ma, C.; Al-gabri, M. Mushroom-YOLO: A Deep Learning Algorithm for Mushroom Growth Recognition Based on Improved YOLOv5 in Agriculture 4.0. In Proceedings of the 2022 IEEE 20th International Conference on Industrial Informatics (INDIN), Perth, WA, Australia, 25–28 July 2022; IEEE: New York, NY, USA, 2022; pp. 239–244.
16. Broussard, W. How to Grow Mushrooms in Buckets & Containers. 2023. Available online: <https://northspore.com/blogs/the-black-trumpet/growing-mushrooms-in-buckets-containers> (accessed on 30 June 2023).
17. Shields, T. Grow Mushrooms Easy in a 5 Gallon Bucket—Freshcap. 2023. Available online: <https://learn.freshcap.com/growing/bucket-grow/> (accessed on 30 June 2023).
18. GreenDelta. OpenLCA—Github Repository. 2023. Available online: <https://github.com/GreenDelta/olca-app> (accessed on 30 June 2023).
19. AGRIBYLASE Program. Agribylase Dataset 3.0. 2022. Available online: <https://nexus.openlca.org/database/Agribalase> (accessed on 30 June 2023).
20. Verboven, P.; Defraeye, T.; Datta, A.K.; Nicolai, B. Digital Twins of Food Process Operations: The Next Step for Food Process Models? *Curr. Opin. Food Sci.* **2020**, *35*, 79–87. [CrossRef]
21. Neo4j. Neo4j: Graphs for Everyone—Github Repository. 2023. Available online: <https://github.com/neo4j/neo4j> (accessed on 30 June 2023).
22. Raspberry Pi Foundation. Raspberry Pi HQ Camera. 2023. Available online: <https://www.raspberrypi.com/products/raspberrypi-high-quality-camera/> (accessed on 18 July 2023).

23. Matterport, Inc. Mask R-CNN for Object Detection and Segmentation—Github Repository. 2019. Available online: https://github.com/matterport/Mask_RCNN (accessed on 30 June 2023).
24. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection With Region Proposal Networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 91–99. [CrossRef] [PubMed]
25. Long, J.; Shelhamer, E.; Darrell, T. Fully Convolutional Networks for Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
26. OpenCV. Computer Vision Annotation Tool (CVAT)—Github Repository. 2023. Available online: <https://github.com/opencv/cvat> (accessed on 30 June 2023).
27. Meta Research. Detection2—Github Repository. 2023. Available online: <https://github.com/facebookresearch/detector2> (accessed on 30 June 2023).
28. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. In Proceedings of the Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014; pp. 740–755.
29. Raspberry Pi Foundation. Raspberry Pi 4 Tech Specs. 2023. Available online: <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/> (accessed on 30 June 2023).
30. COCO Consortium. COCO Dataset—Detection Evaluation. 2021. Available online: <https://cocodataset.org/#detection-eval> (accessed on 30 June 2023).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.



Article

An Adaptive Nutcracker Optimization Approach for Distribution of Fresh Agricultural Products with Dynamic Demands

Daqing Wu ^{1,2,†}, Rong Yan ^{1,†}, Hongtao Jin ^{1,*} and Fengmao Cai ³¹ College of Economics and Management, Shanghai Ocean University, Shanghai 201306, China² Department of Nanchang Technology, 901 Yingxiang Dadao, Economic and Technological Development Zone, Nanchang 330044, China³ Department of Electronic and Electrical Engineering, The University of Sheffield, Sheffield S37AZ, UK

* Correspondence: htjin@shou.edu.cn

† These authors contributed equally to this work.

Abstract: In the operational, strategic and tactical decision-making problems of the agri-food supply chain, the perishable nature of the commodities can represent a particular complexity problem. It is, therefore, appropriate to consider decision support tools that take into account the characteristics of the products, the needs and the requirements of producers, sellers and consumers. This paper presents a green vehicle routing model for fresh agricultural product distribution and designs an adaptive hybrid nutcracker optimization algorithm (AH-NOA) based on k-means clustering to solve the problem. In the process, the AH-NOA uses the CW algorithm to increase population diversity and adds genetic operators and local search operators to enhance the global search ability for nutcracker optimization. Finally, the experimental data show that the proposed approaches effectively avoid local optima, promote population diversity and reduce total costs and carbon emission costs.

Keywords: dynamic demand; adaptive nutcracker optimizer algorithm; green vehicle routing problem; fresh agricultural products

Citation: Wu, D.; Yan, R.; Jin, H.; Cai, F. An Adaptive Nutcracker Optimization Approach for Distribution of Fresh Agricultural Products with Dynamic Demands. *Agriculture* **2023**, *13*, 1430. <https://doi.org/10.3390/agriculture13071430>

Academic Editors: Gniewko Niedbala, Sebastian Kujawa, Magdalena Piekutowska and Tomasz Wojciechowski

Received: 7 June 2023
Revised: 14 July 2023
Accepted: 16 July 2023
Published: 19 July 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In recent years, China has issued many relevant documents on accelerating the high-quality development of cold chain logistics and transportation. It has led to the great development of the fresh agricultural products supply chain. More people are choosing to buy fresh agricultural products through e-commerce. On the one hand, fresh agricultural products are perishable, with a short shelf life, high losses and require temperature control. On the other hand, during the cold chain logistics delivery process, fuel consumption and carbon emissions are much higher than in normal logistics. The harm to the environment is higher. Therefore, the study of the fresh dynamic vehicle routing problem will be significant in reducing the commodity value loss, achieving green logistics and sustainable economic development.

This paper investigates the green capacitated vehicle routing problem with dynamic demand (GCVRPDD), which encompasses four topical problems in current vehicle routing research: the vehicle routing problem in cold chain logistics distribution (VRPCLD), the green vehicle routing problem (GVRP), the dynamic vehicle routing problem (DVRP) and the capacitated vehicle routing problem (CVRP). For VRPCLD, Hsu et al. [1] studied the optimal food delivery cycle under multi-vehicle delivery and multi-temperature delivery. Gharehyakheh et al. [2] combined the path optimization problem with temperature, commodity shelf life and energy consumption prediction models. Wu [3] studied the time-dependent split delivery green vehicle routing problem with multiple time windows (TSDSGVRPMTW). Zhang [4] started with a minimized delivery cost function to obtain

a more satisfying cold chain delivery. From the above, it can be found that the multi-temperature distribution of fresh agricultural products provides a methodological entry point for research on cold chain distribution.

For the GVRP, Zhou et al. [5] provided a review of models and solution algorithms for green vehicle paths. Zhang et al. [6] compared the effect of considering carbon emissions on path optimization in solving DVRP. Elgharably et al. [7] proposed a stochastic GVRP under the condition that economic, environmental and social aspects are considered simultaneously. Bruglieri et al. [8], based on new energy vehicles, studied the vehicle routing problem in the existence of gas stations. Zhou et al. [9], Li and Zhang [10] and Cai et al. [11] took the optimization of speed into account when studying the carbon emission pollution problem. Yin et al. [12] took the carbon emission allowance and trading policy as the research object and built an upper-layer carbon trading benefit model from the government's perspective and a lower-layer model from the company's perspective to describe the vehicle routing problem based on the Stackelberg game framework. The above research literature provides the theoretical basis for the calculation of carbon emission costs in this paper.

For the DVRP, Yang et al. [13] proposed a dynamic optimization strategy based on linear programming theory to cope with customer demand changes. Zhang et al. [14] analyzed the dynamic events and transformed the dynamic problem into a static problem. Guo et al. [15] studied the dynamic carpooling problem based on the school buses dynamically carrying students and solved it based on the decomposition of the shared network concepts. Pan et al. [16] proposed a deep reinforcement learning framework to meet the uncertain customer service demands and the training path planning process dynamics. From the above, the majority of research on the green cold chain vehicle path problem has focused on the static vehicle path problem, without considering the customers' dynamic demands in fresh agricultural products distribution.

For the CVRP, Kucuk et al. [17] presented constraint programming-based solution approaches for the three-dimensional loading CVRP. Aydinalp Birecik et al. [18] presented an interactive fuzzy approach for solving green CVRP with imprecise travel time for each vehicle and supplier demand. Wang et al. [19] proposed a novel genetic programming approach to simplify the routing policies. Souza et al. [20] proposed a hybrid algorithm based on a discrete adaptation of the differential evolution meta-heuristic, which is designed for continuous problems, combined with local search procedures to solve the CVRP. The above literature provides many excellent algorithms for solving the CVRP.

Vehicle routing problems are difficult to solve using exact mathematical analytical methods. While heuristic algorithms have good results in solving vehicle routing optimization problems, such as ant colony algorithms [21–23], genetic algorithms [24–26], and particle swarm algorithms [27,28]. Genetic algorithms have disadvantages such as large computational effort and slow convergence, and heuristic algorithms have the disadvantage of large influencing parameters. The nutcracker optimizer algorithm (NOA) [29] has the advantages of high accuracy, fast calculation speed and few parameters. However, there are disadvantages such as easily falling into local optimum and slow convergence speed. To cope with these disadvantages, this paper investigates the impact of carbon emissions on dynamic vehicle path optimization based on temperature control. Based on temperature and geographical location, customers are clustered. To establish a distribution optimization model to minimize the total cost, which contains the vehicle dispatch cost, vehicle transportation cost, value loss, temperature control cost and carbon emission cost, this paper designs an adaptive hybrid nutcracker optimization algorithm (AH-NOA) based on k-means clustering to plan pathways and provides a theoretical basis for logistics companies.

The innovations and contributions of this paper are described as follows:

- An adaptive hybrid nutcracker optimization algorithm combining genetic algorithm and local search operation is designed, which considers both the search breadth and depth. The population is disturbed by the crossover and mutation operation of the

genetic algorithm, and the excellent nutcrackers in the population are deeply searched by the local search operation.

- The GCVRPDD model has been applied rarely in the research of fresh commodity distribution. To make the model better simulate the actual conditions of fresh commodity distribution, this study will consider and evaluate dynamic demand, carbon emissions and temperature control on the basis of GCVRPDD.
- In order to improve the quality and diversity of the initial population, two different methods were used to generate the initial population in this paper. The two methods are, respectively, the CW saving algorithm and the Random method.

