

Article

UAV-Derived Multispectral Datasets and Index-Guided Segmentation for Maize Water Stress and Common Rust Detection Under Real Field Conditions

Çağrı Suiçmez ^{1,2,*}, Cemal Yılmaz ¹, Hamdi Tolga Kahraman ³ and Mehmet Akif Erdoğan ⁴

¹ Electrical Electronics Engineering, Faculty of Technology, Gazi University, 06500 Ankara, Turkey; cemal@gazi.edu.tr

² Graduate School of Natural and Applied Sciences, Gazi University, 06500 Ankara, Turkey

³ Industrial Engineering, Faculty of Engineering, Karadeniz Technical University, 61080 Trabzon, Turkey; htolgakahraman@ktu.edu.tr

⁴ Landscape Architecture, Faculty of Architecture, Hatay Mustafa Kemal University, 31060 Hatay, Turkey; maerdogan@mku.edu.tr

* Correspondence: cagrisuicmez@gazi.edu.tr

Abstract

The segmentation model achieved Mean IoU values of 0.7723 for Water Stress 2025, 0.9164 for Common Rust 2025, and 0.9531 on the benchmark dataset. The classifier achieved 99.54% accuracy for the five-class task; however, the improvement over the strongest baselines and the RGB + multispectral configuration was limited. Therefore, the classification component is not presented as a substantially superior classification-only model. Instead, it is interpreted as an exploratory multimodal analysis that quantifies the contribution and limitation of RGB, multispectral, Wavelet, and GLCM branches under the adopted UAV dataset protocol. For classification-only deployment, simpler alternatives such as DenseNet201 or the RGB + multispectral configuration may be more practical because they provide comparable accuracy with lower architectural or preprocessing complexity. Ablation, modality-controlled, and 21-run stability experiments showed reproducible segmentation results and clarified the behavior of the classification branches. RGB and multispectral branches mainly provided the peak classification accuracy, whereas Wavelet and GLCM branches mainly affected offline convergence rather than final accuracy. RGB, NDVI, and NDRE visualizations were also added for qualitative support. Since direct physiological ground measurements were not available for all samples, the masks are interpreted as adaptive index-guided labels rather than direct physiological ground truth. Overall, the main evidence of practical benefit is associated with UAV-based dataset construction, adaptive index-guided segmentation, and field-scale stress/disease mapping, while the classification experiments should be interpreted as modality-contribution and convergence analyses rather than proof of a practically superior complex classifier.

Keywords: UAV multispectral and RGB imagery; maize water stress and rust disease detection; dual-branch UNet++ segmentation model; multimodal deep learning fusion; Precision agriculture applications

Academic Editors: Douglas O'Shaughnessy and Nektarios Valous

Received: 1 June 2026
Revised: 2 July 2026
Accepted: 3 July 2026
Published: 8 July 2026

Copyright: © 2026 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Maize is one of the most widely cultivated cereal crops and has a major role in food security [1]. However, its limited resistance to diseases and susceptibility to environmental stress factors have led to substantial yield reductions across various regions [2]. Notably, during the transition from the vegetative to the reproductive growth stage, particularly between July and August, the prevalence of foliar fungal diseases such as northern leaf blight (*Exserohilum turcicum*), gray leaf spot (*Cercospora zeae-maydis/zeina*), and southern leaf blight has been reported to cause yield and income reductions exceeding 50% in some regions [3]. Furthermore, common rust (*Puccinia sorghi*) and diplodia ear and stalk rot (*Stenocarpella maydis*) are among the most prevalent maize pathogens, with rust disease, in particular, posing a significant threat to yield in tropical and subtropical regions [4]. Particularly under drought-related water stress conditions, maize yields can decline by as much as 40%, underscoring the critical importance of efficient water management practices [5]. Challenges in detecting and managing water stress result not only in direct yield reductions but also in indirect damage by promoting disease development and exacerbating the effects of existing pathogens [6,7]. These challenges primarily arise from three major gaps in the current literature: (i) the lack of integrative models that jointly capture the physiological and molecular impacts of key stress factors such as water stress and rust disease, (ii) the scarcity of comprehensive datasets capable of effectively representing the problem domain for accurate detection and diagnosis, and (iii) the absence of problem-specific segmentation approaches and tailored deep learning architectures for these stress conditions. Collectively, these deficiencies form the core motivation and research objectives of the present study. The key shortcomings observed in recent research on the detection and diagnosis of maize diseases are summarized as follows.

- Reliable plant disease detection requires real-field datasets that include shadows, soil texture, illumination changes, and different disease levels. Controlled datasets may give high accuracy, but they often do not represent real agricultural conditions [8]. This also limits the performance of modern attention-based and segmentation models [9].
- Segmentation is important because it separates diseased or stressed regions from background and healthy tissue at the pixel level. This can improve classification and severity estimation. However, real-field studies using attention-based or weakly supervised segmentation are still limited [10].
- Many existing studies use datasets with limited variation in background, illumination, and disease severity. This reduces generalization and increases the risk of overfitting under field conditions [11,12]. Although advanced segmentation models have been used, many studies still depend on specific dataset conditions [13]. In addition, performance is often reported only with accuracy, while metrics such as IoU, F1-score, recall, and precision are also needed [12].
- Finally, while most studies in the literature focus primarily on model accuracy in disease detection, diagnostic time and stability have rarely been analyzed in a systematic manner. This limitation creates a critical gap in model selection and field deployment, particularly for real-time agricultural applications.

Deep learning has improved plant disease classification in recent years. Hybrid models also show that combining spectral, spatial, and textural information can improve robustness under complex field conditions. Therefore, this study uses RGB, multispectral, wavelet, and GLCM features together for five-class classification.

Based on the research gaps in the literature detailed above, a comprehensive research was carried out in this article for the detection and diagnosis of water stress and common rust disease in maize. In response to these gaps, the present study contributes to the liter-

ature in four aspects, with the main emphasis placed on UAV-based dataset construction, adaptive index-guided segmentation, and field-scale stress/disease mapping. The classification component is evaluated as a complementary modality-analysis and convergence-analysis module rather than as a substantially superior classification-only model. The main contributions are summarized as follows:

- **Dataset contribution:** Two original UAV-based multispectral datasets, Water Stress 2025 and Common Rust 2025, were generated under real field conditions to address the scarcity of realistic maize stress and disease datasets. These datasets form the primary empirical basis of the study and provide a real-field test environment for evaluating segmentation and classification behavior under UAV imaging conditions.
- **Segmentation contribution:** The proposed dual-branch UNet++ segmentation model, hereafter referred to as scar-U-Net++ DB-based segmentation model, was proposed for pixel-level detection of water stress levels and rust disease in maize under complex field conditions. This component represents the main model-oriented contribution of the study because it directly supports field-scale stress and disease mapping from UAV-derived multispectral data.
- **Classification analysis contribution:** A problem-specific multimodal classification analysis was conducted using RGB, multispectral, Wavelet, and GLCM features for five-class maize stress and disease recognition. This component is not positioned as a substantially superior classification-only model. Instead, it was used to quantify the contribution and limitation of each modality and to determine whether the additional Wavelet and GLCM branches provide practical benefit beyond RGB and multispectral inputs.
- **Verification and stability contribution:** The proposed workflow was evaluated through 21 independent runs to assess reproducibility and threshold-based convergence behavior. These experiments supported the stability of the UAV-based segmentation and monitoring workflow, while also showing that the full classification branch is not uniformly preferable to simpler classification-only alternatives.

This positioning is important because the classification experiments showed only a limited improvement over strong baselines and the RGB + multispectral configuration. Therefore, the full classifier is not presented as the default practical choice for classification-only deployment. Instead, the classification results are used to analyze modality contribution, convergence behavior, and the trade-off between accuracy and model complexity.

Overall, the main findings show that the proposed UAV-based workflow introduces two real-field multispectral maize datasets and provides effective index-guided segmentation performance for water stress and common rust detection. The main practical implication of the study is associated with UAV-based dataset construction, adaptive index-guided segmentation, and field-scale stress/disease mapping, whereas the classification experiments are interpreted as modality-contribution and convergence analyses rather than as evidence of a universally superior classification-only model.

The study is organized as follows. Section 2 presents the literature review. Section 3 describes the materials and methods. Section 4 presents the experimental results. Section 5 discusses the findings, including computational aspects, generalization, and practical implications. Finally, Section 6 concludes the study.

2. Related Studies

This section reviews and discusses current and qualified artificial intelligence based research on maize disease diagnosis in the literature. Important information about the studies in the literature and the work presented in this paper are presented in Tables 1

and 2. The studies in Table 1 are analyzed algorithm-wise and the studies in Table 2 are analyzed dataset-wise.

Among the reviewed studies [14–46], only a few address both segmentation and classification under real field conditions. Most studies focus on either image-level classification or pixel-level segmentation and are often tested on controlled datasets such as PlantVillage. In contrast, the present study evaluates a UAV-based workflow that combines real-field multispectral dataset construction, adaptive index-guided segmentation, and complementary classification analysis. Table 1 provides a compact literature positioning, while the full comparison is provided in Supplementary Table S1. The values in Table 1 should not be interpreted as a direct performance ranking because the reviewed studies differ in crop type, sensor configuration, class definition, dataset scale, and evaluation protocol.

Table 1. Literature review of artificial intelligence-based studies on agricultural products in terms of algorithms and performance metrics.

Ref.	Method	Task	Accuracy	Mean IoU
[43]	EfficientNet-B3 and EfficientNet-B5	Classification	0.9999	-
[19]	Fusion U-Net	Segmentation	0.9750	0.8800
[26]	LW-MUNet	Hybrid	0.9743	0.9689
This study	UAV-based index-guided segmentation and multimodal classification analysis	Hybrid	0.9954	0.9164 (common rust)/0.7723 (water stress)

Note: The values reported in Table 1 are not directly comparable because the studies use different datasets, sensors, crops, class definitions, and evaluation protocols. Therefore, the table is intended for literature positioning rather than for ranking methods by accuracy or Mean IoU. In the internal classification benchmark of the present study, the proposed classifier achieved competitive accuracy, but its gain over DenseNet201 and the RGB + multispectral configuration was limited. Thus, the classification component is interpreted as a modality-contribution and convergence-analysis module rather than as evidence of a practically superior classification-only model.

The disease-specific methods reviewed in this study were not directly reimplemented because they used different crops, diseases, sensors, input types, resolutions, class definitions, and evaluation protocols. In many cases, code, trained weights, preprocessing steps, and dataset splits were also unavailable. Therefore, this study used a controlled internal benchmark with widely used deep learning models. All baselines were trained and tested on the same UAV datasets, class definitions, and evaluation protocol. This provides a fair internal comparison, but it is not a direct reproduction of all previous methods. Within this benchmark, the proposed classification branch achieved competitive performance. However, its improvement over the strongest classification baselines was small. Therefore, the classification results are mainly used to analyze modality contribution, convergence behavior, and the accuracy–complexity trade-off, not to claim a large accuracy advantage over simpler models.

The limitations identified in the studies presented in Table 1, as well as the specific research gaps addressed by the present work, are discussed in detail in the following paragraphs corresponding to each column of Table 1.

- **Segmentation deficiency:** In Supplementary Table S1, only a few studies include both classification and segmentation. Most studies still focus on image-level disease recognition. However, segmentation gives pixel-level information and can separate soil, healthy plants, water-stressed plants, and rust-infected regions. This supports more targeted irrigation and spraying decisions. Therefore, integrated real-field frameworks for both segmentation and classification are still needed.

Contribution to the literature on segmentation: In this study, a segmentation-based approach enables the precise identification of water stress and rust disease at the plant level. The differentiation of stress severity supports optimized irrigation and spraying strategies, while soil segmentation improves resource management in agricultural fields. The proposed dual-branch U-Net++ architecture provides robust class separation across multiple resolutions, and the Skip Connection Attention Block preserves spatial information while improving boundary delineation [47].

- Discussion on the limitations of existing classification approaches: The analysis of Table 1 and Supplementary Table S1 shows that many classification studies report high final accuracy. However, these values are difficult to compare because the studies use different datasets, input modalities, and evaluation protocols. Many studies also focus on peak accuracy without discussing the trade-off between accuracy, convergence, parameter count, preprocessing complexity, and deployment feasibility. This trade-off is important for real-field UAV applications because a small accuracy gain may not justify a much more complex classifier. Therefore, the classification component of this study is evaluated not only by final accuracy, but also by branch-level ablation, modality-controlled stability, convergence behavior, and model complexity.

Contribution of the classification analysis to the literature: This study uses a problem-specific classifier to analyze the contribution of RGB, multispectral, Wavelet, and GLCM representations under the same UAV-based five-class protocol. The Ghost module, SE block, CBAM, and residual connections were included to support efficient feature generation, channel-wise recalibration, spatial-channel attention, and residual feature propagation, respectively [48–51]. The multi-branch structure was used to combine RGB, multispectral, textural, and frequency-domain representations within a unified classification analysis [52]. However, the experimental results showed that the RGB and multispectral branches were the main contributors to peak classification accuracy, whereas the Wavelet and GLCM branches mainly affected offline convergence rather than final accuracy. Therefore, this component is interpreted as a modality-contribution and convergence-analysis module, not as a lightweight or universally superior classification-only model.

- Limitations of the benchmarked models: Another limitation observed in Table 1 concerns benchmarking strategies. Many studies compare their proposed models with only a limited set of baseline architectures. As a result, the reported improvements may not fully reflect the true competitiveness of the models under broader experimental conditions.

