YOLO-NPK: A Lightweight Deep Network for Lettuce Nutrient Deficiency Classification Based on Improved YOLOv8 Nano †

Jordane Sikati 1 and Joseph Christian Nouaze 2,3, *

1 R&D Center, Guinee Biomedical Maintenance, Bentourayah, Conakry 3137, Guinea; jordanesikati@pusan.ac.kr
2 Department of Electronics Engineering, Pusan National University, Busan 46241, Republic of Korea
3 R&D Center, CAS Corporation, Headquarters, Yangju 11415, Republic of Korea

* Correspondence: krxsange@pusan.ac.kr; Tel.: +82-1094-011-957

† Presented at the 10th International Electronic Conference on Sensors and Applications (ECSA-10), 15–30 November 2023; Available online: https://ecsa-10.sciforum.net/.

Abstract: When it comes to growing lettuce, specific nutrients play vital roles in its growth and development. These essential nutrients include full nutrients (FN), nitrogen (N), phosphorus (P), and potassium (K). Insufficient or excess levels of these nutrients can have negative effects on lettuce plants, resulting in various deficiencies that can be observed in the leaves. To better understand and identify these deficiencies, a deep learning approach is employed to improve these tasks. For this study, YOLOv8 Nano, a lightweight deep network, is chosen to classify the observed deficiencies in lettuce leaves. Several enhancements to the baseline algorithm are made, the backbone is replaced with VGG16 to improve the classification accuracy, and depthwise convolution is incorporated into it to enrich the features while keeping the head unchanged. The proposed network, incorporating these modifications, achieved superior classification results with a top-1 accuracy of 99%. This method outperformed other state-of-the-art classification methods, demonstrating the effectiveness of the approach in identifying lettuce deficiencies. The objective of this research was to improve the baseline algorithm to complete the classification task with a top-1 accuracy above 85%, a FLOP inferior to 10G, and classification latency below 170 ms per image.

Keywords: lettuce nutrient deficiency; classification; deep learning; YOLOV8 Nano

1. Introduction

Lettuce (Lactuca sativa) is a widely cultivated leafy vegetable with significant economic and dietary importance. Adequate nutrient supply, particularly of nitrogen (N), phosphorus (P), and potassium (K), is essential for optimal lettuce growth and quality. Nitrogen is a primary component of chlorophyll and essential for photosynthesis. Nitrogen deficiency in lettuce results in stunted growth, pale leaves, and reduced leaf size, affecting the overall yield and nutritional content of lettuce, as well as its susceptibility to diseases [1]. Phosphorus is crucial for energy transfer in plants and plays a key role in root development. Lettuce plants deficient in phosphorus exhibit poor root growth, delayed maturity, and smaller heads. Phosphorus deficiency can also lead to decreased nutrient uptake, negatively impacting overall plant health [2]. Potassium is vital for maintaining plant turgor, enzyme activation, and disease resistance. Lettuce plants with potassium deficiency display wilted leaves, necrosis at the leaf margins, and reduced resistance to pathogens [3]. Potassium deficiency can reduce lettuce’s marketability due to decreased visual appeal [4].

This paper is structured as follows: Section 2 discusses previous research on lettuce deficiencies, Section 3 presents the materials and methods, Section 4 discusses the experimental results of the proposed method, and finally, Section 5 provides the conclusions of this article and discusses future work.
2. Related Work

In recent years, there has been growing interest in the development of deep learning-based approaches for the diagnosis and early detection of nutrient deficiencies in lettuce plants. Watchareeruetai et al. introduced, in 2018, an image analysis method for identifying nutrient deficiency in plants based on their leaves using convolutional neural networks [5], setting the stage for subsequent research in this area. In addition, a deep convolutional neural network for the image-based diagnosis of nutrient deficiencies in plants grown through aquaponics is proposed by Taha et al. in 2022 [6]. Furthermore, Lu et al., in 2023, introduced a lettuce pant trace-element-deficiency symptom identification method via machine vision methods [7]. Collectively, these studies represent significant contributions to the field of lettuce NPK deficiency detection and illustrate the increasing reliance on deep learning methodologies for precision agriculture applications. Continued research in this area is crucial to developing sustainable agricultural practices that can meet the increasing demand for high-quality lettuce. In this way, a deep learning approach called YOLO-NPK based on YOLOv8 Nano Classification algorithms [8,9] is employed in this study, to classify those deficiencies. The objective of this research is to enhance the baseline algorithm to achieve a top-1 accuracy above 85%, FLOP inferior to 10G, and classification latency below 170 ms per image.