The remainder of this paper is organized as follows: Section 2 briefly introduces the problem and constructs the GCVRPDD model. Section 3 presents a new solution algorithm. In Section 4, the results of the experiments are analyzed. Finally, the conclusion is given in Section 5.

2. Problem Description and Model Construction

2.1. Problem Description

The GCVRPDD proposed in this paper can be described as follows: the cold chain distribution vehicles depart from the distribution center visit customer nodes in the order of the distribution scheme, update the distribution route periodically when the customer's demand is dynamically adjusted and finally return to the distribution center. Suppose $\{0\} \cup V_c$ denotes the set of nodes and V_c denotes the set of customer points before starting distribution. At this time, the customer coordinates and the goods demand q_{im}^w is known. $K = \{1, 2, 3, \dots, k\}$ is the set of vehicles and the models are the same. The operating hours of the distribution center are $[T_s, T_f]$. The vehicle does not stay at the customer's location during the delivery. The road conditions are stable and fluctuations in vehicle speeds can be ignored. The dispatch center can hold all the information and regulate the vehicles in real time. At the same time, the delivery time does not exceed T_k in order to ensure the goods are in good condition.

2.2. Vehicle Fuel Consumption Model

Demir et al. [30], Suzuki [31] and Hickman [32] have studied the influence of vehicle fuel consumption and carbon emissions. It is pointed out that the distance traveled plays a major role in fuel consumption and carbon emissions. In addition, factors such as vehicle speed, load and characteristics also have a significant impact. At the same time, various fuel consumption and carbon emission measurement models are proposed. In this paper, the MEET model proposed by Hickman is used to complete the calculation. The MEET model includes a carbon emission rate estimation function, a load correction factor and a road slope correction factor. This is applicable to heavy goods vehicles in the weight range of 3.5–32 t. The carbon emission estimation function is:

$$\varepsilon(v) = \omega_0 + \omega_1 v + \omega_2 v^2 + \omega_3 v^3 + \frac{\omega_4}{v} + \frac{\omega_5}{v^2} + \frac{\omega_6}{v^3} \tag{1}$$

where $\varepsilon(v)$ denotes the carbon emission rate when the vehicle is unloaded and driving on a road with a slope of zero. v denotes the vehicle speed. $\omega_0, \omega_1, \omega_2, \omega_3, \omega_4, \omega_5$ and ω_6 denotes a predefined parameter, which takes different values for trucks with different loads. The load correction factor (LC) is as follows:

$$LC = x_0 + x_1 \gamma + x_2 \gamma^2 + x_3 \gamma^3 + x_4 v + x_5 v^2 + x_6 v^3 + \frac{x_7}{v} \tag{2}$$

where γ denotes the ratio between the actual load of the vehicle and its capacity. $\chi_0, \chi_1, \chi_2, \chi_3, \chi_4, \chi_5, \chi_6$ and χ_7 are predefined parameters. The value varies with trucks. The vehicle carbon emission rate is $c = \varepsilon(v)LC/1000$.

During the whole distribution process, the carbon emissions produced by the unit of fuel consumption usually have a fixed value. If the carbon emission caused by 1 L

of gasoline is 2.3 kg, the fuel consumption generated by 1 kg of carbon emission is $\hat{o} = 1/2.3 = 0.4348$ L/kg. The vehicle fuel consumption rate is $f = \hat{o}c$. Therefore, the fuel consumption of vehicle k traveling from node i to j is $F_{ijw}^k = fd_{ij}$.

2.3. Fresh Agricultural Products Quality Attenuation Model

During transportation, the quality of fresh agricultural products gradually declines. The decline rate is strongly related to the transportation environment, time and transport process stability [33,34].

In this paper, the method discussed in [35] is used to calculate the value loss (vl_{im}^{wk}) based on the time length and temperature ($T(K)$).

$$vl_{im}^{wk} = q_{im}^w \left[1 - e^{-\lambda \partial_m^w t_{i0w}^k} \right] p_m \tag{3}$$

where q_{im}^w denotes the demand of customer i for the fresh products (m) at a temperature of w . ∂_m^w denotes the freshness factor of a unit fresh product (m) for delivery at a temperature of w . p_m is the price of the fresh commodity m . t_{i0w}^k denotes the delivery time that the vehicle takes to get from the distribution center to the customer (i).

2.4. Temperature Cost Model

During transport, different fresh agricultural products have different temperature requirements and the delivery vehicle needs to set the appropriate temperature to suit the attributes carried. When the performance index COP is used to calculate the energy value change [36], there is $COP = T_L / (T_H - T_L)$. Where T_H is the absolute outside ambient temperature and T_L is the absolute target temperature after temperature control. The physical quantity corresponding to the absolute temperature is the thermodynamic temperature, denoted as $T(K)$, with the symbol K . $T(K)$ is related to the Celsius temperature (t) by the equation $T(K) = 273 + t(^{\circ}C)$. For example, when T_H is $25^{\circ}C$ (298 K) and T_L is $6^{\circ}C$ (279 K). $COP = 14.68$ means that at an ambient temperature of $25^{\circ}C$, the vehicle temperature is maintained at 6 and needs the temperature control device to absorb 14.68 units of heat. Assuming a temperature control cost of 1 at this point, the temperature control cost factor (θ_{mw}) of $5^{\circ}C$ is $14.68 \cdot 13.9^{-1} \approx 1.06$. Similarly, the temperature control cost factor (θ_{mw}) for the other temperature conditions can be obtained, as shown in Table 1.

Table 1. θ_{mw} under different temperatures.

Temperature	COP	θ_{mw}	Temperature	COP	θ_{mw}
$6^{\circ}C$	14.68	1	$-3^{\circ}C$	9.64	1.52
$5^{\circ}C$	13.9	1.06	$-4^{\circ}C$	9.28	1.58
$4^{\circ}C$	13.19	1.11	$-5^{\circ}C$	8.93	1.64
$3^{\circ}C$	12.55	1.17	$-8^{\circ}C$	8.03	1.83
$2^{\circ}C$	11.96	1.23	$-9^{\circ}C$	7.76	1.89
$1^{\circ}C$	11.42	1.29	$-10^{\circ}C$	7.51	1.95
$0^{\circ}C$	10.92	1.34	$-11^{\circ}C$	7.28	2.02
$-1^{\circ}C$	10.46	1.40	$-12^{\circ}C$	7.05	2.08
$-2^{\circ}C$	10.04	1.46	$-13^{\circ}C$	6.84	2.15

2.5. Green Capacitated Vehicle Routing Problem with Dynamic Demand Model

In this paper, a combination of the initial distribution path and dynamic adjustment strategy is used to solve GCVRPDD. The concept is shown in Figure 1.

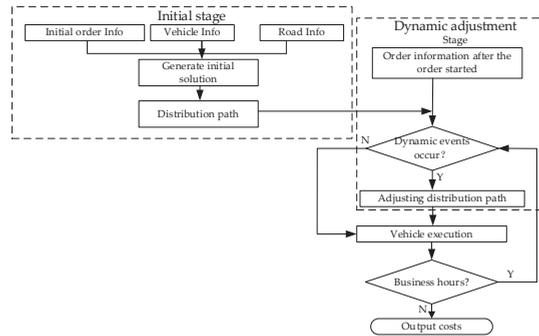


Figure 1. Initial planning + dynamic adjustment strategy.

2.5.1. Initial Stage

(1) Symbol Description

Based on the needs of building the model, this paper uses the corresponding symbols which are listed in Table 2.

Table 2. Symbol definition in the GCRPDD optimization.

Parameter	Definition	Parameter	Definition
C_1	The vehicle dispatch cost	st_0^k	Time of vehicle departure from distribution center
C_2	The transportation cost per distance	T_k	The maximum vehicle time in transit
C_3	The fuel price	W	A set of temperature control ranges
C_4	The unit price of carbon emissions	F_1	The delivery vehicle dispatch cost
Cap	The rated load capacity of the vehicle	F_2	The transport cost
TS_i	The customer service time	F_3	The temperature control cost
d_{ij}	The distance between i and j	F_4	The temperature control cost
t_{ijw}^k	The travel time of vehicle k from i to j at temperature w	F_5	The carbon emission cost
x_{ij}^{wk}	0–1 decision variable, 1 if vehicle k travels from i to j at temperature w and 0 otherwise	$u_{mw}^k = C_3 f_{kw} \theta_{mw}$	The temperature-controlled cost of fresh commodity m at temperature w
Q_{ijw}^k	The load capacity of vehicle k from customer i to j at w	f_{kw}	The fuel consumption of a hundred kilometers of vehicle k at temperature w
a	The carbon emissions per hour of temperature-controlled equipment	θ_{mw}	The temperature-controlled cost factor unit of fresh commodity at w
V_q^a	The new set of customer	T_s^{ak}	The vehicle's departure time at the distribution center
V_c^a	The original set of orders not served constitutes	VD	A set of virtual distribution center
z_k	0–1 decision variable, the newly dispatched vehicle	Q^k	The remaining goods of the vehicle k
K^a	The set of vehicles currently performing the delivery task	K^{ac}	The set of new vehicles to be dispatched (K^{ac} is the complement of K^a)
T_{qi}^a	The change moment in customer demand	st_d^k	The time when the vehicle leaves the distribution center or virtual distribution center.
T_{end}	The order deadline		

(2) Initial Stage Vehicle Path Model

Objective function:

$$Z = \min\{F_1 + F_2 + F_3 + F_4 + F_5\} \tag{4}$$

$$F_1 = C_1 \sum_{j \in V_c} \sum_{k \in K} x_{0j}^{wk} \tag{5}$$

$$F_2 = C_2 \sum_{w \in W} \sum_{j \in \{0\} \cup V_c} \sum_{i \in \{0\} \cup V_c} \sum_{k \in K} F_{ijw}^k x_{ij}^{wk} \tag{6}$$

$$F_3 = \sum_{k \in K} \sum_{m \in M} \sum_{w \in W} \sum_{j \in \{0\} \cup V_c} \sum_{i \in \{0\} \cup V_c} t_{ijw}^k u_{mw}^k Q_{ijw}^k \tag{7}$$