Contributions to the literature regarding the limitations of benchmarked models: In this study, the benchmark was expanded by testing baseline and optimized versions of several ImageNet-based models. These included VGG, ResNet, DenseNet, Inception, EfficientNet, and Xception. This comparison was used to test whether the proposed multimodal classifier gives a meaningful practical advantage over strong classification-only baselines under the same UAV-based five-class protocol. The results showed competitive accuracy, but the gain over the strongest baselines was small. Therefore, the benchmark results do not support presenting the full classifier as a clearly superior classification-only model. Instead, they show a more limited contribution. The classification experiments mainly analyze modality contribution, convergence behavior, and the trade-off between accuracy and architectural complexity.

- Insufficiency of multi-class classification approaches: Although many studies perform multi-class classification, most focus only on disease detection and do not simultaneously consider soil, graded water stress levels, and rust disease within a uni-

fied framework. However, separating soil areas from vegetation and identifying different stress levels are essential for precision agriculture applications.

Moreover, it has been reported that classification performance may decrease as the number of classes increases due to higher model complexity and the sensitivity of the loss function, as illustrated in Equations (1) and (2). Therefore, the joint solution of multi-class classification and segmentation remains a challenging problem in the literature.

$$L_{binary} = -\frac{1}{N} \sum_{i=1}^N [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)] \quad (1)$$

$$L_{categorical} = -\frac{1}{N} \sum_{i=1}^N \sum_{k=1}^C y_{i,k} \log(\hat{y}_{i,k}) \quad (2)$$

where C represents the number of classes, N denotes the number of samples, y_i indicates the ground truth values, $\hat{y}_i$ refers to the predicted values. As observed in Equations (1) and (2), it is evident that an increase in the number of classes adversely affects model performance.

Contributions to the Literature on Multi-Class Classification: This study jointly classifies soil, healthy maize, low water stress, high water stress, and common rust. Separating soil can reduce unnecessary spraying, while stress and rust detection can support better irrigation and chemical application. The segmentation outputs were used in a customized classification model, which keeps the workflow consistent with the UAV-based problem.

- **Lack of Stability Analysis:** In the existing literature, no stability analysis has been reported for segmentation or classification studies. However, performance metrics derived from single-run experiments are insufficient to reflect the true capability of deep learning models, which inherently involve stochastic behavior and randomness in their optimization processes.

Contribution to the Literature on Stability Analysis: Model stability was evaluated using multiple independent runs. Mean IoU was used for segmentation, and Accuracy was used for classification. SR, ACD, and MEC were also reported to measure success rate, computation time, and convergence speed. This allows the models to be compared not only by peak performance but also by reproducibility and learning efficiency.

Maize stress and disease studies use both controlled and field datasets [14,22–46,53–60]. However, many common datasets, such as PlantVillage, were collected under controlled conditions and do not represent the complexity of UAV field imagery. UAV-based datasets are more realistic but still limited. Therefore, Table 2 gives a short dataset comparison, and the full version is provided in Supplementary Table S2.

Table 2. Literature review of artificial intelligence-based studies on agricultural products in terms of dataset and disease type.

Ref.	Dataset	Disease Type
[43]	Plant Village	-
[58]	Original Dataset	Water stress and healthy
[59]	Original Dataset	Water stress and healthy
[60]	Original Dataset	Water stress and healthy class
[42]	Plant Village	Well-watered, Reduced-watered, Drought-stressed
[39]	Plant Village	Common rust, Healthy, Gray leaf spot, Northern leaf blight
[44]	Plant Village	Corn plant widespread rust, blight and gray leaf spot
[45]	Original Dataset	Well-watered condition, Moderate water stress, Severe water stress
[46]	Original Dataset	Well-watered, Reduced-watered, Drought-stressed

This study	Original Dataset	Gradual water stress, common rust, healthy and soil
------------	------------------	---

The deficiencies identified in the studies presented in Table 2 and the research gaps addressed by this study are examined below in relation to the dataset column of Table 2.

- Lack of UAV-based dataset preparation: (i) Most of the reviewed studies rely on close-range imaging methods based on handheld RGB cameras. However, these approaches are inadequate for large-scale agricultural analysis due to their limited coverage and lack of scalability in assessing the spatial distribution of stress and disease. (ii) The high spatial and spectral resolution provided by UAVs facilitates the simultaneous detection of water stress and rust disease, as well as the prediction of disease spread. (iii) Fixed flight altitude and pre-planned flight routes ensure uniformity and control in both image resolution and illumination-shadow conditions.
- Contribution of the UAV-based dataset: UAV imaging allowed large field areas to be covered in one flight and enabled spatial analysis of water stress and rust disease. The high spatial and spectral resolution supported precise detection [61]. Fixed flight altitude and planned routes also helped keep resolution and illumination conditions more consistent.
- This study analyzes common rust and graded water stress within a UAV-based workflow. The workflow includes dataset construction, adaptive index-guided segmentation, and complementary classification analysis. The five classes are soil, healthy maize, low water stress, high water stress, and rust. This class structure can support field-scale interpretation of vegetation, stress severity, and disease distribution. Therefore, the main practical value of the study comes from UAV dataset construction, adaptive index-guided segmentation, and field-scale stress/disease mapping. The classification experiments complement this workflow by analyzing modality contribution and convergence behavior. However, they do not show that the full classifier is always preferable to simpler classification-only models.

3. Methodology

This section first summarizes the proposed workflow shown in Figure 1. It then explains the segmentation and classification stages in detail.

The proposed method has three phases: dataset preparation and mask generation, the proposed dual-branch UNet++ segmentation model, hereafter referred to as scar-UNet++ DB segmentation, and five-class classification using RGB, multispectral, wavelet, and GLCM features.

As shown in Figures 1 and 2, the evaluated workflow has three main stages: dataset preparation and mask generation, segmentation with the scar-UNet++ DB model, and five-class classification using RGB, multispectral, Wavelet, and GLCM features. The classification stage is used to analyze modality contribution and convergence behavior under the same UAV-based protocol. It is not presented as a substantially superior classification-only model. The evaluated pipeline combines segmentation, classification, multimodal fusion, attention-based blocks, and stability analysis. These components were examined in terms of segmentation performance, modality contribution, convergence behavior, reproducibility, and the trade-off between classification accuracy and model complexity.

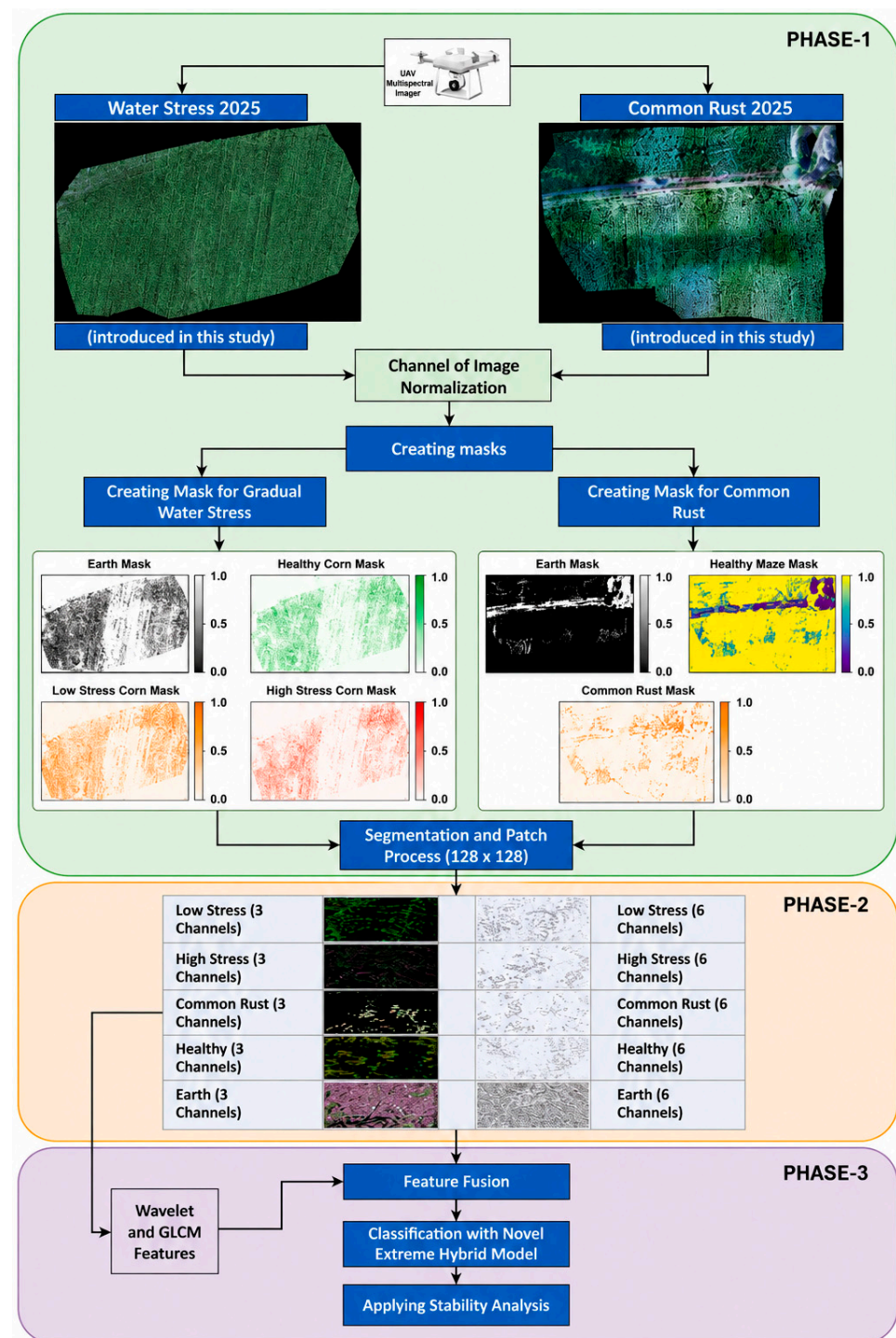


Figure 1. Graphical abstract of the UAV-based workflow evaluated in this study. The workflow includes dataset preparation, adaptive index-guided mask generation, segmentation-based field mapping, and complementary classification analysis. The blue blocks indicate the main evaluated components; they do not imply that the full classification branch is practically superior to simpler classification-only alternatives.

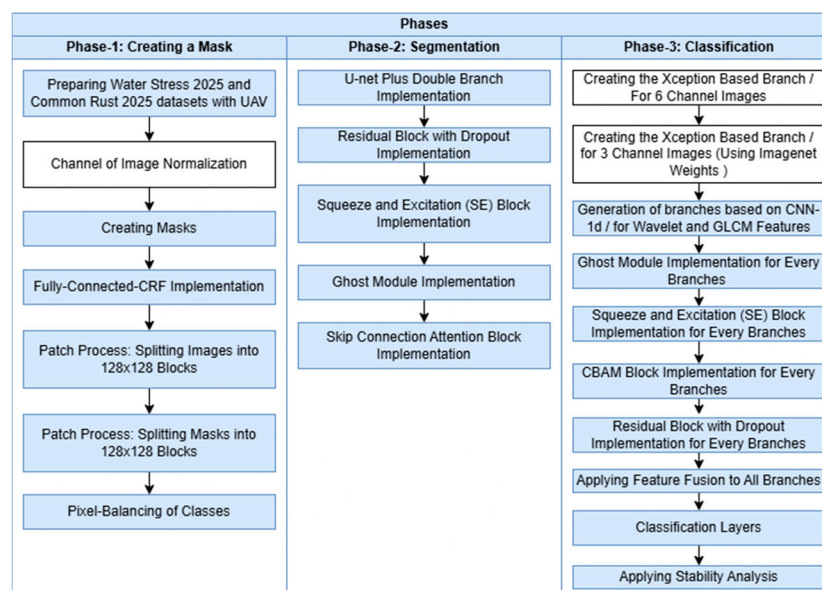


Figure 2. Methodological phases of the proposed study. Blue blocks indicate the main workflow components. Ghost and SE blocks in the segmentation phase represent candidate modules evaluated during the ablation stage rather than components retained in the final selected segmentation configuration. The classification phase summarizes the evaluated multimodal configurations and should be interpreted as a modality-contribution analysis rather than as a default deployment-oriented classifier.

Figure 3 shows the mask-generation pipeline for gradual water stress and common rust. The pipeline combines vegetation-index-based segmentation with automatic thresholding for complex remote sensing imagery.

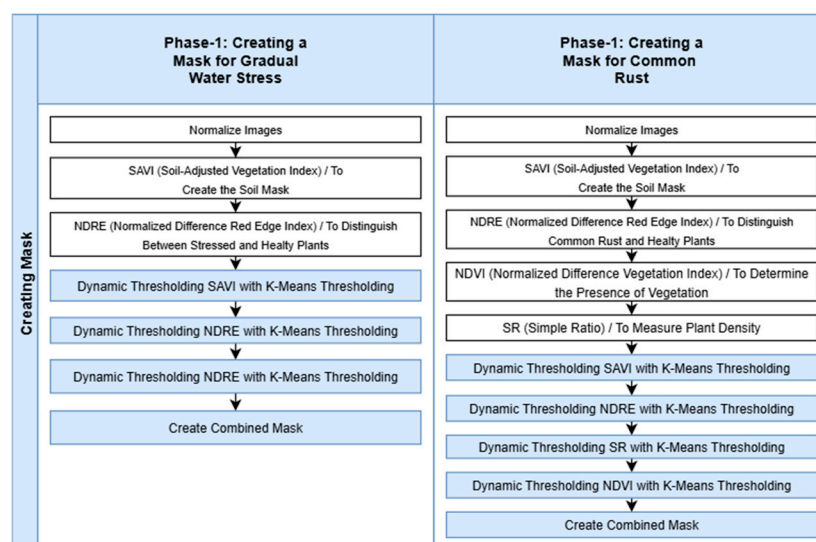


Figure 3. The stages of the mask extraction method we proposed for segmentation on two different datasets (the layers in blue are the layers we recommend).