3. Materials and Methods

3.1. Data Acquisition and Augmentation Strategy

The lettuce NPK dataset [10], sourced from Kaggle, comprises images representing various lettuce deficiency categories alongside Fully Nutritional (FN) lettuce samples. The dataset includes images categorized as FN with 12 images, Nitrogen-deficient (-N) with 58 images, Phosphorus-deficient (-P) with 66 images, and Potassium-deficient (-K) with 72 images. Captured in a controlled environment for hydroponic lettuce deficiency project, the dataset aims to facilitate the development of a system capable of recognizing lettuce deficiencies from images. This system would not only identify deficiencies in hydroponics but also find applications in diverse fields. Figure 1 provide a visual representation of the dataset samples, showcasing a fully nutritional, as well as nitrogen, phosphorus, and potassium deficiency [10].

![Figure 1](image-url)

**Figure 1.** The dataset samples. (a) Fully nutritional lettuce (FN); (b) nitrogen deficiency (-N); (c) phosphorus deficiency (-P); (d) potassium deficiency (-K).

Augmentation techniques were used to increase the training set and the validation set. The following pre-processing was applied to each image: auto-orientation of pixel data (with EXIF-orientation stripping) and resizing to $640 \times 640$ (Stretch). Furthermore, successive augmentation was applied to create augmented versions of each source image (50% probability of horizontal flip, 50% probability of vertical flip, and equal probability of one of the following 90-degree rotations: none, clockwise, counter-clockwise, upside-down, randomly cropped between 0 and 20 percent of the image, and random shear of between $-15^\circ$ and $+15^\circ$ horizontally and $-15^\circ$ and $+15^\circ$ vertically). In total, 3192 samples were obtained from augmentation, with -K 1175, -N 975, -P 847, and FN 195. Therefore, the dataset was split into 70% for the training and 30% for the validation.
3.2. VGG16 (Visual Geometry Group 16) Feature Extractor

VGG16 (Visual Geometry Group 16) is a convolutional neural network (CNN) architecture for deep learning that was developed by the Visual Geometry Group at the University of Oxford [11]. It is part of the VGG family of models and is known for its simplicity and effectiveness in image classification tasks. It consists of 16 weight layers, including 13 convolutional layers and 3 fully connected layers. The architecture uses 3 × 3 convolutional filters with a stride of 1, and 2 × 2 max-pooling layers with a stride of 2. Also, it is characterized by its deep architecture, with small 3 × 3 convolutional filters stacked multiple times. This depth helps the network learn complex hierarchical features from images. The network uses 3 × 3 convolutional filters with a stride of 1 and “same” padding, which means the spatial dimensions of the feature maps do not change after convolutions. Rectified Linear Units (ReLUs) are used as the activation function in the network, helping with the vanishing gradient problem and improving training.

3.3. Depthwise Convolution

Depthwise convolution is a specific type of convolutional operation used in deep learning and convolutional neural networks (CNNs). It is a fundamental building block for various lightweight and efficient neural network architectures, particularly those designed for mobile and edge devices [12]. Depthwise convolution differs from standard convolution in how it processes input channels. In a standard convolution operation, a kernel (also called a filter) slides through the entire input volume, considering all input channels simultaneously. In contrast, in depthwise convolution, each input channel is convolved with a separate kernel. This means that if you have k input channels and k separate kernels, each kernel is responsible for convolving with its corresponding input channel. It significantly reduces the number of parameters in the model compared to standard convolution. This reduction in parameters can lead to models that are more memory-efficient and that compute faster, making them suitable for resource-constrained environments. They are often used in conjunction with pointwise convolution (1 × 1 convolution). This combination is referred to as a depthwise separable convolution. In depthwise separable convolution, the depthwise convolution layer is followed by a 1 × 1 pointwise convolution layer. The pointwise convolution combines the information from the separate channels produced by the depthwise convolution. Lastly, it maintains the spatial dimensions (width and height) of the input, but it can change the number of channels (depth). This contrasts with standard convolution, which can also change spatial dimensions. So, it is particularly efficient when dealing with low-level features in an image, where inter-channel correlations are not as significant. Separating the channels reduces computational complexity.

3.4. YOLOv8 (You Only Look Once Version 8)

The YOLO (You Only Look Once) series [13–18] refers to a family of real-time object detection models that have been widely used in computer vision and deep learning. YOLO was initially introduced by Redmon et al. [9] in 2016 and has since seen several iterations, each with improvements and enhancements [9]. The primary idea behind YOLO is to perform object detection in a single forward pass of a neural network, making it very efficient and suitable for real-time applications. YOLOv8, developed by Ultralytics, represents the most recent iteration of the YOLO series. As an advanced and state-of-the-art model, it expands upon the achievements of its predecessors by introducing novel features and enhancements, resulting in elevated levels of performance, adaptability, and resource efficiency. YOLOv8 boasts comprehensive support for a wide spectrum of vision-based artificial intelligence tasks, encompassing detection, segmentation, pose estimation, tracking, and classification. This versatility empowers users to harness the diverse capabilities of YOLOv8 across a multitude of applications and domains.
3.5. YOLO-NPK