To simplify the calculation, it is assumed that no temperature control cost is generated in return.

$$F_4 = \sum_{i \in V_c} \sum_{m \in M} \sum_{w \in W} \sum_{k \in K} v_{im}^{wk} \tag{8}$$

$$F_5 = C_4 \sum_{w \in W} \sum_{j \in \{0\} \cup V_c} \sum_{i \in \{0\} \cup V_c} \sum_{k \in K} \left(F_{ijw}^k x_{ij}^{wk} / \hat{o} + at_{ijw}^k x_{ij}^{wk} \right) \tag{9}$$

s.t.

$$\sum_{i \in \{0\} \cup V_c} \sum_{j \in V_c} \sum_{w \in W} x_{ij}^{wk} q_{jm}^w \leq Cap, \forall k \in K \tag{10}$$

$$\sum_{j \in \{0\} \cup V_c} \sum_{k \in K} \sum_{w \in W} x_{ij}^{wk} = 1, \forall i \in V_c \tag{11}$$

$$\sum_{i \in \{0\} \cup V_c} \sum_{k \in K} \sum_{w \in W} x_{ij}^{wk} = 1, \forall j \in V_c \tag{12}$$

$$\sum_{i \in \{0\} \cup V_c} \sum_{w \in W} x_{ij}^{wk} = \sum_{i \in \{0\} \cup V_c} \sum_{w \in W} x_{ji}^{wk}, \forall j \in V_c, \forall k \in K \tag{13}$$

$$\sum_{j \in V_c} \sum_{w \in W} x_{0j}^{wk} = \sum_{i \in V_c} \sum_{w \in W} x_{0i}^{wk} \leq 1, \forall k \in K \tag{14}$$

$$T_s + \sum_{i \in \{0\} \cup V_c} \sum_{j \in \{0\} \cup V_c} \sum_{w \in W} t_{ijw}^k x_{ij}^{wk} + \sum_{i \in \{0\} \cup V_c} \sum_{j \in V_c} \sum_{w \in W} q_{jm}^w TS_j x_{ij}^{wk} \leq T_f, \forall k \in K \tag{15}$$

$$\sum_{i \in \{0\} \cup V_c} \sum_{j \in V_c} \sum_{w \in W} d_{ij} x_{ij}^{wk} \leq vT_k, \forall k \in K \tag{16}$$

$$\sum_{i \in S} \sum_{j \in S} \sum_{w \in W} x_{ij}^{wk} \leq |S| - 1, S \subseteq V_c, S \neq \emptyset, \forall k \in K \tag{17}$$

$$\sum_{i, j \in V_c} \sum_{k \in K} \sum_{w \in W} x_{ij}^{wk} \leq |K|, \forall k \in K \tag{18}$$

$$x_{ij}^{wk} \in \{0, 1\}, \forall i, j \in \{0\} \cup V_c, \forall k \in K, w \in W \tag{19}$$

Equation (10) indicates that the sum of the orders does not exceed the vehicle load limit. Equations (11) and (12) indicate that the customer point is served by a vehicle once and only once. Equation (13) indicates that the vehicle must leave after completing the order. Equation (14) represents each vehicle serving only one route, and the vehicle departs from the distribution center and returns after completing the order. Equation (15) represents the vehicle returns to the distribution center at a time no later than the operating deadline. Equation (16) represents the delivery time constraint. Equation (17) represents the

elimination of the sub-circuit constraint. Equation (18) represents the number of vehicles performing distribution tasks that do not exceed the maximum number in the distribution center. Equation (19) is the decision variable.

2.5.2. Dynamics Adjustment Stage

(1) Dynamic Adjustment Strategy

The dynamic character of GCVRPDD is mainly reflected in customer order updates, where the demand changes in real time during the distribution. In the dynamic adjustment stage, this paper divides the operation time into several homogeneous time slices and adopts a periodic optimization strategy for optimization. Dynamic events that occur in the current time slice are not processed immediately but are globally optimized in the next time slice together with customers who have not received service in the path. With demand dynamically updated, the VRPs within each time slice are static, and the optimization solutions S are continuously passed between time slices. As a result, GCVRPDD is converted into a number of correlated static vehicle routing problems (SVRP), as shown in Figure 2. To simplify the calculation, it is set that the customer being delivered cannot change the demand.

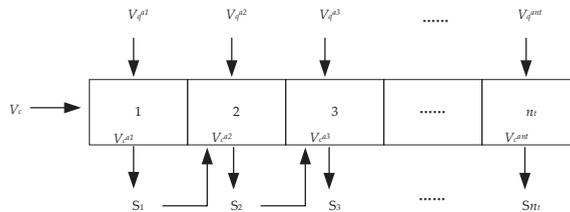


Figure 2. GCVRPDD rolling time domain dynamic optimization.

The number of time slices is set to n_t , and each time slice is $(T_f - T_s) / n_t$ [37]. The dynamic demand in the current time slice is not processed immediately and will be passed to the next time slice for optimization. With a larger n_t , the dynamic demand response is faster. However, a larger n_t does not necessarily result in a better pathway and also leads to system redundancy in computational space [38].

(2) Dynamic Adjustment Model

In this paper, according to the dynamic event occurrence time, the vehicle location is determined and set as a virtual distribution point. The objective is to minimize the cost and value loss to complete subsequent distribution tasks after a dynamic event has occurred. During the optimization process, the position of vehicle k and the goods surplus (Q^k) in the dynamic adjustment phase can be determined from the previous phase. According to parameters, such as vehicle speed and load, the cost and value loss of subsequent distribution tasks are solved. Suppose the vehicle's (k) initial distribution path is 0-1-2-3-4-0. In dynamic adjustment, if the vehicle is at customer point 2, the reconfiguration takes customer point 2 as the virtual distribution center. At this point, the goods remaining of vehicle k is $Q^k = Cap - q_{1m}^w - q_{2m}^w$. If the vehicle is between customer points 2 and 3, the reconfiguration takes the vehicle's current location as the virtual point and does not change the delivery task to customer point 3. The goods remaining are $Q^k = Cap - q_{1m}^w - q_{2m}^w - q_{3m}^w$. If the optimized subsequent distribution path is 3-5-6-0, the distribution path for vehicle k is 0-1-2-3-5-6-0, based on the previous path. The model is as follows:

$$Z = \min\{F_1 + F_2 + F_3 + F_4 + F_5\} \tag{20}$$

$$F_1 = C_1 \sum_{j \in V_c^a} \sum_{k \in K^a} (x_{0j}^{wk} - z^k) \tag{21}$$

$$F_2 = C_2 \sum_{w \in W} \sum_{j \in \{0\} \cup V_c} \sum_{i \in \{0\} \cup V_c} \sum_{k \in K} F_{ijw}^k x_{ij}^{wk} \tag{22}$$

$$F_3 = \sum_{k \in K} \sum_{m \in M} \sum_{w \in W} \sum_{j \in \{0\} \cup V_c \cup VD} \sum_{i \in \{0\} \cup V_c \cup VD} t_{ijw}^k u_{mw}^k Q_{ijw}^k \tag{23}$$

$$F_4 = \sum_{i \in \{0\} \cup V_c \cup VD} \sum_{m \in M} \sum_{w \in W} \sum_{k \in K} v_{im}^{wk} \tag{24}$$

$$F_5 = C_4 \sum_{w \in W} \sum_{j \in \{0\} \cup V_c} \sum_{i \in \{0\} \cup V_c} \sum_{k \in K} \left(F_{ijw}^k x_{ij}^{wk} / \widehat{o} + at_{ijw}^k x_{ij}^{wk} \right) \tag{25}$$

s. t.

$$\sum_{i \in VD \cup V_c^a} \sum_{j \in V_c^a} \sum_{w \in W} x_{ij}^{wk} q_{jm}^w \leq Q^k, \forall k \in K^a \tag{26}$$

$$\sum_{i \in \{0\} \cup V_c^a} \sum_{j \in V_c^a} \sum_{w \in W} x_{ij}^{wk} q_{jm}^w \leq Cap, \forall k \in K^a \tag{27}$$

$$|K^a| + \sum_{j \in V_c^a} \sum_{k \in K^a} \sum_{w \in Q} x_{0j}^{wk} \leq |K| \tag{28}$$

$$\sum_{j \in \{0\} \cup V_c^a} \sum_{k \in K} \sum_{w \in W} x_{ij}^{wk} = 1, \forall i \in V_c^a \tag{29}$$

$$\sum_{i \in \{0\} \cup V_c^a} \sum_{k \in K} \sum_{w \in W} x_{ij}^k = 1, \forall j \in V_c^a \tag{30}$$

$$\sum_{i \in \{0\} \cup V_c^a \cup VD} \sum_{w \in W} x_{ij}^{wk} = \sum_{i \in \{0\} \cup V_c^a} \sum_{w \in W} x_{ji}^{wk}, \forall j \in V_c^a, \forall k \in K \tag{31}$$

$$\sum_{j \in V_c^a} \sum_{w \in W} x_{0j}^{wk} = \sum_{i \in V_c^a} \sum_{w \in W} x_{i0}^{wk} \leq 1, \forall k \in K^a \tag{32}$$

$$\sum_{k \in K^a} \sum_{j \in V_c^a} \sum_{w \in W} x_{ij}^{wk} = 1, \forall i \in VD \tag{33}$$

$$\sum_{k \in K^a} \sum_{i \in \{0\} \cup V_c^a} \sum_{w \in W} x_{ij}^{wk} = 0, \forall i \in VD \tag{34}$$

$$\sum_{k \in K} \sum_{i \in VD \cup V_c^a} \sum_{w \in W} x_{i0}^{wk} = \sum_{k \in K^a} \sum_{i \in V_c^a} \sum_{w \in W} x_{0j}^{wk} + |K^a| \tag{35}$$

$$T_{qi}^a \leq T_{end}, \forall i \in V_q^a \tag{36}$$

$$T_s^{ak} + \sum_{i \in \{0\} \cup V_c^a \cup VD} \sum_{j \in \{0\} \cup V_c^a} \sum_{w \in W} t_{ijw}^k x_{ij}^{wk} + \sum_{i \in \{0\} \cup V_c^a \cup VD} \sum_{j \in V_c^a} \sum_{w \in W} q_{jm}^w TS_j x_{ij}^{wk} \leq T_f, \forall k \in K \tag{37}$$