Mask generation for gradual water stress (left panel)

For water-stress mask generation, raw multispectral images were first normalized. SAVI was used for soil–plant separation, and NDRE was used for stress detection. K-Means-based dynamic thresholding produced image-specific masks. Finally, SAVI- and NDRE-based masks were combined into one mask containing soil, stressed, and healthy plant regions.

Mask generation for common rust (right panel)

For rust detection, multispectral images were normalized, and SAVI was used for soil separation. NDRE, NDVI, and SR were then used to identify rust-affected areas, vegetation, and vegetation density. K-Means-based dynamic thresholding defined image-specific boundaries, and all intermediate masks were combined into one composite mask.

This mask-generation pipeline shows that multispectral indices combined with adaptive K-Means thresholding can produce more flexible masks than fixed thresholds. Because thresholds are estimated from each image, class boundaries can be defined more objectively.

Thus, the adaptive index-based process improved mask quality for both water stress and common rust. Since mask quality affects all later deep learning stages, this step is important for agricultural image analysis.

3.1. Creating a Mask

In this section, effective mask creation methods for mask segmentation success in Phase 1 and dataset creation for segmentation will be explained.

3.1.1. Preparation of Dataset

Maize water stress and common rust data were collected in Kırşehir, Turkey, using a DJI Phantom 4 Multispectral UAV at 15 m flight height. The Water Stress 2025 dataset includes 592 RGB and six-channel multispectral images, while the Common Rust 2025 dataset includes 568 images with the same configuration. The images have a spatial resolution of 1600×1300 pixels and 96 dpi. Representative RGB, multispectral patch, and orthomosaic examples are provided in Supplementary Figure S1, and additional comparisons with benchmark datasets are shown in Supplementary Figure S2.

The UAV data include RGB and six-channel multispectral images. Segmentation mainly used multispectral orthomosaic patches because vegetation indices require spectral bands. Classification was treated as a multimodal fusion task using RGB patches, multispectral patches, GLCM features, and wavelet features. Patch-level label assignment criterion: For the classification dataset, each 128×128 patch was assigned a single mutually exclusive class label using the majority-class rule based on the corresponding generated mask. In this procedure, the number of pixels belonging to each class was counted within the patch, and the class with the highest pixel proportion was used as the patch-level label. This rule was used to convert pixel-level adaptive index-guided masks into image-level labels required by the single-label classification model. Therefore, if a patch contained mixed regions such as soil, healthy maize, water-stressed vegetation, and rust-infected tissue, the final patch label was determined by the dominant class in that patch. RGB provides visible color and spatial information, multispectral data provides stress-sensitive spectral information, and GLCM/wavelet features provide texture and frequency information.

UAV images were collected under real field conditions using an RTK-enabled platform. Flights were conducted in the afternoon to keep lighting relatively stable. The orthomosaics had a ground sampling distance (GSD) of about 0.95–1.0 cm/pixel and included blue, green, red, red-edge, NIR, and one additional exported band. Sun position and crop rows were considered during flight planning to reduce shadows.

The data were collected from two maize production plots in Kaman, Kırşehir, Turkey. The first site was in İsaahocalı ($39^{\circ}25'02''$ N, $33^{\circ}54'30''$ E; 1275 m asl), and the second was in Aydınlar ($39^{\circ}20'41''$ N, $33^{\circ}56'57''$ E; 1200 m asl). Their locations are shown in Figure 4.

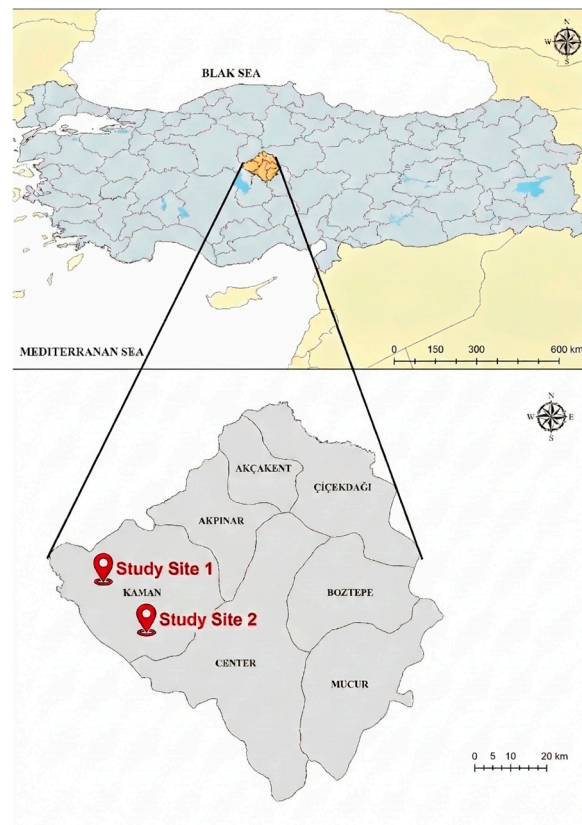


Figure 4. Geographical location of the study area.

It should be noted that the masks generated in this study were not manually annotated biological ground-truth masks. Instead, they were produced using adaptive vegetation-index rules, K-Means-based dynamic thresholding, and CRF refinement. Therefore, the segmentation task should be interpreted as learning and reproducing adaptive index-guided labels derived from multispectral information, rather than directly validating disease or water-stress symptoms against expert-annotated biological ground truth. This design was used as a first approximation to construct large-scale UAV-based segmentation labels under real field conditions, where dense manual pixel-level annotation is highly time-consuming.

3.1.2. Normalization of 6 Channel Multispectral Images

The two orthomosaic images were normalized to improve segmentation and classification. Robust scaling in Python 3.10.12 was used because it is less sensitive to outliers. The method is given in Equation (3).

$$X_{scaled} = \frac{X - Median(X)}{IQR(X)} \quad (3)$$

where X_{scaled} : new scaled, i.e., normalized image, X : original image, $Median(X)$: median of the image and $IQR(X)$: 75th percentile (Q3)–25th percentile (Q1) is the difference between. RobustScaler makes each band more stable against off-center values and better adapts to the distribution of data.

3.1.3. Creation of Masks for Segmentation of 6-Channel Multispectral Images

Gradual water-stress masks included four classes: low water stress, high water stress, healthy plants, and soil. Common rust masks included three classes. For water stress, SAVI was used for soil–vegetation separation, and NDRE was used for stress separation.

Dynamic thresholding, Otsu thresholding, K-Means clustering with $K = 4$, morphological filtering, and CRF refinement were then used to produce adaptive masks.

In addition to the above methods, the Simple Ratio Index (SR), whose equation is given in Equation (6), and the Normalized Difference Vegetation Index (NDVI), presented in Equation (7), were incorporated to generate the mask for common rust disease. Subsequently, combinations of these indices were subjected to Otsu thresholding [62], K-Means clustering [63] and Conditional Random Field (CRF) refinement [64–66].

Otsu thresholding and K-Means clustering were used for adaptive mask generation. Otsu supported binary separation, while K-Means supported multi-level grouping. Fully connected CRF refined the masks and improved boundary coherence. Detailed formulations are given in Supplementary Sections S15–S17.

The Otsu method was used to determine adaptive threshold values by maximizing the inter-class variance of pixel intensity distributions [62]. This approach was particularly useful for separating foreground and background regions under varying field conditions. The detailed mathematical formulation is provided in the Supplementary Section S15.

K-Means clustering grouped pixels with similar spectral responses and supported multi-level stress separation. Its formulation is provided in Supplementary Section S16. For common rust mask generation, $K = 3$ represented soil/background, healthy vegetation, and rust-affected vegetation. Binary K-Means was also used for soil–plant separation when needed.

Spectral indices and thresholding methods were selected using field knowledge and data-driven analysis. SAVI was used for soil–plant separation because it reduces soil brightness effects. NDRE was used for stress detection because it is sensitive to red-edge reflectance and chlorophyll changes.

Fixed thresholds were not used because vegetation-index values can change with location, illumination, crop rows, and growth stage. Instead, thresholds were estimated adaptively from each orthomosaic region. Otsu thresholding was used for binary separation, and K-Means clustering was used for multi-class separation. Morphological filtering and fully connected CRF were then applied to improve spatial consistency and boundaries.

Otsu thresholding was used to determine adaptive global thresholds from pixel distributions. K-Means was used when multi-level class separation was needed. The number of clusters was selected according to the dataset classes, such as $K = 4$ for water stress. Preliminary trials supported this choice. This hybrid approach produced more flexible masks under different field conditions.

$$SAVI = \frac{(NIR - RED)}{(NIR + RED + L)} \times (1 + L) \quad (4)$$

$$NDRE = \frac{(NIR - RedEdge)}{(NIR + RedEdge)} \quad (5)$$

$$SR = \frac{NIR}{RED} \quad (6)$$

$$NDVI = \frac{NIR - RED}{NIR + RED} \quad (7)$$

In these equations, NIR is the near-infrared band, RED is the red band, RedEdge is the red-edge band, and L is the soil brightness correction factor. L reduces soil effects in low-density vegetation areas and is generally set to 0.5.

Vegetation-index-based masks are not direct physiological ground measurements and should not be considered error-free ground truth. In this study, spectral indices were used to create spatially consistent training labels from UAV multispectral orthomosaics. Because index values can change with location, illumination, growth stage, and season,

Otsu and K-Means thresholding were applied adaptively. The masks were also refined with morphological operations and fully connected CRF. Therefore, they should be interpreted as adaptive index-guided labels.

3.1.4. Fully Connected-CRF Implementation

CRF is a probabilistic graphical model used to refine pixel-based segmentation outputs. Fully connected CRF considers relationships between all pixels and helps produce smoother class boundaries [64–66]. In this study, it was used as a post-processing step to improve mask consistency. The detailed formulation is provided in Supplementary Sections S17.

3.1.5. Splitting Images and Masks into 128×128 Patches

Processing full orthomosaic images requires high memory and computation, especially for multispectral data. Therefore, all images and masks were divided into 128×128 patches. This reduced computational cost, accelerated training, and increased data diversity.

Each orthomosaic and its mask were divided into matching 128×128 non-overlapping patches. This allowed each image patch to be paired with its corresponding mask patch. The approach also made large images easier to process on limited hardware and helped the model learn local details.

3.1.6. Class Distribution Analysis and Patch-Level Balancing

Class imbalance was analyzed using the pixel-level class distribution obtained from the generated masks. In segmentation, dominant classes can bias the learning process and reduce the visibility of minority classes such as water-stressed and rust-infected regions. Therefore, pixel-level class counts were first calculated from the training masks using Equation (25). However, these pixel counts were used only to analyze the class distribution and guide patch-level sampling. They were not used to randomly delete, mask, replace, or alter individual pixels inside the original image or mask patches.

For a training mask patch M_p the total number of pixels belonging to class k over the training set was calculated as:

$$n_k = \sum_{p=1}^P \sum_{i=1}^H \sum_{j=1}^W 1(M_p(i,j) = k) \quad (8)$$

where n_k denotes the total number of pixels belonging to class k , P is the number of training patches, H and W are the height and width of each patch, respectively, and $1(\cdot)$ is the indicator function. In this study, $H = W = 128$.

After obtaining the global class distribution, the relative class proportion inside each patch was calculated using Equation (26):

$$r_{p,k} = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W 1(M_p(i,j) = k) \quad (9)$$

where $r_{p,k}$ represents the proportion of class k inside patch p . This value was used to determine whether a patch contained minority-class information, but it did not change the spatial structure of the patch.

Then, the patch-level sampling weight was defined using class proportions and inverse class-frequency information, as shown in Equation (27):

$$s_p = \sum_{k=1}^K \frac{r_{p,k}}{n_k + \epsilon} \quad (10)$$

where s_p is the sampling weight of patch p , K is the number of classes, and ϵ is a small constant used to avoid division by zero. As shown in Equations (25)–(27), pixel-level counts were used only to estimate class distribution and compute patch-level sam-

pling weights. This formulation gives higher sampling priority to patches that contain relatively rare classes. The balanced training set was constructed by selecting complete image-mask patch pairs according to these patch-level sampling weights.

It is important to clarify that the balancing procedure described in Equations (25)–(27) was applied at the patch-sampling and dataset-composition level, not at the isolated-pixel level. Individual pixels were not removed, resampled, masked, or modified inside the original images or masks. Therefore, the spatial continuity of each 128×128 image-mask pair was fully preserved, and the convolutional input grid required by the segmentation model was not disrupted. Each selected image patch and its corresponding mask patch kept their original 128×128 spatial structure during training. Thus, the balancing strategy improved the representation of minority classes without destroying local texture information or introducing dimensional inconsistencies in the convolutional model.