To enhance classification accuracy, a VGG16 feature extractor is integrated into the backbone of YOLOv8n-cls (YOLOv8 Nano Classification). Furthermore, depthwise convolution is introduced within the feature extractor to facilitate feature reuse and empower the deep network to extract more complex and richer features. The diagram below provides an overview of the proposed approach to classifying lettuce deficiencies. The diagram below provides an overview of the proposed approach to classifying lettuce deficiencies. The proposed feature extractor receives a 640 × 640 RGB deficient lettuce image as an input and extracts richer features. The classification head fuses the learned feature and performs a classification task, returning the classification result as the output. The schematic representation in Figure 2 delineates the architecture of YOLO-NPK, providing a visual guide to the components and their interactions. In this illustration, Conv signifies the convolutional layer, DW represents depthwise convolution, MP denotes the max-pooling layer, and nc stands for the number of classes. It’s important to note that the proposed feature extractor replaces the original backbone of YOLOv8n-cls, while the classification head remains unchanged. This visual scheme aims to elucidate the structural intricacies of our approach, aiding in a comprehensive understanding of the YOLO-NPK framework.

![Figure 2. The architecture of YOLO-NPK. Conv, DW, MP, and nc, respectively, stand for convolution, depthwise convolution, max-pooling layer, and number of classes. The original backbone of YOLOv8n-cls was replaced with the proposed feature extractor, and the classification head remained unchanged.](image)

4. Results and Discussion

4.1. Experimental Setup

The experiments were carried out on a computer equipped with the following specifications: an 11th Generation Intel® Core™ i5-11400H processor with 64-bit architecture, equipped with a dual-core CPU, and running at 2.70 GHz. Additionally, the computer was equipped with an NVIDIA GeForce RTX 3050 GPU. The model received input images sized at 640 × 640 pixels. However, due to constraints on GPU memory, the batch size was set to 8 during training. The training process spanned 116 epochs and commenced with an initial learning rate of 0.01, which was later adjusted to a final learning rate of 0.1. Moreover, the following specific hyperparameters were set: a momentum of 0.937 and a weight decay of 0.0005. During the warmup epoch, warmup momentum, and warmup bias learning rate stages, the values were configured at 3.0, 0.8, and 0.1, respectively. The optimizer employed for training the models was Stochastic Gradient Descent (SGD). Data augmentation techniques, such as mosaic, paste-in, and scaling, were used proportionally while training the deep network to avoid unbalanced classes. The early stop mechanism was employed to overcome overfitting.

In the context of classification accuracy, top-1 accuracy refers to the proportion of correctly classified samples where the model’s top prediction matches the true label. It can be mathematically expressed as follows:

$$
Top1\ Accuracy = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \times 100
$$

(1)
In this expression, *Number of Correct Predictions* is the count of instances where the model’s top prediction matches the true class labels, and *Total Number of Predictions* is the total number of instances or samples in the dataset. The result is typically expressed as a percentage to represent the accuracy rate. *Top-1 accuracy* is a common metric used to evaluate the performance of classification models, where only the highest-confidence prediction is considered for each sample.

### 4.2. Ablation Study

Several components of the YOLOv8n-cls backbone were modified to obtain the desired results. The overall structure of the backbone was replaced by the VGG16 feature to improve the classification accuracy, and the depthwise convolutional layers were inserted along the feature extractor to allow efficient memory computation and better reuse of features. These operations showed interesting improvement. Table 1 provides details on these diverse modifications.

**Table 1.** Ablation study on different modifications of YOLO-NPK.

<table>
<thead>
<tr>
<th>VGG16 Depthwise Convolution</th>
<th>Top-1 Accuracy (%)</th>
<th>FLOPs (G)</th>
<th>CPU Latency (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="%E2%9C%93" alt="Tick" /></td>
<td>93</td>
<td>3.3</td>
<td>19.8</td>
</tr>
<tr>
<td><img src="%E2%9C%93" alt="Tick" /></td>
<td>97.5</td>
<td>14.5</td>
<td>68.3</td>
</tr>
<tr>
<td><img src="%E2%9C%93" alt="Tick" /></td>
<td>95.2</td>
<td>2.4</td>
<td>18.2</td>
</tr>
<tr>
<td><img src="%E2%9C%93" alt="Tick" /></td>
<td>99</td>
<td>9.2</td>
<td>64.1</td>
</tr>
</tbody>
</table>

### 4.3. Classification Performance

The performance of YOLO-NPK was measured on the validation set, which represents all the classes. Notably, it shows acceptable results in terms of classification. The model performs efficiently on the FN set and achieves good classification results on other classes (-N, -P, and -K). To gain a comprehensive understanding of the model’s overall performance and the intricacies of class-wise classification, refer to Figure 3, which presents a confusion matrix. For a detailed illustration of the model’s classification output for each class, consult Figure 4 below.