$$\sum_{i \in S} \sum_{j \in S} \sum_{w \in W} x_{ij}^{wk} \leq |S| - 1, |S| = \sum_{j \in V_c^a} \sum_{w \in W} x_{ij}^{wk}, \forall k \in K \tag{38}$$

$$\sum_{i \in \{0\} \cup V_c} \sum_{j \in V_c} \sum_{w \in W} d_{ij} x_{ij}^{wk} \leq vT_k, \forall k \in K \tag{39}$$

$$x_{ij}^{wk} \in \{0, 1\}, \forall i, j \in V_c^a, \forall k \in K, w \in W \tag{40}$$

$$z^k \in \{0, 1\}, \forall k \in K^a \tag{41}$$

Equations (26) and (27) represent the vehicle load constraint. Equation (28) is the number of delivery vehicles constraint. Equations (29) and (30) denote that the customer has and will only be served once. Equation (31) means that the vehicle leaves the customer's point after service. Equation (32) indicates that each vehicle serves only one path and that the vehicle departs from the distribution center and returns to the distribution center after completing the order. Equations (33) and (34) mean that only one vehicle leaves the virtual distribution point and does not return. Equation (35) indicates that all vehicles performing a distribution task return to the distribution center after completing all tasks. Equation (36) indicates that the time to process order changes does not exceed the order deadline of the distribution center. Equation (37) represents the vehicle returns to the distribution center at a time no later than the operating deadline. Equation (38) eliminates the sub-loop constraint. Equation (39) is the delivery time constraint. Equations (40) and (41) are the decision variable attributes.

3. Problem Solution and Algorithm Design

3.1. Problem Solution

Step 1: Determine the initial distribution path of the vehicles. The vehicles all depart from the distribution center. According to the pre-departure order information, using the AH-NOA based on k-means clustering to determine the initial plan.

Step 2: Determine whether a dynamic event occurs according to the rolling time domain. If it does, skip to Step 3, otherwise, end.

Step 3: According to the event occurrence time, the vehicle driving path before the dynamic event occurs is derived.

Step 4: Dynamic events occur and relevant order information is updated.

Step 5: Problem transformation. Set up a virtual distribution center based on the location information of vehicles in transit. Integrate undelivered orders and new orders into a new order requirement. Transforming the single-center DVRP into a new multi-center SVRP.

Step 6: Determine the vehicle path after the dynamic event occurs. Based on the new problem obtained in Step 5, the AH-NOA is used to solve it. Use the vehicle path obtained in Step 3 to replace the path between the distribution center and the virtual distribution center. Skip to Step 2.

3.2. The AH-NOA Design Based on K-Means Clustering Algorithm

3.2.1. K-Means Clustering Algorithm

The k-means clustering analysis of customers can reduce the decay of commodity quality and increase customer satisfaction. Because of the integrated consideration of the commodity's temperature and customer location, it can make the solution more realistic.

Suppose the location coordinates of the customer (i) and demand temperature interval of fresh agricultural products are $(x_i, y_i), (lp_i, up_i)$, respectively. The clustering distance between customers i and j is $d_{ij}^c = |x_i - x_j| + |y_i - y_j| + tr|lp_i - lp_j| + tr|up_i - up_j|$ [39]. tr denotes the conversion factor between distance and temperature. The specific steps are as follows:

- (1) Import customer location, commodity temperature and tr .
- (2) The products are divided into several temperature intervals according to their temperature characteristics while taking into account the customer's freshness demands.
- (3) The initial number of clustering units is determined as qc based on the k-means clustering algorithm, and the qc initial clustering centers are selected in each temperature control interval.
- (4) Under different temperature intervals, calculate the distance from the customer point to the cluster center and assign the clustered customers to the closest clustered unit.
- (5) Add the number of clustering units as $qc = qc + 1$ and generate new clustering units and centers.
- (6) Repeat (4) and (5) until the cluster centers no longer change.

3.2.2. Nutcracker Optimizer Algorithm

The nutcracker optimizer algorithm is a swarm intelligence optimization algorithm proposed in 2023. In the summer and autumn, nutcrackers search for food at random and choose to transport the better food to a storage area away from the foraging area. In winter and spring, nutcrackers go to storage areas to find food, which may not necessarily be their food but may also become the food of others. NOA simulates the foraging and food storage behavior based on the nutcracker’s ability to find food by relying on spatial memory mechanisms and achieves the goal of finding the best food.

Heuristic algorithms have good results for solving vehicle routing problems. However, it has a lack of robustness, and the results are easily influenced by a larger number of parameters. NOA has the advantages of high accuracy, fast computation and a small number of parameters. Moreover, due to the characteristics of the problem studied in this paper, a fast computational algorithm is more beneficial to solve the problem. However, there are also disadvantages such as the tendency to fall into local optima and slow convergence. The flow of this algorithm is shown in Figure 3.

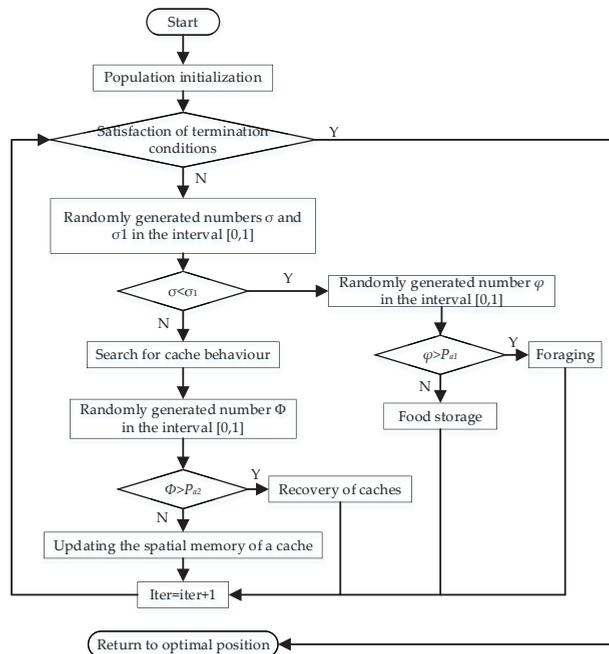


Figure 3. NOA flow chart.

3.2.3. Adaptive Hybrid Nutcracker Algorithm

To address the disadvantages of the NOA, an adaptive hybrid NOA is proposed. To ensure the diversity of the population, the CW algorithm was used to initialize some of the solutions. To keep the overall quality of the population, better nutcrackers are selected to generate new nutcrackers through the genetic operators and local search operators. A nutcracker winter food search probability and a local optimization avoidance probability are introduced into the decline function so that it can be adaptively adjusted with iterations. In this way, the algorithm’s convergence speed can be increased. It is possible to avoid falling into a local optimum to a certain extent. The AH-NOA flow chart is shown in Figure 4.

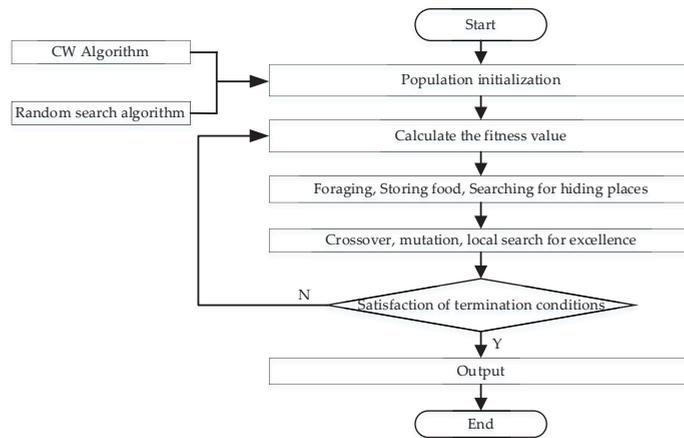


Figure 4. AH-NOA flow chart.

The specific steps of the AH-NOA are as follows:

- (1) Initialize the population. To increase the diversity of solutions, the initial population is generated in two ways: CW algorithm and random search.
- (2) Adaptability function. The total cost is used as the fitness function of the algorithm
- (3) Parameter iteration. Pa_1 controls the search direction in the storage phase. Pa_2 determines the swapping probability between the cache search and recovery phases. δ avoids getting stuck in a local optimum. The Pa_1 , Pa_2 and δ are improved to adaptively change with the algorithm running using the following equations:

$$Pa_1 = (iter_max - iter) / iter_max \tag{42}$$

$$Pa_2 = 1 / (1 - 1.5(\log_2(1/iter))) \tag{43}$$

$$\delta = \frac{iter_max - iter + 1}{\sqrt{\exp(-iter^4 / iter_max^3)}} \cdot 0.05 \tag{44}$$

- (4) Movement strategy. Each nutcracker performed foraging, storing food, searching for caches and obtaining food behaviors according to Equations (45)–(51), respectively, based on its current food search status. $\vec{X}_k^{\rightarrow iter+1}$ denotes the position of the k nutcracker in the $iter$ generation. U_j and L_j denote the upper and lower bounds of the nutcracker’s position at the point j . γ is a random number generated from a levy flight. X_{Mj}^{iter} denotes the average position of all nutcrackers at point j in the $iter$ generation. A, B and C are three nutcrackers randomly selected from the population to facilitate the search for high-quality food sources. τ_1, τ_2, r and r_1 are real numbers randomly generated in the interval $[0, 1]$. μ is randomly generated in the interval $[0, 1]$ based on a normal distribution.