3.2. Segmentation with Novel (Scar-U-Net++ DB)

In this section, UNet++ was used as the base segmentation architecture. Four candidate architectural modules, namely Ghost Module, Squeeze-and-Excitation (SE) Block, Skip Connection Attention Block (SkipAttention), and Residual Block, were evaluated through ablation experiments to determine the most effective configuration. The ablation results showed that the SkipAttention + Residual configuration provided the best overall segmentation performance. Therefore, the final segmentation model was defined as the proposed dual-branch UNet++ segmentation model, hereafter referred to as scar-U-Net++ DB architecture enhanced primarily with SkipAttention and Residual refinement. For the purpose of selecting these blocks, Ghost Modul provides parameter efficiency and gives similar accuracy with less computation. It significantly reduces FLOPs and model size in deep models [48,67]. SE Block adds channel sized attention. Frequently shown to significantly improve performance in areas such as ImageNet and segmentation [68,69]. SkipAttention transfers long distance dependencies between encoder and decoder more efficiently in architectures such as UNet; especially effective in segmentation of fine details [70]. Residual Block facilitates learning, reduces the vanishing gradient problem and facilitates optimization in deep networks [71]. The selected blocks offer less parameterization and more practical implementation. In addition, these blocks were chosen because they are the ‘most widely used and transferable’ modules in the literature. Then, in order to further evaluate the contribution of each block to the model performance, ablation studies were carried out. In this context, all possible combinations were generated according to the active/passive status of the four blocks and the segmentation success (Mean IoU) of each configuration was evaluated on the test dataset. The ranking of each combination was determined according to the Friedman scores given by the mathematical equations given in Section 4.4. It has also been proven that the most successful results among these combinations are Skip Attention and Residual blocks. The visual representations of the main model architecture from multiple perspectives are presented in Supplementary Figure S3.

3.2.1. Skip Connection Attention Block

Skip connections improve gradient flow, while attention helps the model emphasize important information [47]. The feature-map transformation is described in Equations (28)–(33).

$$\theta(x) = W_{\theta}x \quad (11)$$

$$\phi(s) = W_{\phi}s \quad (12)$$

where x , Input tensor. s , Skip connection tensor. W_{θ} and W_{ϕ} , 1×1 convolution weights. The calculation of the attention coefficient is carried out with the following equations.

$$f = \text{LeakyReLU}(\theta(x) + \phi(s)) \quad (13)$$

$$\psi(f) = \sigma(W_\psi f) \quad (14)$$

where W_ψ , 1×1 convolution weights and σ , Sigmoid is the activation function. The stage of the application of attention is as follows.

$$s' = s \cdot \psi(f) \quad (15)$$

$$x' = x \quad (16)$$

3.2.2. Residual Block

The Residual Block improves trainability and reduces gradient loss [51]. In this study, it consists of Conv, BN, ReLU, Dropout, Conv, BN, and shortcut layers. Its formulation is given in Equations (34)–(38).

$$z_1 = \text{BN}(W_1 \times x + b_1), z_2 = \text{ReLU}(z_1) \quad (17)$$

$$z_3 = \text{Dropout}_p(z_2) \quad (18)$$

$$z_4 = \text{BN}(W_2 \times \tilde{z} + b_2), \tilde{z} = z_3 \quad (19)$$

$$s = W_s \times x + b_s \quad (20)$$

$$y = \text{ReLU}(z_4 + s) \quad (21)$$

3.2.3. UNet++ Working Principle of the Model

The working principle of the main block is explained as encoder stage, bottleneck layer, decoder stage and output layer, respectively. In the encoder stage, both branches process the input image in parallel and feature maps are extracted using residual blocks. In the bottleneck layer, the encoder outputs are combined and processed in the bottleneck stage. In the decoder stage, the decoder upsamples the information using skip connection attention block. The mathematical formulations of the applied method are explained in Equations (39)–(46).

$$E_i^{(1)} = \text{ResidualBlock}(P_{i-1}^{(1)}), P_i^1 = \text{MaxPool}(E_i^{(1)}) \quad (22)$$

$$E_i^{(2)} = \text{ResidualBlock}(P_{i-1}^{(2)}), P_i^2 = \text{MaxPool}(E_i^{(2)}) \quad (23)$$

$$B = \text{Concat}(\text{ResidualBlock}(P_L^1), \text{ResidualBlock}(P_L^2)) \quad (24)$$

$$U_i = U_p(D_i + 1) \quad (25)$$

$$A_i = \text{Attention}(U_i, E_i^{(1)}) \quad (26)$$

$$\tilde{X}_i = \text{Concat}(A_i, E_i^{(2)}) \quad (27)$$

$$D_i = \text{ResidualBlock}(\tilde{X}_i) \quad (28)$$

In the output layer, n-class segmentation output is produced by using softmax activation in the last layer.

$$\hat{Y} = \text{Softmax}(W_{out} \times D_0 + b_0) \quad (29)$$

The proposed segmentation model uses the proposed dual-branch UNet++ segmentation model, hereafter referred to as scar-U-Net++ DB architecture with Residual refinement and Skip-Attention. Ghost and SE modules were tested during ablation but were not retained in the final configuration. Detailed 3D architecture views are provided in Supplementary Figure S3.

Algorithm 1 presents the pseudocode of the proposed segmentation architecture, while detailed 3D architectural views are provided in the Supplementary Materials (Figure S3).

Algorithm 1 presents the pseudocode of the proposed segmentation architecture.

Algorithm 1. Pseudocode of proposed segmentation model.

Input

Agriculture image dataset (N images, 3 or 6 channels)

Output

Segmentation masks, True mean IoU, accuracy

Encoder–Decoder Dual Branch Segmentation Process (scar-U-net++ DB)

```

1  for each image  $i$  in dataset do
2      Preprocessing:
3          Resize image to fixed size (e.g.,  $128 \times 128$ ).
4          Normalize pixel values to  $[0, 1]$ .  $\leftarrow$  Equation (3)
5      Feature Encoding—Branch 1:
6           $e1\_1$  = Residual block with dropout (filters = 64)  $\leftarrow$  Equations (17)–(20)
7           $p1\_1$  = MaxPool( $e1\_1$ )
8           $e2\_1$  = Residual block (filters = 128)  $\leftarrow$  Equations (17)–(20)
9           $p2\_1$  = MaxPool( $e2\_1$ )
10          $e3\_1$  = Residual block with dropout (filters = 256)  $\leftarrow$  Equations (17)–(20)
11          $p3\_1$  = MaxPool( $e3\_1$ )
12          $e4\_1$  = Residual block (filters = 512)  $\leftarrow$  Equations (17)–(20)
13          $p4\_1$  = MaxPool( $e4\_1$ )
14     Feature Encoding—Branch 2:
15          $e1\_2$  = Residual block (filters = 64)  $\leftarrow$  Equations (17)–(20)
16          $p1\_2$  = MaxPool( $e1\_2$ )
17          $e2\_2$  = Residual block with dropout (filters = 128)  $\leftarrow$  Equations (17)–(20)
18          $p2\_2$  = MaxPool( $e2\_2$ )
19          $e3\_2$  = Residual block (filters = 256)  $\leftarrow$  Equations (17)–(20)
20          $p3\_2$  = MaxPool( $e3\_2$ )
21          $e4\_2$  = Residual block with dropout (filters = 512)  $\leftarrow$  Equations (17)–(20)
22          $p4\_2$  = MaxPool( $e4\_2$ )
23     Bottleneck:
24          $b1$  = Residual block with dropout ( $p4\_1$ , filters = 1024)  $\leftarrow$  Equations (17)–(20)
25          $b2$  = Residual block ( $p4\_2$ , filters = 1024)  $\leftarrow$  Equations (17)–(20)
26          $concat\_bottleneck$  = Concatenate( $[b1, b2]$ )
27     Decoder Path:
28         for each level  $l$  from deepest to shallowest do
29              $up\_l$  = UpSampling2D( $concat\_bottleneck$  or previous decoder output)  $\leftarrow$  Equation (22)
30              $skip\_l$  = Skip Connection Attention Block( $up\_l$ , encoder feature, inter_channel)  $\leftarrow$  Equations (11)–(16)
31              $concat\_l$  = Concatenate( $[up\_l, skip\_l, \text{encoder feature (branch 2)}]$ )  $\leftarrow$  Equations (23) and (24)
32              $d\_l$  = Residual block( $concat\_l$ , filters)  $\leftarrow$  Equations (25)–(28)

```

33	<i>end for</i>
34	Final Segmentation Output:
35	Apply 1×1 Conv2D (filters = 3,4, activation = 'softmax') on last decoder output $\rightarrow$ predicted mask $\leftarrow$ Equation (29)
36	Training and Evaluation
37	Train the model with categorical cross entropy and true mean IoU metrics

Algorithm 1 gives the pseudocode of the proposed segmentation model, and the blue shaded parts show the proposed combinations.

Algorithm 1 summarizes a dual branch UNet++-derived encoder–decoder architecture designed to generate multi class segmentation masks from multi-channel agricultural images. Input images are resampled to a fixed size and normalized Equation (3). In each branch, residual blocks with progressively increasing channel depths are employed, incorporating dropout for regularization where necessary; after each block, spatial resolution is reduced via max pooling Equations (34)–(37). At the deepest level, features from both branches are processed through separate residual bottlenecks and fused at the bottleneck stage. Within the decoder, features are progressively up sampled Equation (39), aligned and enhanced using Skip Attention units that emphasize encoder representations Equations (28)–(33), and then concatenated with corresponding encoder features from both branches Equations (40)–(41) before being refined through residual blocks Equations (42)–(45). In the final layer, a 1×1 convolution followed by a softmax activation produces pixel level class probabilities Equation (46). The model is trained using categorical cross entropy loss, and evaluated based on accuracy and bias corrected true mean IoU, among other relevant performance metrics.

3.3. Classification with Novel (XGhost-SECBAM-r-QB) and Wavelet, GLCM Features

Although some architectural components used in the proposed classifier, such as Xception, Ghost, Squeeze-and-Excite, CBAM, residual connections, and attention mechanisms, were originally introduced in general computer vision or have also been used in medical image analysis, they are not specific to medical imaging. These modules operate on generic image-feature representations, including channel-wise feature recalibration, spatial attention maps, residual feature propagation, and depthwise separable convolution. Therefore, they can be transferred to UAV-based agricultural image analysis when the input data are represented as structured image tensors or feature vectors. In this study, these components were adapted to RGB, multispectral, Wavelet, and GLCM branches to improve feature representation under real field conditions. Thus, the cited studies are used to support the methodological basis of the architectural modules rather than to claim direct contextual equivalence between medical imaging and agricultural UAV monitoring. This clarification is provided to avoid any misunderstanding regarding the contextual scope of the cited references.

The structured multi-branch, fusion based classification architecture, comprising parallel branches that process RGB, multispectral, wavelet, and GLCM features, is illustrated in Figure 5.

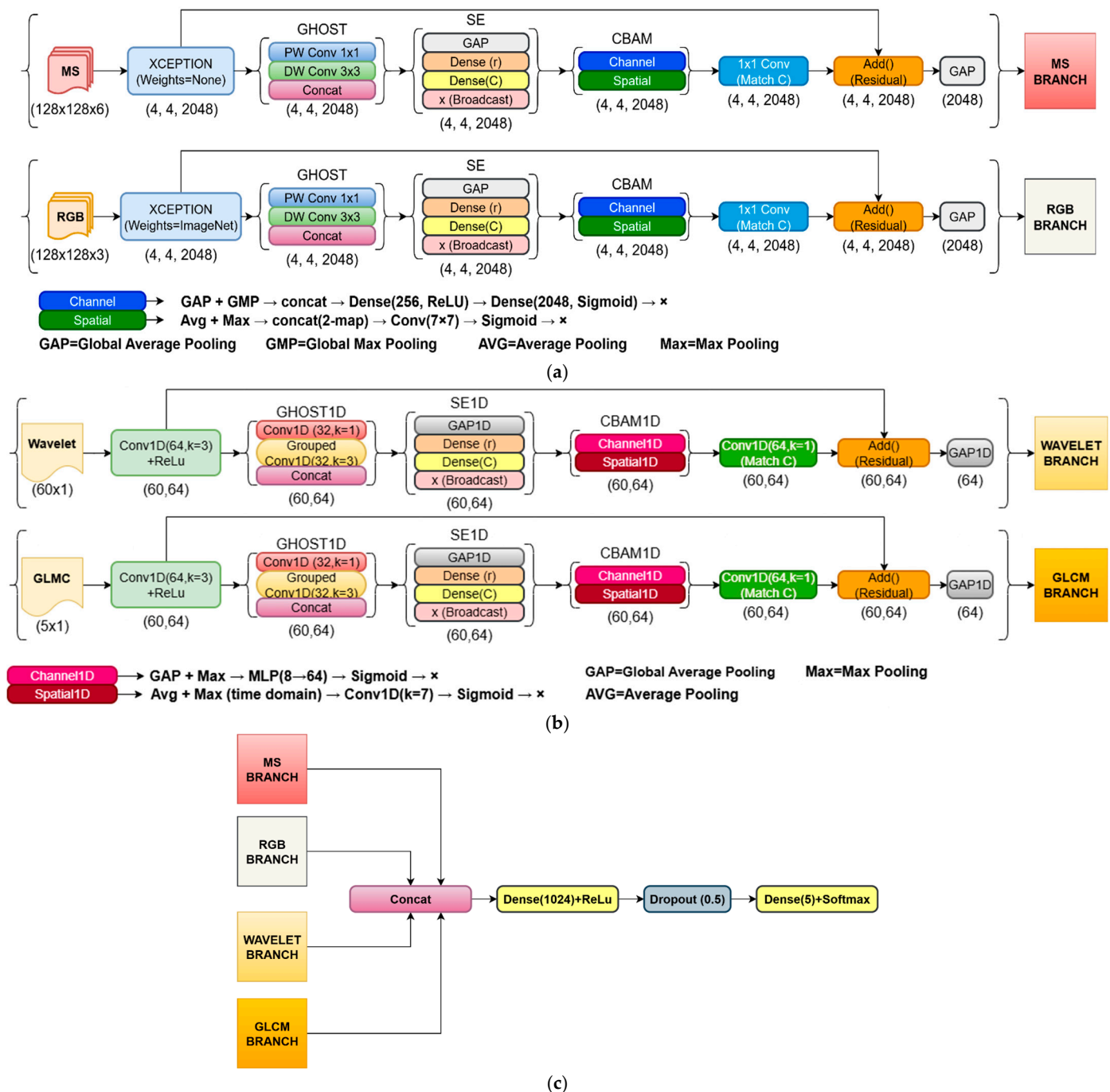


Figure 5. Classification model: (a) details of the MS and RGB branches; (b) details of the Wavelet and GLCM branches; (c) concatenation of branches. Different colors are used to distinguish the functional components of the classifier. Red/pink denotes the multispectral branch, light gray denotes the RGB branch, light yellow denotes the Wavelet branch, and orange/yellow denotes the GLCM branch. The purple/pink block indicates feature concatenation and fusion, the blue block indicates dropout-based regularization, and the yellow blocks indicate the final dense classification layers.