![Figure 3](image_url) **Figure 3.** The confusion matrix of YOLO-NPK. (a) Confusion matrix. (b) Confusion matrix normalized. True represents the ground truth in the dataset, predicted is the classification result, and the background is the images that were missed by the model. This proves the learning capability of the proposed method. More details are provided in Table 2.
Figure 4. The classification output of YOLO-NPK. (a) Fully nutritional lettuce (FN); (b) phosphorus deficiency (-P); (c) nitrogen deficiency (-N); (d) potassium deficiency (-K).

Table 2. Classification performance of YOLO-NPK. FN, -N, -P, and -K, respectively, represent fully nutritional, nitrogen-deficient, phosphorus-deficient, and potassium-deficient lettuce.

<table>
<thead>
<tr>
<th>Classes</th>
<th>Images</th>
<th>Correctly Classified</th>
<th>Falsely Classified</th>
<th>Missed</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Count</td>
<td>Rate</td>
<td>Count</td>
<td>Rate</td>
</tr>
<tr>
<td>FN</td>
<td>53</td>
<td>100%</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>-N</td>
<td>279</td>
<td>98.21%</td>
<td>5</td>
<td>1.79%</td>
</tr>
<tr>
<td>-P</td>
<td>256</td>
<td>99.22%</td>
<td>2</td>
<td>0.78%</td>
</tr>
<tr>
<td>-K</td>
<td>370</td>
<td>99.19%</td>
<td>3</td>
<td>0.81%</td>
</tr>
</tbody>
</table>

4.4. Comparison of State-of-the-Art Methods

The proposed method, YOLO-NPK, was compared with different state-of-the-art methods. The proposed method shows better classification accuracy. The top-1 accuracy reached 99%, The FLOP 9.2G, and the classification latency per image 64.1 ms. This is in line with the guidelines established before the experiments (top-1 accuracy above 85%, FLOP under 10G, and latency below 170 ms). The other methods satisfied the FLOP and latency conditions but could not fulfil the top-1 accuracy expectation, proving the efficiency and robustness of the proposed model. Table 3 gives details of these comparisons.

Table 3. Comparison of the state-of-the-art method.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Images Size</th>
<th>Top-1 Accuracy (%)</th>
<th>FLOPs (G)</th>
<th>CPU Latency (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>640</td>
<td>85.3</td>
<td>12</td>
<td>141.6</td>
</tr>
<tr>
<td>VGG16</td>
<td>640</td>
<td>87.9</td>
<td>15.2</td>
<td>170.3</td>
</tr>
<tr>
<td>MobileNetV2</td>
<td>640</td>
<td>82.5</td>
<td>3.4</td>
<td>41.6</td>
</tr>
<tr>
<td>ShuffleNetv2</td>
<td>640</td>
<td>81.6</td>
<td>2.1</td>
<td>30.8</td>
</tr>
<tr>
<td>YOLOV8n-cls</td>
<td>640</td>
<td>93</td>
<td>3.3</td>
<td>19.8</td>
</tr>
<tr>
<td>YOLO-NPK</td>
<td>640</td>
<td>99</td>
<td>9.2</td>
<td>64.1</td>
</tr>
</tbody>
</table>

5. Conclusions and Future Work

This study introduces YOLO-NPK, a lightweight deep neural network tailored to lettuce deficiency classification, building upon the foundation of YOLOv8 Nano Classification. This research aimed to enhance the baseline algorithm by introducing a custom feature extractor aligned with the study’s needs. This goal was successfully met, achieving a top-1 accuracy exceeding 85%, maintaining a FLOP count under 10G, and ensuring a CPU latency below 170 ms per image, meeting the predefined objectives. Future plans involve integrating this solution into more complex systems for smart farming applications.

Author Contributions: Conceptualization, J.S. and J.C.N.; validation, J.S. and J.C.N.; investigation, J.S. and J.C.N.; data curation, J.S. and J.C.N.; formal analysis, J.S. and J.C.N.; methodology, J.S. and J.C.N.; software, J.S. and J.C.N.; visualization, J.S. and J.C.N.; supervision, J.C.N.; writing—original draft preparation, J.S.; writing—review and editing, J.S. and J.C.N.; project administration, J.C.N. 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 Lettuce NPK dataset used in this project was provided by Kaggle. The dataset is available in [10].

Conflicts of Interest: The authors declare no conflicts of interest. CAS Corporation and Guineal Maintenance have no conflict of interest.

References


2. Khan, F.; Siddique, A.B.; Shabala, S.; Zhou, M.; Zhao, C. Phosphorus Plays Key Roles in Regulating Plants’ Physiological Responses to Abiotic Stresses. Plants 2023, 12, 2861. [CrossRef] [PubMed]


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.