$$\vec{X}_k^{\rightarrow iter+1} = \begin{cases} X_{kj}^{iter} & \tau_1 < \tau_2 \\ X_{Mj}^{iter} + \gamma(X_{Aj}^{iter} - X_{Bj}^{iter}) + \mu(r^2 U_j - L_j) & \tau_1 \geq \tau_2, iter < iter_max/2 \\ X_{Cj}^{iter} + \mu(X_{Aj}^{iter} - X_{Bj}^{iter}) + \mu(r_1 < \delta)(r^2 U_j - L_j) & otherwise \end{cases} \tag{45}$$

$$\vec{X}_k^{\rightarrow iter+1} = \begin{cases} \vec{X}_k^{\rightarrow iter} + \mu \left(\vec{X}_{best}^{\rightarrow iter} - \vec{X}_k^{\rightarrow iter} \right) |\lambda| + r_1 \left(\vec{X}_A^{\rightarrow iter} - \vec{X}_B^{\rightarrow iter} \right) & \tau_1 < \tau_2 \\ \vec{X}_{best}^{\rightarrow iter} + \mu \left(\vec{X}_A^{\rightarrow iter} - \vec{X}_B^{\rightarrow iter} \right) & \tau_1 < \tau_3 \\ \vec{X}_{best}^{\rightarrow iter} & otherwise \end{cases} \quad (46)$$

$$RP_{k,1}^{\rightarrow iter} = \begin{cases} \vec{X}_k^{\rightarrow iter} + \alpha \cos(\theta) \left(\vec{X}_A^{\rightarrow iter} - \vec{X}_B^{\rightarrow iter} \right) + \alpha RP & \theta = \pi/2 \\ \vec{X}_k^{\rightarrow iter} + \alpha \cos(\theta) \left(\vec{X}_A^{\rightarrow iter} - \vec{X}_B^{\rightarrow iter} \right) & otherwise \end{cases} \quad (47)$$

$$RP_{k,2}^{\rightarrow iter} = \begin{cases} \vec{X}_k^{\rightarrow iter} + \left(\alpha \cos(\theta) \left(\left(\vec{U} - \vec{L} \right) \tau_3 + \vec{L} \right) + \alpha RP \right) \vec{U}_2 & \theta = \pi/2 \\ \vec{X}_k^{\rightarrow iter} + \alpha \cos(\theta) \left(\left(\vec{U} - \vec{L} \right) \tau_3 + \vec{L} \right) \vec{U}_2 & otherwise \end{cases} \quad (48)$$

λ is a random number generated from a levy flight. X_{best}^{iter} is the position of the current optimal nutcracker. τ_3 is a randomly generated real number in the interval $[0, 1]$. l is the diversity factor of the solution linearly decreasing from 1 to 0. $(RP_{k,1}^{\rightarrow iter}, RP_{k,2}^{\rightarrow iter})$ denotes the location of the k nutcracker storing food in generation $iter$. $\vec{U} = (U_1, U_2, \dots, U_j)$ and $\vec{L} = (L_1, L_2, \dots, L_j)$ denote the range of regions where the nutcracker stores food. RP is a random location. α avoids the algorithm falling into a local optimum and adjusts according to Equation (49):

$$\alpha = \begin{cases} (1 - iter/iter_max)^{2iter/iter_max} & r_1 > r_2 \\ (iter/iter_max)^{2/iter} & otherwise \end{cases} \quad (49)$$

$$\vec{X}_k^{\rightarrow iter+1} \begin{cases} \vec{X}_k^{\rightarrow iter} & \tau_7 < \tau_8, \tau_3 < \tau_4 \text{ or } \tau_5 < \tau_6 \\ \vec{X}_k^{\rightarrow iter} + r_1 \left(\vec{X}_{best}^{\rightarrow iter} - \vec{X}_k^{\rightarrow iter} \right) + r_2 \left(RP_{k,1}^{\rightarrow iter} - \vec{X}_k^{\rightarrow iter} \right) & \tau_7 < \tau_8, \tau_3 \geq \tau_4 \\ \vec{X}_k^{\rightarrow iter} + r_1 \left(\vec{X}_{best}^{\rightarrow iter} - \vec{X}_k^{\rightarrow iter} \right) + r_2 \left(RP_{k,2}^{\rightarrow iter} - \vec{X}_k^{\rightarrow iter} \right) & \tau_7 < \tau_8, \tau_5 \geq \tau_6 \end{cases} \quad (50)$$

$$\vec{X}_k^{\rightarrow iter+1} = f^{-1} \left(\min \left(f \left(\vec{X}_k^{\rightarrow iter} \right), f \left(RP_{k,1}^{\rightarrow iter} \right), f \left(RP_{k,2}^{\rightarrow iter} \right) \right) \right) \quad (51)$$

$r_2, \tau_4, \tau_5, \tau_6, \tau_7$ and τ_8 are a randomly generated real number in the interval $[0, 1]$.

Suppose a vehicle departs from distribution center 0 and needs to make a delivery to customer points 1 and 2. X_1^{iter} is the position of the current nutcracker 1 and X_2^{iter} is the position of the current nutcracker 2. If $X_1^{iter} \leq X_2^{iter}$, $x_{12}^{wk} = 1$, $x_{21}^{wk} = 0$ and the order of vehicle visits is 0-1-2-0. Otherwise, $x_{12}^{wk} = 0$, $x_{21}^{wk} = 1$ and the order of vehicle visits is 0-2-1-0.

(5) Crossover, mutation and local search behavior. Incorporating the crossover and mutation operators from the genetic algorithm into the NOA can effectively overcome the NOA's lack of global search capability. The key to enhancing the global search ability is to enhance the diversity of the population, which both genetic operators can achieve. Therefore, the genetic operator is embedded in the NOA to expand the solution space and improve the global search capability.

① Crossover operations

Multiple crossovers can avoid individuals from converging prematurely, but too many crossovers will destroy some better solutions. In this paper, according to the roulette strategy, two crosses, OA and OB , are selected. After the two individuals have been crossed,

the recurring points are deleted, resulting in two new nutcrackers. The process is shown in Figure 5.

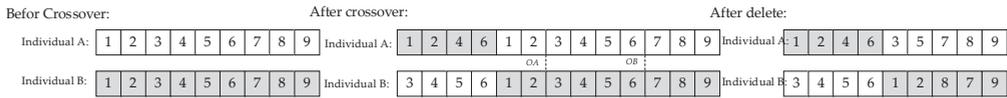


Figure 5. Crossover operation.

② Mutation operation

Select two random positions and directly exchange them to complete the mutation operation, as shown in Figure 6.



Figure 6. Mutation operation.

③ Local search operations

Remove some similar location points based on similarity by using the destruction operator. Under load capacity and time constraints, removed points are inserted back into the total cost. The similarity is calculated as $R_{ij} = 1 / (S_{ij} + d_{ij} / d_{ij_max})$, where $S_{ij} = 1$ means that i and j are not on the same path and $S_{ij} = 0$ means that i and j are on the same path, $d_{ij_max} = \max\{j \in V_c, i \neq j | d_{ij}\}$.

(6) Algorithm termination. The algorithm is terminated when the maximum number of iterations has been reached.

3.2.4. Time Complexity Analysis of AH-NOA

In the base NOA, if the number of populations is n , then when simulating the foraging and storage behavior, the time magnitude is n . In the cache-search and recovery strategy, generating the RP matrix needs at most n times and the time magnitude is n . Each nutcracker has to search for the best food at most $iter_max$ times. The time magnitude is $iter_max$. In summary, the time complexity of the NOA is $O((n + n) \cdot iter_max)$. AH-NOA adds crossover, mutation and local search behavior to the NOA, which has a time dimension of n . The time complexity of the AH-NOA is $O((n + n) \cdot iter_max + n)$.

3.2.5. Algorithm Effectiveness Verification

To verify the effectiveness of the algorithm improvements, a comparison experiment between the basic NOA and AH-NOA was made. Customer location coordinates and service demands were obtained by improving the Solomon dataset [40]. At the same time, the temperature intervals and prices for fresh agricultural products are added, and 12 examples were selected for analysis and comparison. The main parameters are set as follows: $pop_size = 50$, $iter_max = 200$, $p_c = 0.8$, $p_m = 0.1$, $tr = 1.2$. Due to the high information level of the order, the communication time between the delivery personnel and the customer is greatly reduced. To simplify the calculation, assume $TS_i = 0$. The characteristics of the example data set are shown in Table 3. Each set was calculated 20 times and the best values were chosen, as shown in Table 4. Where TD denotes total cost, VL denotes loss of value, VN denotes the number of vehicles and CEC denotes the carbon emissions cost. $Gap_c = (TD_{NOA} - TD_{AH-NOA}) / TD_{NOA}$, $Gap_g = (CEC_{NOA} - CEC_{AH-NOA}) / CEC_{NOA}$.

Table 3. Dataset characteristics.

Case No.	Customer Numbers	Temperature Control Interval	Price
1–3	40	(−5 °C)–(0 °C)	10
		(1 °C)–(6 °C)	7
4–6	60	(−5 °C)–(0 °C)	10
		(−13 °C)–(−8 °C)	15
7–9	80	(−5 °C)–(0 °C)	10
		(1 °C)–(6 °C)	7
10–12	100	(−13 °C)–(−8 °C)	15
		(−5 °C)–(0 °C)	10
		(1 °C)–(6 °C)	7
		(−13 °C)–(−8 °C)	15

Table 4. Comparison of algorithm solution results.

No.	Name	AH-NOA				NOA				Gap_c	Gap_g
		TD	VL	CEC	VN	TD	VL	CEC	VN		
1	C101	964	183	203	5	1716	205	235	5	43.82%	13.62%
2	R101	1383	192	341	5	1713	185	428	6	19.26%	20.33%
3	RC101	1812	245	403	7	2540	272	571	9	28.66%	29.42%
4	C102	1746	379	371	6	2150	364	495	8	18.79%	25.05%
5	R102	2226	317	547	8	2896	270	732	10	23.14%	25.27%
6	RC102	3308	422	724	10	3983	357	905	14	16.95%	20.00%
7	C103	2900	398	403	7	3705	407	561	9	21.73%	28.16%
8	R103	3340	299	486	7	4178	325	634	8	20.06%	23.34%
9	RC103	4222	391	632	9	5637	346	893	13	25.10%	29.23%
10	C104	3756	511	545	9	5986	559	972	13	37.25%	43.93%
11	R104	4175	420	705	10	4522	391	860	12	7.67%	18.02%
12	RC104	4943	446	638	9	6374	394	921	13	22.45%	30.73%
	Ave	2898	350	500	8	3783	340	684	10	24%	25.59%
		<i>t</i> -test				−6.035		−6.81	−6.325		
		<i>p</i> -value				0.000059		0.000019	0.000038		

Table 4 shows that the AH-NOA outperforms the NOA in terms of *TD*, *CEC* and *VN*. In the pathway optimization model with the lowest total cost, the AH-NOA obtained better results than the NOA. Total costs were optimized by a maximum of 43.82% and an average of 24%. The carbon emissions costs were optimized by a maximum of 43.93% and an average of 25.59%. The *t*-test and *p*-value analysis showed that the difference between the calculated *TD*, *CEC* and *VN* for the AH-NOA solution and the NOA was significant. Therefore, the AH-NOA is an effective improvement to the NOA.