This study introduces a multi-branch classifier that integrates RGB, multispectral, wavelet, and GLCM features. RGB and multispectral images are processed with Xception-based branches, while wavelet and GLCM features are processed with CNN1D branches. Ghost, SE, CBAM, and Residual components improve feature learning in each branch. The extracted features are fused and passed through Dense–Dropout–Softmax layers to produce the final five-class prediction.

3.3.1. Xception Base Model

The Xception (Extreme Inception) model is an extended version of the Inception architecture and improves parameter efficiency by using Depthwise Separable Convolution. By factorizing classical convolution operations, it reduces the computational cost and increases the learning capacity of deep networks [72].

In this section, the mathematical formulations of the Xception model will be explained in detail. The Xception model generally consists of t stages. These are Standard Convolution Process and Depthwise Separable Convolution (the basic concept of Xception). Mathematical models of these stages are given below, in Equations (30)–(41), respectively.

$$Y(i, j, k) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \sum_{c=0}^{C-1} X(i+m, j+n, c) W(m, n, c, k) \quad (30)$$

$X(i, j, c)$, is the channel value c at pixel (i, j) of the input image. $W(m, n, c, k)$ convolution filter of size (m, n) with k numbers. $Y(i, j, k)$ output activation in channel k . This process has a large computational burden. The Xception model divides the standard convolution into two stages. These are depthwise convolution and pointwise convolution (1×1 Convolution), whose equations are given below. Depthwise convolution performs an independent convolution for each channel. Pointwise convolution connects each channel and is weighted.

$$Y_d(i, j, c) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} X(i+m, j+n, c) W_d(m, n, c) \quad (31)$$

where $W_d(m, n, c)$, separate filter per channel for Depthwise convolution. $Y_d(i, j, c)$, intermediate output calculated separately for each channel.

$$Y(i, j, k) = \sum_{c=0}^{C-1} Y_d(i, j, c) W_p(c, k) \quad (32)$$

where $W_p(c, k)$, 1×1 convolution filter. $Y_d(i, j, c)$, final output activation. Below are the general layers of the Xception model and their mathematical expressions. The first layer is the input layer, the second is the entry flow (first phase), the third is the middle flow (main deep network blocks) and finally the exit flow (last phase). The mathematical expressions are given below.

The input size of the model starts with $X \in \mathbb{R}^{H \times W \times C}$.

$$X_0 = \text{ReLU}(\text{BatchNorm}(W_0 \times X)) \quad (33)$$

W_0 , First convolution filters. BatchNorm, Batch normalization is applied. ReLU, Used as activation function. In the entry flow stage, it starts with standard convolution and continues with deep decomposed convolutions. This process is repeated 3 times and maximum pooling is applied. The aim of this stage is to reduce the size of the feature maps and compress the visual information.

$$X_1 = \text{ReLU}(\text{BatchNorm}(\text{DepthwiseConv}(X_0) + \text{PointwiseConv}(X_0))) \quad (34)$$

$$X_2 = \text{MaxPooling}(X_1) \quad (35)$$

The middle flow stage consists of linked residual blocks. Each residual block is modeled as follows:

$$X_{i+1} = X_i + \text{ReLU}(\text{BatchNorm}(\text{DepthwiseConv}(X_i) + \text{PointwiseConv}(X_i))) \quad (36)$$

These repeated blocks increase the depth of the model, enabling more powerful feature learning.

Exit flow is the exit stage of the model and includes the final deep decomposed convolution, global average pooling (GAP) and fully connected layers, respectively.

$$X_f = \text{ReLU}(\text{BatchNorm}(\text{DepthwiseConv}(X_n) + \text{PointwiseConv}(X_n))) \quad (37)$$

$$G_c = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W X_f(i, j, c) \quad (38)$$

$$Y = \text{Softmax}(WY_f + b) \quad (39)$$

The Xception backbone was used as the core feature extractor for the RGB and multispectral image branches of the proposed classification framework. In the main manuscript, the overall multi-branch classification architecture is summarized in Figure 5 while the detailed Xception-based branch architecture and the internal structure of Xception blocks are provided in the Supplementary Materials (Figure S4). This organization keeps the main text focused on the proposed multimodal fusion framework while preserving the full architectural details for reproducibility.

Xception and Xception ('imagenet') were used to process multispectral and RGB images, respectively.

Xception output,

$$f_{\text{multispektral}} = \text{ResidualAttentionBlock}(\text{Xception}(I_{\text{multispektral}})) \quad (40)$$

Xception ('imagenet') output,

$$f_{\text{RGB}} = \text{ResidualAttentionBlock}(\text{Xception}^{\text{'imagenet'}}(I_{\text{RGB}})) \quad (41)$$

3.3.2. CBAM Block

CBAM (Convolutional Block Attention Module) implements both channel and spatial attention mechanisms. In this block, channel attention and spatial attention are applied together. The channel equations are given below in Equations (42)–(45) [50].

Channel average and maximum pooling equations,

$$f_{\text{avg}} = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W X_{i,j} \quad (42)$$

$$f_{\text{max}} = \max_{i,j} (X_{i,j}) \quad (43)$$

Fully connected layers and activation equation,

$$M_c = \sigma(W_2 \delta(W_1 [f_{\text{avg}}, f_{\text{max}}])) \quad (44)$$

Scaling,

$$X' = X \cdot M_c \quad (45)$$

The spatial attention equations are given below in Equations (46)–(49). Average and maximum pooling,

$$f_{\text{avg_spatial}} = \frac{1}{C} \sum_{c=1}^C X_c \quad (46)$$

$$f_{\text{max_spatial}} = \max_c (X_c) \quad (47)$$

Convolution attention map,

$$M_s = \sigma(W_s \times [f_{avg_spatial}, f_{max_spatial}]) \quad (48)$$

Conclusions,

$$X'' = X' \cdot M_s \quad (49)$$

3.3.3. Residual Attention Block

Residual Attention Block includes Ghost, SE and CBAM blocks. Ghost module application, the mathematical formulations of the applied method are explained in Equations (50)–(53).

$$X_1 = \text{GhostModule}(X) \quad (50)$$

SE block application,

$$X_2 = \text{SEBlock}(X_1) \quad (51)$$

CBAM block application,

$$X_3 = \text{CBAM}(X_2) \quad (52)$$

Last layer and skip connection,

$$X_f = X + \text{Conv}(X_3) \quad (53)$$

3.3.4. CNN1D Based Double Branch (for Wavelet and GLCM Features)

Wavelet and GLCM features were processed using a 1D CNN-based classification branch. This branch includes Ghost1D, SE1D, CBAM1D, and Residual1D components to improve efficiency and feature representation. The mathematical descriptions are given below.

Ghost Module (1D)

The Ghost Module is based on the idea that similar representations can be generated more cheaply by using a smaller number of basic convolution filters. A given input tensor $X \in \mathbb{R}^{T \times C_{in}}$, number of target filters C_{out} and the ratio r the mathematical formulation of the module is presented in Equations (54)–(56).

True convolution;

$$Y_1 = \text{Conv1D}(X, W_1), \quad W_1 \in \mathbb{R}^{k \times C_{in} \times \frac{C_{out}}{r}} \quad (54)$$

Ghost convolution (grouped);

$$Y_2 = \text{GroupConv1D}(Y_1, W_2), \quad W_2 \in \mathbb{R}^{k \times \frac{C_{out}}{r} \times \frac{C_{out}}{r}} \quad (55)$$

Output merging;

$$Y = \text{Concat}(Y_1, Y_2) \in \mathbb{R}^{k \times C_{out}} \quad (56)$$

Squeeze and Excitation Block (1D)

This block highlights important feature maps with a channel-based attention mechanism. Given an input tensor $X \in \mathbb{R}^{T \times C}$, the mathematical formulation of this mechanism is provided in Equations (57)–(59).

Squeeze, global average pooling;

$$z_c = \frac{1}{T} \sum_{t=1}^T X_{t,c}, \quad z \in \mathbb{R}^C \quad (57)$$

Excite, fully connected network;

$$s = \sigma(W_2 \cdot \text{ReLU}(W_1 \cdot z)), \quad s \in \mathbb{R}^C \quad (58)$$

Rescaling;

$$Y_{t,c} = X_{t,c} \cdot s_c \quad (59)$$

where σ is the sigmoid activation.

CBAM: Convolutional Block Attention Module (1D)

CBAM computes attention in both channel and spatial dimensions. Channel attention consists of channel summary, dense layer and attention score, rescaling and is expressed by the following Equations (60)–(62).

Channel summary;

$$\text{avg}_c = \frac{1}{T} \sum_{t=1}^T X_{t,c}, \quad \text{max}_c = \max_t X_{t,c} \quad (60)$$

Attention score with dense layer;

$$M_c = \sigma(W_2 \cdot \text{ReLU}(W_1 \cdot [\text{avg}, \text{max}])) \quad (61)$$

Rescaling;

$$X'_{t,c} = X_{t,c} \cdot M_c \quad (62)$$

Spatial attention consists of channel-wise mean and maximum operations, inclusive convolution, and a final output mapping. The mathematical formulation of the average and maximum operations in the channel dimension is provided in Equations (63)–(65).

$$\text{avg}_t = \frac{1}{C} \sum_{c=1}^C X'_{t,c}, \quad \text{max}_t = \max_c X'_{t,c} \quad (63)$$

Inclusive convolution;

$$M_s = \sigma(\text{Conv1D}([\text{avg}_t, \text{max}_t])) \quad (64)$$

Final output;

$$Y_{t,c} = X'_{t,c} \cdot M_{s,t} \quad (65)$$

Residual Attention Block (1D)

This block combines the Ghost + SE + CBAM components and implements a residual connection. Given input $X \in \mathbb{R}^{T \times C}$, with output size C' , the mathematical formulation of this combined block is presented in Equations (66)–(68).

$$F(X) = \text{CBAM}(\text{SE}(\text{Ghost}(X))) \quad (66)$$

$$F'(X) = \text{Conv1D}_{1 \times 1}(F(X)) \in \mathbb{R}^{T \times C} \quad (67)$$

$$Y = X + F'(X) \quad (68)$$

This structure ensures that the original information is preserved and important features are emphasized through attention mechanisms.

Double Branch Model (1D)

The model operates on Wavelet and GLCM vectors. Both arms consist of a structure including Ghost Module, SE Block, CBAM and Residual connection. The output representations of the two branches are combined and transmitted to the classification layer. For the Wavelet branch, the input tensor $X_{Wavelet} \in \mathbb{R}^{T \times C}$, is processed according to the formulations provided in Equations (69)–(72), while the GLCM branch the input tensor $X_{GLMC} \in \mathbb{R}^{T \times C}$ follows the mathematical expressions defined in Equations (73)–(76). Then, the merging and classification process is performed as described in Equations (77)–(78).

$$H_1 = \text{Conv1D}(X_{Wavelet}) \quad (69)$$

$$H_2 = \text{ResidualAttentionBlock}(H_1) \quad (70)$$

$$H_3 = \text{GAP}(H_2) \quad (71)$$

$$H_4 = \text{Dense}(H_3) \quad (72)$$

$$L_1 = \text{Conv1D}(X_{GLCM}) \quad (73)$$

$$L_2 = \text{ResidualAttentionBlock}(L_1) \quad (74)$$

$$L_3 = \text{GAP}(L_2) \quad (75)$$

$$L_4 = \text{Dense}(L_3) \quad (76)$$

Then, the merging and classification process is performed.

$$Z = \text{Concat}(H_4, L_4) \quad (77)$$

$$\hat{y} = \text{Softmax}(\text{Dense}(Z)) \quad (78)$$

Feature Fusion

The outputs of each arm are combined into a single representation.

$$F_{concat} = \text{Concat}(F_{RGB}, F_{MS}, F_{Wavelet}, F_{GLCM}) \in \mathbb{R}^{d_1+d_2+256+256} \quad (79)$$

The merged feature representation F is given to the following classification layers.

$$h = \text{Dropout}(\text{ReLU}(W_1 F + b_1)) \in \mathbb{R}^{1024} \quad (80)$$

$$\hat{y} = \text{Softmax}(W_2 h + b_2) \in \mathbb{R}^n \quad (81)$$

where $W_1 \in \mathbb{R}^{1024 \times (d_1+d_2+512)}$, $W_2 \in \mathbb{R}^{n \times 1024}$ and $\hat{y}$ are the class probabilities predicted by the model. Output $\hat{y}$, target class, Sparse Categorical Crossentropy is calculated by comparing with $y \in \{0, 1, 2, \dots, n\}$.

$$\mathbb{L}(y, \hat{y}) = -\log(\hat{y}_y) \quad (82)$$

The optimization is performed using the Adam algorithm. The mathematical formulations of this classification module are presented in Equations (79)–(82).

Algorithm 2 shows the pseudo code of the classification model. The blue shaded parts are our proposed model combinations.

Algorithm 2. Pseudocode of the proposed classification model.**Input***N* sample images: I_{RGB} : RGB images, shape (128, 128, 3) I_{MS} : Multispectral images, shape (128, 128, 6) $F_{GLCM}, F_{wavelet}$: 1D feature vectors (extracted by GLCM, wavelet)**Output**

Class predictions (5 classes, softmax)

(XGghost-SECBAM-r-QB)**Step 1: Branch Construction***for each input type do***RGB branch:**Pass I_{RGB} through Xception backbone (ImageNet pre-trained) $\leftarrow$ Equations (30)–(39) and (41)Apply Residual Attention Block (Ghost, SE, CBAM) $\leftarrow$ Equations (50)–(53)Apply Global Average Pooling (GAP) $\leftarrow$ Equation (35)**MS branch:**Pass I_{MS} through Xception backbone (untrained) $\leftarrow$ Equations (30)–(40)Apply Residual Attention Block (Ghost, SE, CBAM) $\leftarrow$ Equations (50)–(53)Apply GAP $\leftarrow$ Equation (35)**GLCM branch:**Pass F_{GLCM} through 1D Conv + activation $\leftarrow$ Equations (73)–(76)Apply Residual Attention Block (Ghost, SE, CBAM) $\leftarrow$ Equations (66)–(68)Apply GAP + Dense + Dropout $\leftarrow$ Equations (77)–(78)**Wavelet branch:**Pass $F_{wavelet}$ through 1D Conv + activation $\leftarrow$ Equations (69)–(72)Apply Residual Attention Block (Ghost, SE, CBAM) $\leftarrow$ Equations (66)–(68)Apply GAP + Dense + Dropout $\leftarrow$ Equations (77)–(78)*end for***Step 2: Feature Fusion**

Concatenate feature vectors from all branches:

 $F_{fusion} = [F_{RGB}, F_{MS}, F_{GLCM}, F_{wavelet}] \leftarrow$ Equation (79)**Step 3: Final Classification**Pass F_{fusion} through:Dense layer (1024, ReLU) $\leftarrow$ Equation (80)

Dropout (0.5)

Output Dense layer (5, Softmax) $\leftarrow$ Equation (81)**Step 4: Model Training**Compile model with optimizer = Adam, loss = SparseCategoricalCrossentropy, metrics = [accuracy] $\leftarrow$ Equation (79)**Train model using multi-input data:** $[I_{RGB}, I_{MS}, I_{GLCM}, I_{wavelet}], y$ **Step 5: Evaluation**

Predict classes for test data

Report accuracy, confusion matrix, etc.

Algorithm 2 outlines the operational flow of the proposed multi-branch classification model (XGghost-SECBAM-r-QB). In Step 1, four input modalities (RGB images, multispectral images, GLCM features, and wavelet features) are processed in parallel through dedicated branches. RGB and MS inputs are passed through Xception backbones (pre-trained for RGB, untrained for MS), followed by a Residual Attention Block integrating Ghost, SE, and CBAM modules, and finalized with Global Average Pooling. The GLCM and wavelet

branches apply 1D convolution and activation, the same Residual Attention Block, and GAP with Dense + Dropout regularization. In Step 2, all branch outputs are concatenated to form a unified fusion feature vector. Step 3 performs final classification using a fully connected layer (1024 units, ReLU + Dropout) and a Softmax output for five classes. Step 4 trains the model using Adam and Sparse Categorical Cross Entropy within a multi-input learning scheme, while Step 5 evaluates performance through class predictions, accuracy, and confusion matrix analysis.

4. Experimental Results and Analysis

All comparative experiments followed the same protocol. The proposed and benchmark models used the same data splits, preprocessing, normalization, input size, optimizer family, maximum epoch limit, and evaluation metrics. The threshold-based stopping criterion was applied consistently to all models as a convergence indicator. Therefore, epoch and training-time values show when each model reached the predefined criterion, not an unequal training budget. This design made the comparison focused on architecture rather than data or protocol differences.

This section presents the experimental results. First, the datasets and settings are described. Then, segmentation and classification results are reported. Ablation experiments are used to evaluate the contribution of the proposed components. The same protocol was applied to segmentation baselines, including DeepLabV3+, HRNet, PSPNet, and UPerNet, and to transfer-learning-based classification baselines.

For readability, the results are presented in this order: segmentation results, classification comparison, ablation experiments, and stability analysis. Detailed run-by-run stability results are provided in the Supplementary Materials.

4.1. Datasets and Settings

Balanced training and test sets were created from RGB and multispectral images for five classes. Images were resized to 128×128 , normalized, and balanced using data augmentation. The benchmark segmentation stability experiment used a mixed PlantVillage and PlantDoc dataset. All other stability and ablation experiments used the newly introduced Water Stress 2025 and Common Rust 2025 datasets.

4.1.1. Datasets

Two UAV-based datasets were used for five-class maize stress and disease analysis: clean soil, healthy plant, common rust disease, low water stress, and high water stress. The datasets included RGB three-channel and six-channel multispectral patches with the same class definitions and preprocessing protocol. All images were resized to 128×128 pixels, normalized to $[0-1]$, and encoded with numeric labels.

For six-channel multispectral images, the training set included 1212, 601, 1044, 1015, and 872 samples for clean soil, healthy plant, rust disease, low water stress, and high water stress, respectively. For RGB images, the corresponding numbers were 1213, 589, 1090, 848, and 992. The test sets included approximately 200 samples per class. To reduce class imbalance, 400 original and 400 augmented samples were selected per class where applicable. Augmentation used horizontal flipping and random rotation.

This design enables a controlled comparison between RGB and multispectral inputs. It also allows the effect of spectral diversity on maize health, water stress, and rust detection to be examined. Detailed class-distribution plots are provided in Supplementary Figure S5.

4.1.2. Experimental Settings

- In the mask generation phase for graded water stress, a multi-stage workflow was followed to segment different plant health states and soil areas from multispectral images. Firstly, vegetation indices such as SAVI and NDRE were calculated using NIR, Red and Red Edge bands. In SAVI calculation, the soil adjustment constant $L = 0.5$ was chosen. When normalizing the bands, “robust” normalization methods were used for NIR and Red and “standard” normalization methods were used for NDRE. Subsequently, an automatic threshold value was determined via SAVI with the Otsu thresholding algorithm and soil zones were separated. In the determination of plant health levels, the K-Means algorithm was applied on NDRE and multiple thresholding was performed over four cluster centers ($n_clusters = 4$). The threshold centers were used to distinguish between low water stress, high water stress and healthy plants. The generated mask was subjected to morphological opening and closing operations with a 5×5 kernel to remove noise and small false regions. In addition, a minimum area threshold of 500 pixels was set to eliminate very small areas. After segmentation, Fully Connected Conditional Random Field (CRF) based post-processing was applied to ensure the spatial and spectral integrity of the mask and to correct the boundaries more precisely. In CRF: $sdims$ as spatial dimension in Pairwise Gaussian filter = (3, 3), in the Pairwise Bilateral filter, spatial $sdims = (30, 30)$, spectral $schan = (13, 13, 13)$ values were selected. The number of iterations in the CRF inference is set to 100. With this multi-stage workflow, both automatic thresholding and CRF assisted post-processing resulted in more accurate and smoother segmentation masks at the edge lines.
- In the mask generation phase for common rust disease, a multi-stage workflow based on vegetation indices such as SAVI, NDRE, NDVI and SR was developed for automatic segmentation of plant and soil areas. NIR and Red bands were normalized by RobustScaler and StandardScaler methods. Automatic thresholding with K-Means ($n_clusters = 2$) was performed on the SAVI index for soil and plant separation, and morphological operations with 5×5 kernels were applied. Phytosanitary levels were determined by K-Means ($n_clusters = 3$) on NDRE, NDVI and SR indices, and these masks were combined by maximum value method. Finally, Fully Connected CRF algorithm was applied with Pairwise Gaussian ($sdims = 5$) and Bilateral ($sdims = 10$, $schan = 5$) parameters and 100 iterations. In this way, segmentation masks with high accuracy and edge sensitivity were obtained.
- For the segmentation, the choice of the network architecture and parameters, influenced by the ablation experiments, was made only to isolate the effect of skip connection attention and residual blocks on the UNet++ structure. The input size of the model was set to $128 \times 128 \times 6$, indicating that six-band multispectral images were studied. In the encoder and decoder stages, residual blocks were used, each with LeakyReLU ($\alpha = 0.1$) activation, Batch Normalization and an increasing number of filters (64, 128, 256, 512, 1024). Dropout (ratio = 0.3) was applied in the deep layers to reduce the risk of overfitting.

On the decoder side, in addition to the classical UNet++ skip connections, Skip Connection Attention Block is implemented. In this block, the feature maps from the encoder and the upgraded maps from the decoder are reduced to the same number of channels by 1×1 convolution and then the attention map is obtained by sigmoid activation. The attention map was multiplied by the feature maps from the skip and decoder branches to emphasize both spatial and cross channel important information.

In the last layer of the model, 1×1 convolution and softmax activation were used for three class segmentation. Spatial dimension reduction with MaxPooling2D (2×2) and resolution enhancement with UpSampling2D (2×2) were used throughout the network.

- For classification, a four-arm deep learning classification model is designed that integrates different types of images and features. There are four different inputs in the model: RGB images of size $128 \times 128 \times 3$, multispectral images of size $128 \times 128 \times 6$, and two separate GLCM/wavelet feature vectors of size $(n,1)$ each. The RGB and multispectral images are processed through Xception-based deep branches, pre-trained with ImageNet and initialized from scratch, respectively. In these branches, at the output of the basic Xception network, the Ghost Module, which uses channels efficiently, the Squeeze and Excite (SE) block, which strengthens channel associations, and the Residual Attention Block, in which the CBAM block, which adds both channel and spatial attention, is applied sequentially. Residual connection was achieved by summing the input and output. After Xception, the feature map was converted into a vector using Global Average Pooling. GLCM/WAVELET feature vectors are processed with Conv1D, then passed through a 1D Residual Attention Block including Ghost, SE and CBAM blocks, then compressed with GlobalAveragePooling1D and a 256 neuron Dense layer. Dropout of 50% was applied at the output of both arms. The four separate feature vectors obtained are combined with Concatenate layer and then passed through a Dense layer with 1024 neurons and ReLU activation and 50% Dropout. In the final stage, the output of the model is a softmax activated Dense layer with 5 classes. The Adam optimization algorithm, sparse categorical cross-entropy loss, and accuracy metric were used during training. In each layer of the model, parameters such as filter, neuron, ratio are optimized according to the literature and experimental results. Thanks to this integrated architecture, high accuracy classification performance is aimed by joint learning of multiple data types.
- The methods proposed in the study, namely masking methods and artificial intelligence algorithms, were implemented in the Anaconda based **PyCharm Community Edition 2021.3.3** IDE using libraries such as TensorFlow, Numpy, scikit-learn, Pandas, PIL, imageio, os, cv2, matplotlib, rasterio and sklearn. Implemented on a system with Nvidia RTX 4070, 12 GB GPU, 64 GB DDR5 RAM and Intel Core i5-12600K processor. It has a softmax activation function using the dynamic Adam optimizer with batch size set to 16. The maximum number of epochs was set to 100 for all compared classification models. The same threshold-based convergence protocol was used in all classification experiments. The validation-accuracy threshold only determined when a model reached the practical convergence criterion. Models that did not reach it continued until the maximum epoch limit. Thus, best epoch and training duration values reflect convergence behavior under a common protocol. The patience coefficient was set to 2.

4.2. Comparative Baselines for Segmentation

The results include evaluations of the proposed model in terms of accuracy, MeanIoU and various parameters compared with widely used general-purpose segmentation architectures in both 4 class water stress and 3 class common rust segmentation tasks.

4.2.1. Segmentation Comparison for Water Stress Dataset

DeepLabV3+, HRNet, PSPNet, and UPerNet were used as benchmark segmentation models. These models represent different segmentation designs and allow the proposed model to be evaluated beyond accuracy on the Water Stress dataset.

When Table 3 is analyzed, the experimental comparisons performed on the Water Stress 2025 dataset show that the proposed model achieved the most favorable segmentation performance among the compared architectures under the adopted experimental protocol. Although the strongest benchmark model was UPerNet, with 0.9094 validation ac-

curacy and 0.6546 Mean IoU, the proposed model achieved 0.9388 validation accuracy and 0.7723 Mean IoU.

Table 3. Comparison of widely used benchmark segmentation architectures with the proposed segmentation model on the Water Stress dataset.

Metric Model	Val Accuracy	True MeanIoU	Time (s)	Total Number of Parameters	Best Epoch
DeepLabV3+	0.8861	0.6032	349.5012	10,127,428	91
HRNet	0.8961	0.6573	1640.6156	1,899,460	92
PSPNet	0.8636	0.4921	166.0185	1,341,124	29
UPerNet	0.9094	0.6546	890.8918	6,997,716	80
Proposed model	0.9388	0.7723	1982.5330	15,983,368	38

The improvement is particularly evident in the Mean IoU metric. The proposed model increased Mean IoU from 0.6546 obtained by UPerNet to 0.7723 on the Water Stress 2025 dataset. This corresponds to an absolute gain of 0.1177 Mean IoU. The proposed architecture required 1982.5330 s and contained 15,983,368 parameters, indicating that the improved segmentation performance was achieved with an acceptable computational cost under the adopted experimental protocol.