4. Example Analysis

4.1. Example Design

Before delivery begins, the distribution center receives orders from 60 customers, which are located in different parts of the city. Each customer has different demands and temperature intervals. The distribution center information is shown in Table 5. The specific customer demands are shown in Table 6. Customer location, demand and commodity temperature attributes are randomly generated using the *rand()* function according to objective reality. $Cap = 200$ kg, $n_t = 32$. The hire cost is $C_1 = 60$ CNY. The average vehicle speed is 50 km/h. Assume a unit fuel emission factor of 2.63 kg/L. Referring to the average carbon emission trading price on the Beijing, Shanghai and Guangdong exchanges on 19 April 2019, set the unit carbon emission price to 0.1 CNY/kg.

Table 5. Distribution center information.

X/km	Y/km	T_s	T_f
0	0	8:00	16:00

Table 6. Customer information.

No.	X/km	Y/km	Demand/kg	Temperature Control Interval	No.	X/km	Y/km	Demand/kg	Temperature Control Interval
1	-27	-25	8	(-5 °C)-(0 °C)	31	8	15	23	(-5 °C)-(0 °C)
2	31	3	11	(1 °C)-(6 °C)	32	-18	-10	11	(1 °C)-(6 °C)
3	-4	-29	25	(1 °C)-(6 °C)	33	29	-24	3	(-5 °C)-(0 °C)
4	30	-18	12	(-13 °C)-(-8 °C)	34	25	20	16	(-13 °C)-(-8 °C)
5	26	-1	2	(-13 °C)-(-8 °C)	35	21	14	8	(-13 °C)-(-8 °C)
6	-39	-26	21	(-13 °C)-(-8 °C)	36	-4	-11	11	(-13 °C)-(-8 °C)
7	-23	-23	10	(-5 °C)-(0 °C)	37	34	-12	24	(1 °C)-(6 °C)
8	-22	2	11	(-13 °C)-(-8 °C)	38	1	-11	19	(1 °C)-(6 °C)
9	3	-33	4	(-13 °C)-(-8 °C)	39	13	8	17	(1 °C)-(6 °C)
10	9	33	23	(-5 °C)-(0 °C)	40	20	-27	13	(-5 °C)-(0 °C)
11	6	12	25	(-5 °C)-(0 °C)	41	31	-39	18	(1 °C)-(6 °C)
12	7	1	15	(-13 °C)-(-8 °C)	42	-20	10	8	(-13 °C)-(-8 °C)
13	31	0	2	(1 °C)-(6 °C)	43	-5	-36	3	(-13 °C)-(-8 °C)
14	-9	-29	16	(-13 °C)-(-8 °C)	44	-39	17	8	(-13 °C)-(-8 °C)
15	-27	1	2	(-13 °C)-(-8 °C)	45	-32	3	17	(1 °C)-(6 °C)
16	4	14	17	(-5 °C)-(0 °C)	46	30	-4	25	(1 °C)-(6 °C)
17	-9	22	19	(-5 °C)-(0 °C)	47	23	28	6	(1 °C)-(6 °C)
18	-1	-26	22	(-5 °C)-(0 °C)	48	-22	5	24	(-13 °C)-(-8 °C)
19	34	20	18	(1 °C)-(6 °C)	49	21	-22	4	(-13 °C)-(-8 °C)
20	11	-15	24	(1 °C)-(6 °C)	50	18	28	25	(1 °C)-(6 °C)
21	-32	-32	12	(-13 °C)-(-8 °C)	51	9	16	13	(-5 °C)-(0 °C)
22	2	34	5	(-13 °C)-(-8 °C)	52	-5	-30	15	(1 °C)-(6 °C)
23	10	9	18	(1 °C)-(6 °C)	53	-9	-1	19	(-13 °C)-(-8 °C)
24	-24	-32	22	(-13 °C)-(-8 °C)	54	-29	30	7	(1 °C)-(6 °C)
25	-30	-18	12	(-5 °C)-(0 °C)	55	-38	-22	7	(-5 °C)-(0 °C)
26	9	-9	12	(-13 °C)-(-8 °C)	56	-16	-37	13	(-13 °C)-(-8 °C)
27	-18	28	11	(-13 °C)-(-8 °C)	57	6	-24	9	(1 °C)-(6 °C)
28	-33	21	6	(1 °C)-(6 °C)	58	17	1	7	(-13 °C)-(-8 °C)
29	16	-37	22	(-13 °C)-(-8 °C)	59	-37	40	11	(1 °C)-(6 °C)
30	-40	6	18	(-5 °C)-(0 °C)	60	-14	38	19	(1 °C)-(6 °C)

4.2. Example Solution and Results Analysis

4.2.1. Temperature Sensitivity Analysis

Fresh commodities have different sensitivity coefficients at different temperatures, which leads to differences in the value loss. Using the example R108, the temperature control cost and value loss under the temperature range (1 °C)-(6 °C), (-5 °C)-(0 °C) and (-13 °C)-(-8 °C) were calculated. The results are shown in Figure 7.

Figure 7 shows that the temperature control cost and value loss both change with temperature, but they are both paradoxical. When the temperature is at the left end of the interval, the value loss is the smallest and the temperature control cost is the largest. Conversely, when the temperature is the right endpoint of the interval, the commodity is at a higher temperature. The loss is the largest and the temperature control cost is the smallest. This indicates that under a constant external temperature, the value loss gradually increases and the temperature control cost gradually decreases with increasing temperature. The value loss and temperature control costs were different in the increase or decrease during the temperature change. Based on the Pareto optimal principle [41], the optimal temperature in the temperature control intervals (1 °C)-(6 °C), (-5 °C)-(0 °C) and (-13 °C)-(-8 °C) are 3 °C, -3 °C and -11 °C, respectively. When the temperature is at its optimum, adjusting the temperature up or down does not result in an optimum.

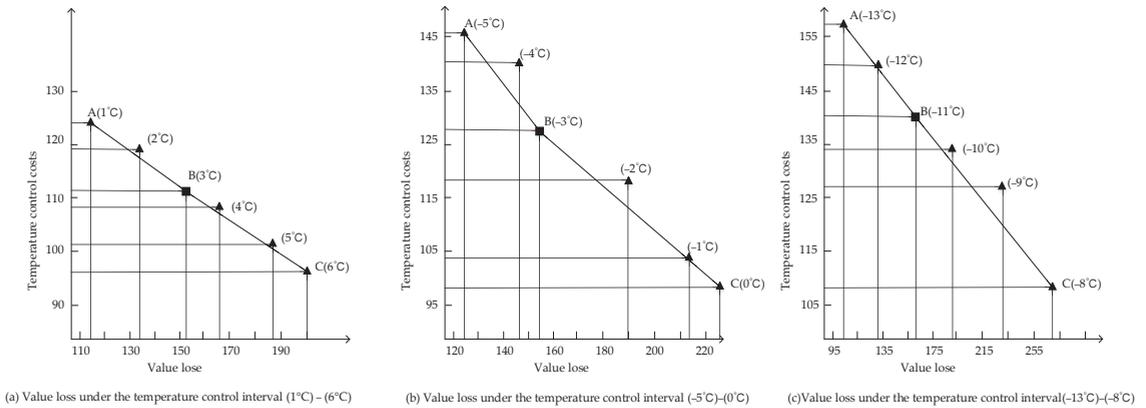


Figure 7. Fresh commodity value loss and temperature control paradox diagram.

4.2.2. Carbon Emissions Impact on GCVRPDD

To research the impact of carbon emission cost on the dynamic vehicle routing problem, this paper uses the AH-NOA to solve the dynamic vehicle routing problem, considering carbon emissions and not considering carbon emissions, respectively. The algorithm parameters used are the same. The relevant parameters are $C_2 = 5.2$ CNY/km and $f_{kw} = 0.12$ L/km. The algorithm was programmed using Matlab R2016a and implemented running on Windows 10, 8 G and 2.80 GHz.

(1) Results and Analysis of the Initial Scheme

(a) Results of Initial Scheme

The AH-NOA was used to solve the vehicle path problem with and without considering carbon emissions. The initial distribution path is shown in Figure 8. Figure 8a shows that eight vehicles were dispatched. Figure 8b shows that nine vehicles were dispatched. There are some differences in the vehicle paths for the two distribution schemes, which are mainly due to whether or not carbon costs are taken into account. The specific routes for the two distribution schemes are shown in Tables 6 and 7, respectively. The stars in Figure 8 represent distribution centers and the dots represent customer points.

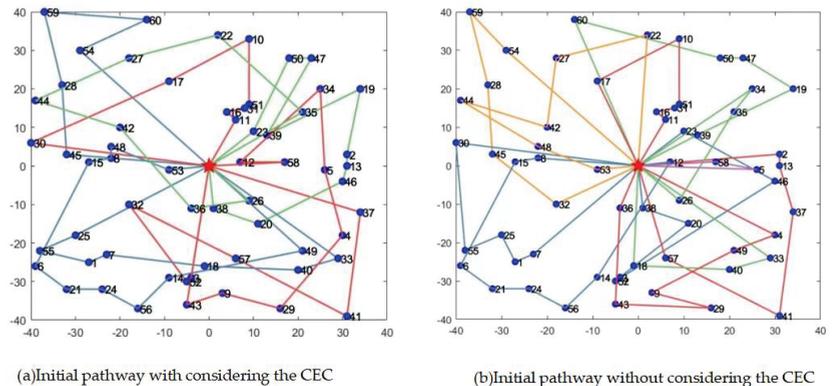


Figure 8. Vehicle routing.

Table 7. Initial route considering the carbon emission costs.

Vehicle Serial No.	Pathway	Temperature Control Interval
1	0->45->28->59->60->54->0	(1 °C)-(6 °C)
2	0->3->52->32->57->41->37->0	
3	0->23->50->47->39->19->2->13->46->20->38->0	
4	0->33->40->18->7->1->55->25->0	(-5 °C)-(0 °C)
5	0->11->16->31->51->10->17->30->0	
6	0->49->14->56->24->21->6->15->8->48->53->0	(-13 °C)-(-8 °C)
7	0->43->9->29->4->5->34->12->58->0	
8	0->26->36->42->44->27->22->35->0	

(b) Results Comparison

Comparing Tables 7 and 8, it can be seen that there are significant differences in the paths of the two schemes. There is a significant difference in the order of delivery customers. The carbon emission cost, value loss and total cost have obvious changes, as shown in Table 9. As can be seen from Table 9, the distribution scheme, considering the carbon emission cost, is CNY 119 more in value loss than without considering. However, the carbon emission cost decreased by 9.9%, while the transportation cost and total cost also decreased by 9.7% and 10.1%, respectively. This is largely attributable that the distribution scheme, considering carbon emission costs, needs to provide priority delivery to customers with larger loads. Although this causes an increase in value loss, it achieves a lower carbon emission cost and total cost during the shelf life of the commodity. Therefore, distribution schemes considering the carbon emission costs must reduce carbon emissions while minimizing the total cost.

Table 8. Initial route without considering the carbon emission costs.

Vehicle Serial No.	Pathway	Temperature Control Interval
1	0->38->20->3->52->46->23->0	(1 °C)-(6 °C)
2	0->2->13->37->41->57->0	
3	0->39->19->47->50->60->0	
4	0->32->45->28->59->54->0	
5	0->7->1->25->55->30->0	(-5 °C)-(0 °C)
6	0->11->16->31->51->10->17->0	
7	0->33->40->18->0	
8	0->8->15->6->21->24->56->14->12->0	(-13 °C)-(-8 °C)
9	0->36->43->29->9->49->4->0	

Table 9. Cost comparison of two distribution schemes.

Distribution Scheme	VN	CEC	VL	TrC	TC
Considering the carbon emission costs	8	599	318	370	2535
Without considering the carbon emission costs	9	665	237	410	2821

(2) Adjustment and Comparison Analysis of Pathway under Real-time Information

During the vehicle operation, the dispatch center receives information on all dynamic events, as shown in Table 10. To simplify the calculation, it is assumed that dynamic events only occur within a certain interval.

Dynamic events occur at [8:45, 9:00]. According to the dynamic adjustment strategy, the current dynamic event is adjusted in the next time slice based on the rolling time domain division. Thus, the dynamic event handling time slice is at [9:15, 9:30]. Based on the updated demand information, the AH-NOA is used to solve the GCVRPDD separately. The distribution route scheme is shown in Figure 9. As can be seen in Figure 9, vehicles

complete their tasks according to the new route arrangement. The specific routes are shown in Table 11.

Table 10. Dynamic event information.

No.	Customer No.	X/km	Y/km	Dynamic Time	Dynamic Event	Temperature Control Interval
1	61	27	-2	8:46	New customer point demand 19 kg	(1 °C)–(6 °C)
2	62	-22	-15	8:49	New customer point demand 18 kg	(-13 °C)–(-8 °C)
3	63	-22	-14	8:50	New customer point demand 10 kg	(-13 °C)–(-8 °C)
4	50	18	28	8:52	Demand reduced by 10 kg	(1 °C)–(6 °C)
5	28	-33	21	8:58	Demand increased by 5 kg	(1 °C)–(6 °C)

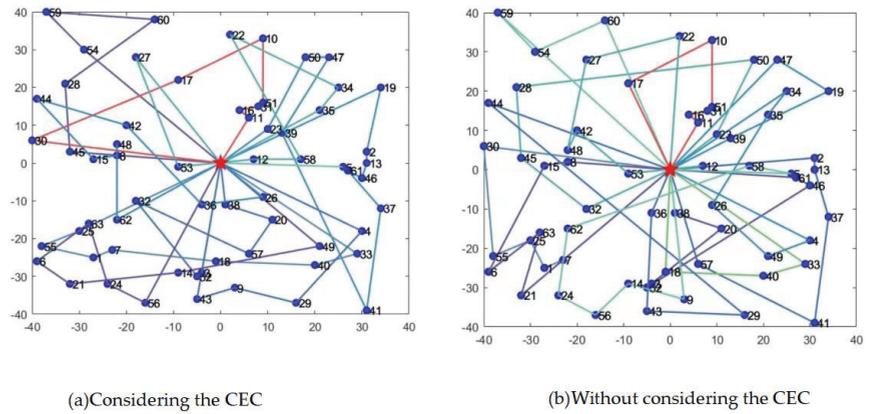


Figure 9. Distribution paths after dynamic events.

Table 11. Specific distribution routes after change under the two distribution schemes.

Distribution Scheme	Vehicle Serial No.	Pathway	Temperature Control Interval
Considering the carbon emission costs	1	0->28->60->59->54->0	(1 °C)–(6 °C)
	2	0->3->52->32->57->20->38->0	
	3	0->23->50->47->39->41->37->61->46->13->2->19->0	
	6	0->49->14->21->6->63->24->56->0	(-13 °C)–(-8 °C)
	7	0->43->9->29->4->0	
	8	0->26->36->42->44->15->8->48->62->0	
	9	0->35->34->22->58->12->53->27->0	
	10	0->5->0	
	1	0->38->20->3->52->46->61->0	(1 °C)–(6 °C)
	2	0->2->13->37->41->57->0	
Without considering the carbon emission costs	3	0->39->19->47->0	
	4	0->32->45->28->50->23->0	
	13	0->60->54->59->0	
	8	0->8->15->6->63->21->0	(-13 °C)–(-8 °C)
	9	0->36->43->29->44->0	
	10	0->34->35->26->49->4->0	
11	0->22->27->48->42->53->12->0		
12	0->5->58->62->24->56->14->9->0		

Comparing Figure 9a,b, it can be seen that the two distribution schemes are completely different. There are also significant differences in the costs of each distribution. This is shown in Table 12. It can be seen from Table 12 that the distribution scheme considering carbon emissions increased in value loss by CNY 105 and decreased the transportation cost and total cost by 6.4% and CNY 36, respectively. Although there is a significant value loss, this loss is still within the goods freshness and the distribution scheme considering carbon emissions is 6.3% lower in carbon emissions, which is good for the positive corporate social image and corporate future development. The examples used are universally applicable because they are randomly generated based on actual situations. The above comparison can demonstrate that considering carbon emissions in the dynamic vehicle path can reduce carbon emissions and total cost. The AH-NOA can effectively solve the GCVRPDD. By comparing the initial route and adjusted costs of the two schemes, it was found that the distribution scheme considering carbon emissions had a greater change in the number of vehicles, carbon emission costs, transport costs and total costs. This is shown in Table 13.

Table 12. Cost comparison of two distribution schemes after dynamic events.

Distribution Scheme	VN	CEC	VL	TrC	TC
Considering the carbon emission costs	10	667	320	412	2823
Without considering the carbon emission costs	13	712	215	440	2859

Table 13. The extent of cost changes before and after distribution program adjustments.

Distribution Scheme	Increased Number of Vehicles	Increased CEC	Increased TrC	Increased TC
Considering the carbon emission costs	2	68	42	288
Without considering the carbon emission costs	1	47	30	38

5. Conclusions

This paper researches GCVRPDD based on the AH-NOA. Firstly, through the analysis of the problem, using virtual customer points to transform dynamic vehicle paths into SVRP based on the rolling time domain, and construct a two-stage GCVRPDD model with the goals of minimizing vehicle dispatch costs, transport costs, temperature control costs, value loss and carbon emissions costs. Secondly, the AH-NOA is designed according to the model characteristics and improvements are made to reduce its disadvantages. The CW algorithm is added to initialize partial solutions to ensure population diversity. Better nutcrackers are selected to ensure the overall quality of the population by generating new nutcrackers. The decay function is introduced to improve the convergence speed of the algorithm and to avoid falling into a local optimum. The algorithm is compared with the original NOA to verify the effectiveness and reasonableness of the algorithm improvements. According to the algorithm and the model, it found the optimal temperature in three different temperature control intervals. Finally, the GCVRPDD is solved based on two different distribution schemes. The results show that the distribution scheme considering carbon emission costs, while higher in value loss, is lower in carbon emissions and total costs and has a significantly fewer number of vehicles. Considering the carbon emission cost in DVRP can lead to a significant increase in carbon emissions, transport costs and total costs compared to SVRP. This is because the quality of fresh agricultural products declines as the vehicle time in transit increases. To reduce the value loss of fresh agricultural products, companies need to generate more carbon emissions to meet customer demand, resulting in increased carbon emissions costs, transportation costs and total costs. However, with the emergence of diversity in consumer demand, it is more realistic to consider carbon emissions in dynamic delivery. At the same time, the AH-NOA adds a new solution method to dynamic vehicle path optimization.

Future research will focus on the effects of distribution vehicle temperature dynamics, customer time windows, customer satisfaction and other factors to make the problem more realistic.

Author Contributions: D.W.: Supervision, Methodology and Writing—original draft. R.Y.: Formal analysis, Validation, Software and editing. H.J.: Conceptualization, Funding acquisition. F.C.: Resources. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the China National Social Science Fund Project “Study on dynamic optimization of urban main and non-staple food reserve and supply system under abnormal conditions” (No. 22BGL274), and the Major Projects of the National Social Science Foundation of China (No. 22ZDA058).

Data Availability Statement: All experimental data in this paper comes from <https://people.idsia.ch/~luca/macsvrptw/problems/welcome.htm> (accessed on 10 March 2023).

Conflicts of Interest: The authors declare no conflict of interest.