4.2.2. Segmentation Comparison for Common Rust Dataset

For the Common Rust dataset, DeepLabV3+, HRNet, PSPNet, and UPerNet were used as reference architectures. These models represent different semantic segmentation paradigms.

When Table 4 is analyzed, the evaluations conducted on the Common Rust 2025 dataset show that the proposed model achieved the highest segmentation performance among the compared benchmark architectures. UPerNet was the strongest benchmark model with 0.8426 validation accuracy and 0.6520 Mean IoU, whereas the proposed model achieved 0.9648 validation accuracy and 0.9164 Mean IoU.

Table 4. Comparison of widely used benchmark segmentation architectures with the proposed segmentation model on the Common Rust dataset.

Metric Model	Val Accuracy	True MeanIoU	Time (s)	Total Number of Parameters	Best Epoch
DeepLabV3+	0.8187	0.5833	190.9284	10,127,171	16
HRNet	0.8316	0.6232	730.1538	1,899,395	23
PSPNet	0.7871	0.5573	118.5692	1,340,995	38
UPerNet	0.8426	0.6520	482.9027	6,997,716	36
Proposed model	0.9648	0.9164	911.6780	15,983,303	87

In terms of Mean IoU, the proposed model increased performance from 0.6520 obtained by UPerNet to 0.9164 on the Common Rust 2025 dataset. This corresponds to an absolute gain of 0.2644 Mean IoU. The proposed architecture required 911.6780 s and contained 15,983,303 parameters, while providing substantially stronger pixel-level class separation for soil, rust-affected vegetation, and healthy maize regions under real-field UAV conditions.

4.3. Comparative Baselines and Complexity-Aware Interpretation for Classification

Widely used transfer-learning models were used as classification baselines. These included VGG, ResNet, DenseNet, Inception, EfficientNet, and Xception. These models represent different architecture types and provide strong classification-only alternatives un-

der the same UAV-based five-class protocol. The comparison was not used only to report the highest accuracy. It was also used to test whether the proposed multimodal classifier gives a meaningful practical advantage when accuracy, parameter count, convergence behavior, and preprocessing complexity are considered together. The modality-controlled experiments were used to isolate the effect of input modality and to clarify the contribution of spectral information.

The proposed XGhost-SECBAM-r-QB classifier achieved the highest accuracy under the adopted protocol. However, its gain over the strongest baseline was small. The proposed model achieved 0.9954 accuracy, while DenseNet201 achieved 0.9931 with fewer parameters. Therefore, the classification results should be interpreted together with model complexity. The proposed classifier should be seen mainly as a task-specific multimodal analysis model, not as a model that gives a large accuracy gain over simpler baselines.

As shown in Table 5, the absolute accuracy difference between the proposed classifier and DenseNet201 was only 0.0023, or 0.23 percentage points. However, the proposed classifier used 97,977,973 parameters, while DenseNet201 used 20,294,213 parameters. This means that the proposed classifier used about 4.83 times more parameters. Therefore, this small accuracy gain does not by itself justify the full classifier for classification-only deployment.

Table 5. Comparison of widely used transfer-learning classification baselines with the evaluated multimodal classification model. The table reports accuracy, F1-score, precision, sensitivity, training time, parameter count, and threshold-based convergence epochs under the adopted UAV-based five-class protocol.

Metric Model	Accuracy	F1 Score	Precision	Sensitivity	Time (s)	Total Number of Parameters	Total Number of Epoch
VGG16	0.1894	0.0603	0.0359	0.1894	1318.21	15,245,125	100
VGG19	0.2088	0.0603	0.0359	0.1894	1577.44	20,554,821	100
ResNet50	0.9800	0.9800	0.9807	0.9800	1039.50	25,691,013	100
ResNet101	0.9856	0.9856	0.9858	0.9856	1985.40	44,761,477	100
ResNet152	0.9644	0.9643	0.9666	0.9644	2729.32	60,474,245	100
DenseNet121	0.9888	0.9869	0.9869	0.9869	1453.23	8,092,229	100
DenseNet169	0.9881	0.9882	0.9883	0.9881	2234.03	14,352,965	100
DenseNet201	0.9931	0.9931	0.9933	0.9931	2269.00	20,294,213	100
InceptionV3	0.9863	0.9850	0.9853	0.9850	1133.32	23,906,085	100
InceptionResNetV2	0.9763	0.9750	0.9752	0.9750	2277.44	55,915,749	100
EfficientNetB0	0.2106	0.0883	0.0917	0.2106	860.45	5,366,440	100
EfficientNetB4	0.2406	0.1297	0.2631	0.2406	1857.21	19,514,980	100
EfficientNetB7	0.3950	0.3015	0.5596	0.3963	3989.34	66,725,276	100
Xception	0.9806	0.9813	0.9814	0.9812	1191.22	22,964,781	100
Proposed Model	0.9954	0.9954	0.9954	0.9954	1122.43	97,977,973	25

The faster threshold-based convergence of the proposed classifier should also be interpreted carefully. It only shows that the model reached the predefined validation-accuracy criterion in fewer epochs under the offline training protocol. It does not prove lower deployment complexity or lower inference cost. Therefore, the epoch and time values are used as convergence indicators, not as proof that the full classifier is computationally preferable for operational use.

Overall, the proposed classifier achieved highly competitive performance compared with the CNN-based baselines. However, the comparison should be interpreted with caution. The accuracy improvement over strong baselines was small. Therefore, the main purpose of this comparison is not to claim practical superiority over simpler classification-only models. Instead, it is used to evaluate the accuracy–complexity trade-off and to support the branch-level and modality-controlled analyses.

To make the multimodal comparison fairer, additional modality-controlled experiments were conducted using RGB-only, six-channel multispectral-only, and fused RGB + multispectral inputs over 21 independent runs. These results are given in Supplementary Section S12 and summarized in Supplementary Table S10.

The accuracy difference between the proposed classifier and strong baselines such as DenseNet201 and Xception was small because these models already performed well on the five-class task. Therefore, accuracy alone is not enough to justify the full classification branch. The ablation, modality-controlled runs, and stability results show how RGB, multispectral, Wavelet, and GLCM inputs affect accuracy, convergence, and robustness. They also show that the full classifier is not always preferable. For classification-only deployment, simpler alternatives may be more practical, while the proposed classifier should mainly be interpreted as a modality-contribution and convergence-analysis component.

4.4. Results of Ablation Experiments for Segmentation Model

Ablation experiments were used to find the most balanced segmentation configuration. Four modules, Ghost, SE, Skip-Attention, and Residual blocks, were evaluated in the proposed dual-branch UNet++ segmentation model, hereafter referred to as scar-U-Net++ DB framework. All module combinations were tested under the same protocol for Water Stress 2025 and Common Rust 2025. The best segmentation performance was obtained with Skip-Attention and Residual blocks together. Skip-Attention improved spatial information transfer and class boundaries, while Residual blocks improved learning stability. Ghost and SE did not provide consistent gains and sometimes increased complexity. Therefore, the final model retained only Skip-Attention and Residual components. The ablation results show that the improvement came from the combined effect of attention-enhanced skip connections and residual refinement. Friedman ranking was also used to evaluate the relative contribution of the components. The detailed formulation and full ranking tables are provided in Supplementary Section S3 and Table S3a,b. The Friedman rankings and the Nemenyi post hoc critical difference comparison were used as auxiliary statistical evidence, while the final configuration was selected by jointly considering validation Mean IoU, architectural simplicity, parameter efficiency, and consistency across both datasets. A Nemenyi post hoc critical difference comparison was also added after the Friedman ranking analysis in Supplementary Section S6. For the Water Stress 2025 dataset, the rank difference between the best-ranked configuration and the selected final Skip-Attention + Residual configuration was below the critical difference, indicating no statistically significant difference. For the Common Rust 2025 dataset, the selected final configuration was already the best-ranked configuration. Therefore, the final segmentation architecture was retained because it provided statistically comparable performance on Water Stress 2025, the best ranking on Common Rust 2025, and a balanced trade-off in terms of Mean IoU, architectural simplicity, parameter efficiency, and consistency across both segmentation tasks.

These findings show that SkipAttention and Residual blocks are important for UNet++ segmentation, while Ghost and SE do not provide clear advantages for this dataset. Detailed results are given in Tables 6 and 7.

Table 6. Ablation experiment of our proposed segmentation model for water stress.

Combination	SkipAttention	Residual	Val Accuracy	True MeanIoU	Time (s)	Total Number of Parameters	Best Epoch
Model-1	-	-	0.9360	0.7184	1728.55	58,117,252	30
Model-2	-	+	0.9353	0.7215	1920.28	61,432,516	56
Model-3	+	-	0.9349	0.7194	1890.54	59,688,904	51
Model-4	+	+	0.9388	0.7723	1982.5330	15,983,368	38

Table 7. Ablation experiment of our proposed segmentation model for common rust.

Combination	SkipAttention	Residual	Val Accuracy	True MeanIoU	Time (s)	Total Number of Parameters	Best Epoch
Model-1	-	-	0.8823	0.7422	1470.71	58,117,187	50
Model-2	-	+	0.9055	0.7889	1651.43	61,432,451	51
Model-3	+	-	0.8797	0.7353	1591.37	59,688,839	50
Model-4	+	+	0.9648	0.9164	911.6780	15,983,303	87

4.4.1. Segmentation Ablation for Water Stress Dataset

This section presents the water-stress ablation results. The experiments were designed to determine which blocks should be included in the model and how each component affected performance.

When Table 6 is analyzed, the ablation analysis shows that the combined use of SkipAttention and Residual blocks provided the highest segmentation performance among the tested configurations. When both components were used together (Model-4), the model achieved 0.9388 validation accuracy and 0.7723 Mean IoU on the Water Stress 2025 dataset. This confirms that attention-enhanced skip connections and residual feature refinement act complementarily, improving both spatial information transfer and learning stability.

4.4.2. Segmentation Ablation for Common Rust Dataset

This section presents the detailed findings of the ablation studies conducted using the Common Rust dataset. The experiments played a decisive role in determining which blocks should be incorporated into the model. In this part, the results of comprehensive analyses performed on the selected blocks are reported, and the contribution of each component to the overall performance of the model is examined in detail.

When Table 7 is analyzed, the Common Rust 2025 ablation results show that the combination of SkipAttention and Residual blocks provided the highest segmentation performance. The final Model-4 configuration achieved 0.9648 validation accuracy and 0.9164 Mean IoU. This result indicates that the Residual block improved optimization stability, while SkipAttention enhanced the transfer of spatially important features for rust-affected and healthy vegetation separation.

4.5. Results of Ablation Experiments for Classification Model

In this section, the contributions of the proposed classification architecture were analyzed under two complementary ablation settings. In the first setting, the input branch configuration was kept fixed, and only the individual and combined effects of the architectural blocks (Ghost, SE, CBAM, and Residual) were examined. In the second setting, the architectural block configuration was kept fixed, and only the contributions of different input branch combinations (RGB, multispectral, Wavelet, and GLCM) were evaluated. This design allowed us to separately assess block-level contributions and input-representation-level contributions.

Accuracy, Precision, Recall, and F1-score were calculated independently using scikit-learn. Precision, Recall, and F1-score were reported as weighted averages. Because the dataset was nearly balanced and the models made few errors, these values became very close to Accuracy. After rounding, they may appear identical in some rows. This reflects weighted averaging and balanced class support, not duplicated reporting.

In addition, all classification metrics were recalculated and cross-checked from the confusion matrix to ensure that the reported values did not result from copying, averaging mismatch, or reporting errors.

4.5.1. Block Ablation Under Fixed Input Branches

The effects of Ghost, SE, CBAM, and Residual blocks were tested while the input branches were kept fixed. Table 8 shows that several block combinations produced very similar classification results. This indicates that the dataset was highly separable under the adopted protocol. The full Ghost + SE + CBAM + Residual configuration reached the same peak accuracy as some simpler configurations. However, it required fewer epochs and shorter offline training time. Therefore, this ablation does not show that the full block configuration is necessary for peak accuracy. Instead, it shows the trade-off between architectural complexity and threshold-based convergence behavior.

Table 8. Results of the block ablation experiment for the evaluated classification configuration, where the input branches were kept fixed. Precision, Sensitivity/Recall, and F1-score are reported as weighted-average metrics. The table is intended to analyze block contribution and convergence behavior rather than to show that the most complex configuration is necessary for peak accuracy.