References

- Hsu, C.-I.; Chen, W.-T.; Wu, W.-J. Optimal delivery cycles for joint distribution of multi-temperature food. *Food Control* **2013**, *34*, 106–114. [CrossRef]
- Gharehyakheh, A.; Krejci, C.C.; Cantu, J.; Rogers, K.J. A Multi-Objective Model for Sustainable Perishable Food Distribution Considering the Impact of Temperature on Vehicle Emissions and Product Shelf Life. *Sustainability* **2020**, *12*, 6668. [CrossRef]
- Wu, D.; Wu, C. Research on the Time-Dependent Split Delivery Green Vehicle Routing Problem for Fresh Agricultural Products with Multiple Time Windows. *Agriculture* **2022**, *12*, 793. [CrossRef]
- Zhang, B. The Optimization of Distribution Path of Fresh Cold Chain Logistics Based on Genetic Algorithm. *Comput. Intell. Neurosci.* **2022**, *2022*, 4667010. [CrossRef] [PubMed]
- Xiancheng, Z.; Kaijun, Z.; Li, W.; Changshi, L.; Xingbin, H. Review of green vehicle routing model and its algorithm in logistics distribution. *Syst. Eng.-Theory Pract.* **2021**, *41*, 213–230.
- Jin-liang, Z.; Chao, L. Research on Dynamic Distribution Vehicle Route Optimization under the Influence of Carbon Emission. *Chin. J. Manag. Sci.* **2022**, *30*, 184–194.
- Elgharably, N.; Easa, S.; Nassef, A.; El Damatty, A. Stochastic Multi-Objective Vehicle Routing Model in Green Environment with Customer Satisfaction. *IEEE Trans. Intell. Transp. Syst.* **2022**, *24*, 1337–1355. [CrossRef]
- Bruglieri, M.; Ferone, D.; Festa, P.; Pisacane, O. A GRASP with penalty objective function for the Green Vehicle Routing Problem with Private Capacitated Stations. *Comput. Oper. Res.* **2022**, *143*, 105770. [CrossRef]
- Xian-cheng, Z.; Yang, L.; Cai-hong, H.; Chang-shi, L.; Kun, Y. Multi-depot green vehicle routing model and its optimization algorithm with time-varying speed. *Control. Decis.* **2022**, *37*, 473–482.
- Li, J.; Zhang, J.H. Practice Vehicle routing problem with time windows based on carbon emissions and speed optimization. *Syst. Eng.-Theory Pract.* **2014**, *34*, 3063–3072.
- Cai, L.; Lv, W.; Xiao, L.; Xu, Z. Total carbon emissions minimization in connected and automated vehicle routing problem with speed variables. *Expert Syst. Appl.* **2021**, *165*, 113910. [CrossRef]
- Yin, F.; Zhao, Y. Optimizing vehicle routing via Stackelberg game framework and distributionally robust equilibrium optimization method. *Inf. Sci.* **2021**, *557*, 84–107. [CrossRef]
- Yang, J.; Jaillet, P.; Mahmassani, H. Real-time multivehicle truckload pickup and delivery problems. *Transp. Sci.* **2004**, *38*, 135–148. [CrossRef]
- Zhang, T.; Lai, P.Z.; He, Q.F.; Jin, Z.H. Optimization of Dynamic Vehicle Routing of Urban Distribution Based on the Real-time Information. *Syst. Eng.* **2015**, *33*, 58–64.
- Guo, X.; Samaranyake, S. Shareability network based decomposition approach for solving large-scale single school routing problems. *Transp. Res. Part C-Emerg. Technol.* **2022**, *140*, 103691. [CrossRef]
- Pan, W.; Liu, S.Q. Deep reinforcement learning for the dynamic and uncertain vehicle routing problem. *Appl. Intell.* **2023**, *53*, 405–422. [CrossRef]
- Kucuk, M.; Yildiz, S.T. Constraint programming-based solution approaches for three-dimensional loading capacitated vehicle routing problems. *Comput. Ind. Eng.* **2022**, *171*, 108505. [CrossRef]
- Aydinalp Birecik, Z.; Oezgen, D. An interactive possibilistic programming approach for green capacitated vehicle routing problem. *Neural Comput. Appl.* **2023**, *35*, 9253–9265. [CrossRef]
- Wang, S.; Mei, Y.; Zhang, M.; Yao, X. Genetic Programming With Niching for Uncertain Capacitated Arc Routing Problem. *IEEE Trans. Evol. Comput.* **2022**, *26*, 73–87. [CrossRef]
- Souza, I.P.; Boeres, M.C.S.; Moraes, R.E.N. A robust algorithm based on Differential Evolution with local search for the Capacitated Vehicle Routing Problem. *Swarm Evol. Comput.* **2023**, *77*, 101245. [CrossRef]

21. Eltoukhy, A.E.E.; Hussein, M.; Xu, M.; Chan, F.T.S. Data-driven Game-theoretic Model Based on Blockchain for Managing Resource Allocation and Vehicle Routing in Modular Integrated Construction. *Int. J. Prod. Res.* **2022**, *61*, 4472–4502. [CrossRef]
22. Song, W.; Yuan, S.; Yang, Y.; He, C. A Study of Community Group Purchasing Vehicle Routing Problems Considering Service Time Windows. *Sustainability* **2022**, *14*, 6968. [CrossRef]
23. Wu, L.-J.; Chen, Z.-G.; Chen, C.-H.; Li, Y.; Jeon, S.-W.; Zhang, J.; Zhan, Z.-H. Real Environment-Aware Multisource Data-Associated Cold Chain Logistics Scheduling: A Multiple Population-Based Multiobjective Ant Colony System Approach. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 23613–23627. [CrossRef]
24. Futalef, J.P.; Munoz-Carpintero, D.; Rozas, H.; Orchard, M.E. An online decision-making strategy for routing of electric vehicle fleets q. *Inf. Sci.* **2023**, *625*, 715–737. [CrossRef]
25. Vieira, B.S.; Ribeiro, G.M.; Bahiense, L. Metaheuristics with variable diversity control and neighborhood search for the Heterogeneous Site-Dependent Multi-depot Multi-trip Periodic Vehicle Routing Problem. *Comput. Oper. Res.* **2023**, *153*, 106189. [CrossRef]
26. Voigt, S.; Frank, M.; Fontaine, P.; Kuhn, H. Hybrid adaptive large neighborhood search for vehicle routing problems with depot location decisions. *Comput. Oper. Res.* **2022**, *146*, 105856. [CrossRef]
27. Abbaspour, S.; Aghsami, A.; Jolai, F.; Yazdani, M. An integrated queueing-inventory-routing problem in a green dual-channel supply chain considering pricing and delivery period: A case study of construction material supplier. *J. Comput. Des. Eng.* **2022**, *9*, 1917–1951. [CrossRef]
28. Wang, Y.; Ran, L.; Guan, X.; Fan, J.; Sun, Y.; Wang, H. Collaborative multicenter vehicle routing problem with time windows and mixed deliveries and pickups. *Expert Syst. Appl.* **2022**, *197*, 116690. [CrossRef]
29. Abdel-Basset, M.; Mohamed, R.; Jameel, M.; Abouhawwash, M. Nutcracker optimizer: A novel nature-inspired metaheuristic algorithm for global optimization and engineering design problems. *Knowl.-Based Syst.* **2023**, *262*, 110248.
30. Demir, E.; Bektas, T.; Laporte, G. A comparative analysis of several vehicle emission models for road freight transportation. *Transp. Res. Part D-Transp. Environ.* **2011**, *16*, 347–357. [CrossRef]
31. Suzuki, Y. A dual-objective metaheuristic approach to solve practical pollution routing problem. *Int. J. Prod. Econ.* **2016**, *176*, 143–153. [CrossRef]
32. Zhou, X.C.; Liu, C.S.; Zhou, K.J.; He, C.H.; Huang, X.B. Improved ant colony algorithm and modelling of time-dependent green vehicle routing problem. *J. Manag. Sci. China* **2019**, *22*, 57–68.
33. Ghare, P.M.; Schrader, G.H. A model for exponentially decaying inventory system. *Int. J. Prod. Res.* **1963**, *21*, 449–460.
34. Hsu, C.-I.; Hung, S.-F.; Li, H.-C. Vehicle routing problem with time-windows for perishable food delivery. *J. Food Eng.* **2007**, *80*, 465–475. [CrossRef]
35. Leng, L.; Zhang, J.; Zhang, C.; Zhao, Y.; Wang, W.; Li, G. Decomposition-based hyperheuristic approaches for the bi-objective cold chain considering environmental effects. *Comput. Oper. Res.* **2020**, *123*, 105043. [CrossRef]
36. Wang, J.; Liu, H.; Zhao, R. The optimization of cold chain operation based on fresh food safety. *Syst. Eng.-Theory Pract.* **2018**, *38*, 122–134.
37. Kilby, P.; Prosser, P.; Shaw, P. Dynamic VRPs: A Study of Scenarios. *Univ. Strathclyde Tech. Rep.* **1998**, *1*.
38. Montemanni, R.; Gambardella, L.M.; Rizzoli, A.E.; Donati, A. Ant colony system for a dynamic vehicle routing problem. *J. Comb. Optim.* **2005**, *10*, 327–343. [CrossRef]
39. Wang, Z. Delivering meals for multiple suppliers: Exclusive or sharing logistics service. *Transp. Res. Part E-Logist. Transp. Rev.* **2018**, *118*, 496–512. [CrossRef]
40. Solomon, M. Algorithms for the vehicle routing and scheduling problem with time window constraints. *Oper. Res.* **1987**, *35*, 254–265. [CrossRef]
41. Ishibuchi, H.; Setoguchi, Y.; Masuda, H.; Nojima, Y. Performance of Decomposition-Based Many-Objective Algorithms Strongly Depends on Pareto Front Shapes. *IEEE Trans. Evol. Comput.* **2017**, *21*, 169–190. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

MDPI AG
Grosspeteranlage 5
4052 Basel
Switzerland
Tel.: +41 61 683 77 34

Agriculture Editorial Office
E-mail: agriculture@mdpi.com
www.mdpi.com/journal/agriculture



Disclaimer/Publisher's Note: The title and front matter of this reprint are at the discretion of the Guest Editors. The publisher is not responsible for their content or any associated concerns. The statements, opinions and data contained in all individual articles are solely those of the individual Editors and contributors and not of MDPI. MDPI disclaims responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.



Academic Open
Access Publishing

[mdpi.com](https://www.mdpi.com)

ISBN 978-3-7258-2528-8