Metric Model	Accuracy	F1 Score	Precision	Sensitivity	Time (s)	Total Number of Parameters	Total Number of Epoch
Ghost_Only FixedBranches	0.9966	0.9966	0.9966	0.9966	524.15	84,291,701	19
SE_Only FixedBranches	0.9954	0.9954	0.9954	0.9954	145.02	48,590,949	6
CBAM_Only FixedBranches	0.9965	0.9966	0.9966	0.9966	336.88	49,642,885	14
Residual_Only FixedBranches	0.9954	0.9954	0.9955	0.9954	454.29	54,900,277	20
Ghost + SE FixedBranches	0.9977	0.9977	0.9977	0.9977	581.31	86,400,037	21
Ghost + CBAM FixedBranches	0.9954	0.9954	0.9954	0.9954	376.79	87,451,973	13
Ghost + Residual FixedBranches	0.9966	0.9966	0.9966	0.9966	369.60	92,709,365	13
SE + CBAM FixedBranches	0.9966	0.9966	0.9966	0.9966	321.57	51,751,221	13
SE + Residual FixedBranches	0.9966	0.9966	0.9966	0.9966	466.41	57,008,613	20
CBAM + Residual FixedBranches	0.9954	0.9955	0.9956	0.9954	391.65	58,060,549	16
Ghost + SE + CBAM FixedBranches	0.9977	0.9977	0.9977	0.9977	700.20	89,560,309	24
Ghost + SE + Residual FixedBranches	0.9954	0.9954	0.9955	0.9954	403.17	94,817,701	14
Ghost + CBAM + Residual FixedBranches	0.9954	0.9955	0.9956	0.9954	387.33	95,869,637	13
SE + CBAM + Residual FixedBranches	0.9966	0.9966	0.9966	0.9966	303.71	60,168,885	12
Ghost + SE + CBAM + Residual FixedBranches	0.9977	0.9977	0.9977	0.9977	301.44	97,977,973	10

In this experimental setting, the term FixedBranches indicates that the same input branch configuration was preserved across all experiments, while only the block combinations were changed.

Table 8 shows that the tested blocks affected both accuracy and convergence. Some configurations reached the same peak accuracy. For example, Ghost + SE, Ghost + SE + CBAM, and Ghost + SE + CBAM + Residual all achieved 0.9977 accuracy. Therefore, the full block configuration should not be seen as giving a large accuracy gain over simpler combinations.

From a classification-only deployment perspective, the most complex block configuration is not necessary when the goal is only peak accuracy. The full Ghost + SE + CBAM + Residual configuration reduced the number of epochs and offline training time. However, the lighter Ghost + SE configuration achieved the same peak accuracy with lower architectural complexity. Therefore, the full configuration should be seen as an offline convergence-oriented option, not as a default low-complexity classifier for operational use.

Its main advantage was faster threshold-based convergence during offline training. The full configuration reached 0.9977 accuracy in 301.44 s and 10 epochs. Ghost + SE needed 581.31 s and 21 epochs, while Ghost + SE + CBAM needed 700.20 s and 24 epochs. This shows that the full block design mainly improved convergence speed, not final accuracy. Therefore, the final block configuration should be interpreted as a convergence-ori-

ented design for block-contribution analysis, not as proof of a practically superior classification-only model.

4.5.2. Branch Ablation Under Fixed Architectural Blocks

The branch-level ablation was conducted to test whether each input branch adds useful information beyond RGB and multispectral inputs. In this experiment, the block configuration was fixed, and only the input branches were changed. This analysis is important because the full classifier uses extra Wavelet and GLCM preprocessing steps. These steps may increase preprocessing time during deployment. Therefore, the branch-level ablation was evaluated not only by peak accuracy, but also by convergence behavior and deployment simplicity.

The branch-level ablation results in Table 9 show that RGB and multispectral branches are the main contributors to classification performance. Wavelet-only and GLCM-only models gave lower results. This shows that these handcrafted features are not strong enough when used alone. The multispectral branch performed better than the RGB branch, which shows the importance of spectral information for separating healthy maize, water stress, and common rust.

Table 9. Results of the branch-level ablation experiment for the evaluated classification configuration, where the block configuration was kept fixed and only the input branches were changed. The table is intended to identify whether additional Wavelet and GLCM branches improve final accuracy or mainly affect offline convergence.

Metric Model	Accuracy	F1 Score	Precision	Sensitivity	Time (s)	Total Number of Parameters	Total Number of Epoch
Wavelet FixedBlocks	0.8526	0.8523	0.8541	0.8526	258.28	324,866	100
GLCM FixedBlocks	0.8194	0.8126	0.8182	0.8194	257.75	324,866	100
MS FixedBlocks	0.9863	0.9863	0.9864	0.9863	1323.85	48,673,776	100
RGB FixedBlocks	0.9737	0.9735	0.9736	0.9737	1300.37	48,672,912	100
GLCM + Wavelet FixedBlocks	0.9143	0.9131	0.9150	0.9143	437.713	643,583	100
MS + Wavelet FixedBlocks	0.9909	0.9909	0.9909	0.9909	1439.31	48,992,493	100
RGB + Wavelet FixedBlocks	0.9783	0.9782	0.9786	0.9783	1410.58	48,991,629	100
RGB + GLCM FixedBlocks	0.9771	0.9771	0.9774	0.9772	1414.93	48,991,629	100
RGB + MS FixedBlocks	0.9954	0.9954	0.9954	0.9954	1016.75	97,340,539	38
MS + GLCM FixedBlocks	0.9863	0.9863	0.9863	0.9863	1429.94	48,992,493	100
RGB + MS + GLCM FixedBlocks	0.9954	0.9954	0.9954	0.9954	427.65	97,659,256	15
RGB + MS + Wavelet FixedBlocks	0.9954	0.9954	0.9954	0.9954	484.26	97,659,256	17
RGB + GLCM + Wavelet FixedBlocks	0.9771	0.9771	0.9780	0.9771	1529.61	49,310,346	100
MS + GLCM + Wavelet FixedBlocks	0.9909	0.9908	0.9908	0.9909	1561.46	49,311,210	100
RGB + MS + GLCM + Wavelet FixedBlocks	0.9954	0.9954	0.9954	0.9954	361.38	97,977,973	12

The RGB + multispectral configuration reached the same peak accuracy as the full RGB + multispectral + GLCM + Wavelet configuration. Both achieved 0.9954 accuracy. Therefore, Wavelet and GLCM did not improve final accuracy under the adopted protocol. Their main effect was on offline convergence. The RGB + multispectral configuration reached 0.9954 accuracy in 1016.75 s and 38 epochs, while the full four-branch configuration reached the same accuracy in 361.38 s and 12 epochs.

However, this convergence advantage does not mean that the full classifier is the most practical option for deployment. In real UAV use, inference is repeated on many patches, and Wavelet and GLCM extraction may increase preprocessing time. Therefore, for low-latency or hardware-limited classification, the simpler RGB + multispectral configuration may be more practical. The full four-branch classifier should be interpreted as

an offline convergence and modality-contribution analysis, not as a practically superior classification-only model.

4.6. Stability Analysis

The threshold used in stability analysis was not a model-specific early stopping rule. It was a common Practical Success Point used to check whether each model reached a predefined performance level. Models that failed to reach the threshold were still included, and this was reflected in the Success Rate. Thus, PSP, SR, ACD, and MEC describe reproducibility, convergence, and computational behavior.

Stability analysis evaluated reproducibility across repeated runs instead of relying on one trial. Because deep learning results can vary with initialization, data shuffling, and optimization, each experiment was repeated 21 times under the same protocol. True Mean IoU was used for segmentation, and validation Accuracy was used for classification. PSP, SR, ACD, and MEC were also used to evaluate success rate, convergence, and computational cost. Summarized results are reported in the main text, while full run-by-run results are provided in Supplementary Sections S7–S13. Metric definitions are given in Supplementary Section S18.

4.6.1. Stability Analysis with Water Stress and Common Rust Datasets for Segmentation

This section presents the stability analysis results obtained on the Water Stress 2025 and Common Rust 2025 segmentation datasets. The primary objective is to evaluate the consistency, reproducibility, and convergence behavior of the proposed segmentation model in comparison with both high-capacity and lightweight benchmark architectures under varying initialization conditions. All experiments were conducted over 21 independent runs. For the Water Stress 2025 and Common Rust 2025 segmentation stability analyses, the Practical Success Point (PSP) thresholds were recalculated from the updated 21-run results. The PSP threshold was 0.7139 for Water Stress 2025 and 0.7759 for Common Rust 2025. These thresholds were used to determine whether each independent run reached a successful stability criterion. These PSP values were used as threshold criteria to determine successful runs. The Success Rate (SR) represents the percentage of runs that reached or exceeded the PSP, the Average Computation Duration (ACD) denotes the mean training time of successful runs, and the Mean Epoch of Convergence (MEC) indicates the average number of epochs required to reach the PSP threshold. The same evaluation protocol was consistently applied across all evaluated models. In addition to the primary comparison models (Proposed model, HRNet, and UPerNet), the analysis was extended to include additional benchmark architectures such as UNet, SegNet, and SegFormer. While these models were not included in the PSP calculation, they were evaluated using the same PSP thresholds to ensure a fair and consistent comparison across all methods. Supplementary Tables S7–S9, the proposed segmentation model consistently achieved higher SR values and more favorable ACD–MEC trade-offs compared to all competing architectures. Although some baseline models such as UNet achieved high SR values, they required significantly higher convergence epochs ($\text{MEC} \approx 100$), indicating slower learning behavior. In contrast, SegNet failed to reach the PSP threshold under most conditions, while SegFormer demonstrated moderate stability with lower SR values in certain scenarios. As shown in Supplementary Tables S7–S9, the proposed segmentation model achieved $\text{SR} = 100.00\%$ on both Water Stress 2025 and Common Rust 2025 under the updated PSP thresholds. For Water Stress 2025, UNet reached the PSP threshold in 38.10% of the runs but required the full training budget, with $\text{MEC} = 100$. HRNet, UPerNet, SegNet, and SegFormer did not reach the updated PSP threshold. For Common Rust 2025, only the proposed model reached the PSP threshold. These findings indicate that the pro-

posed model provides the most stable segmentation performance under the updated 21-run stability protocol.

Overall, the results indicate that the proposed model provides the most balanced performance in terms of stability, convergence speed, and computational efficiency. These findings confirm that the proposed framework exhibits strong robustness, reproducibility, and reliable segmentation performance under different initialization conditions for both water stress detection and rust disease segmentation tasks.

4.6.2. Stability Analysis with Benchmark Dataset for Segmentation

This section presents a comparative stability evaluation on the benchmark segmentation dataset, which consists of a mixed dataset derived from PlantVillage and PlantDoc. The analysis focuses on the proposed model and the two best-performing benchmark architectures identified based on Mean IoU performance. All experiments were repeated over 21 independent runs to evaluate reproducibility and convergence stability under different initialization conditions. The Practical Success Point (PSP) was defined as the average Mean IoU of the top three algorithms, consistent with the definition used in the previous section. For the benchmark dataset, the PSP threshold was determined as $mIoU = 0.9478$. The corresponding SR, ACD, and MEC metrics were computed to assess model stability and efficiency. Specifically, SR represents the percentage of runs exceeding the PSP threshold, ACD denotes the average training time of successful runs, and MEC indicates the average number of epochs required to reach the PSP. The results, presented in Supplementary Tables S10 and S11, show that both the proposed model and HRNet reached $SR = 100.00\%$, whereas UPerNet did not reach the PSP threshold. However, the proposed model reached the threshold earlier than HRNet, with $MEC = 20.52$ and $ACD = 984.82$ s, compared with HRNet's $MEC = 37.10$ and $ACD = 1256.71$ s. This indicates more stable convergence behavior and improved computational efficiency under repeated training conditions. These findings confirm that the proposed segmentation framework maintains both stability and efficiency even under cross-domain conditions represented by the benchmark dataset.

4.6.3. Modality-Controlled Stability Analysis for Classification

The classification stability analysis was used to test whether the evaluated classifier gives consistent results under different input settings. These results should not be interpreted as evidence that the proposed classifier is always superior. Instead, they show how stability changes for RGB-only, multispectral-only, and fused RGB + multispectral inputs. The detailed 21-run results are given in Supplementary Section S12, and the summary SR, MEC, and ACD values are given in Supplementary Table S10.

The best model changed depending on the input modality. In the RGB-only setting, the proposed classifier achieved $SR = 100.00\%$. In the multispectral-only setting, DenseNet121 and DenseNet201 were more stable, with SR values of 85.71% and 80.95%, while the proposed classifier reached $SR = 28.57\%$. In the fused RGB + multispectral setting, DenseNet121 achieved the highest SR value of 47.62%, while the proposed classifier reached $SR = 14.29\%$. These results show that the proposed classifier is not uniformly superior across all input settings.

Although the proposed classifier had lower ACD in the fused setting, this only reflects threshold-based convergence under the offline protocol. It does not mean lower deployment complexity. In real UAV classification, preprocessing and inference costs must also be considered, especially when extra feature branches are used.

Overall, classification stability depends strongly on input modality and model architecture. Therefore, the classification component is best interpreted as a modality-contribution and convergence-analysis module. For classification-only deployment, simpler al-

ternatives such as DenseNet121, DenseNet201, or the RGB + multispectral configuration may be more suitable depending on the input setting and computational constraints.

4.7. Confusion Matrix and Learning-Curve Analysis

This section visually examines the classification results using the confusion matrix, accuracy curve, and loss curve. The confusion matrix shows how the evaluated classifier separates the five classes under the adopted test protocol. The accuracy and loss curves show the learning behavior during training and validation. These visual results support the within-dataset classification analysis. However, they do not prove that the full classifier is practically superior to simpler classification-only alternatives.

Figure 6 shows that the evaluated classifier learned the five-class task consistently under the adopted dataset split. The confusion matrix also shows concentrated predictions. However, these visual results should be interpreted together with the quantitative analyses. DenseNet201 achieved very similar accuracy with fewer parameters, and the RGB + multispectral configuration reached the same peak accuracy as the full four-branch classifier.

Therefore, Figure 6 supports the within-dataset performance of the evaluated classifier. However, it does not change the main interpretation that simpler classification-only alternatives may be more practical for deployment. The visual results are complementary evidence. The practical interpretation should mainly rely on the baseline comparison, branch ablation, and modality-controlled stability analysis, which show that the full classifier is competitive but not always preferable to simpler alternatives.

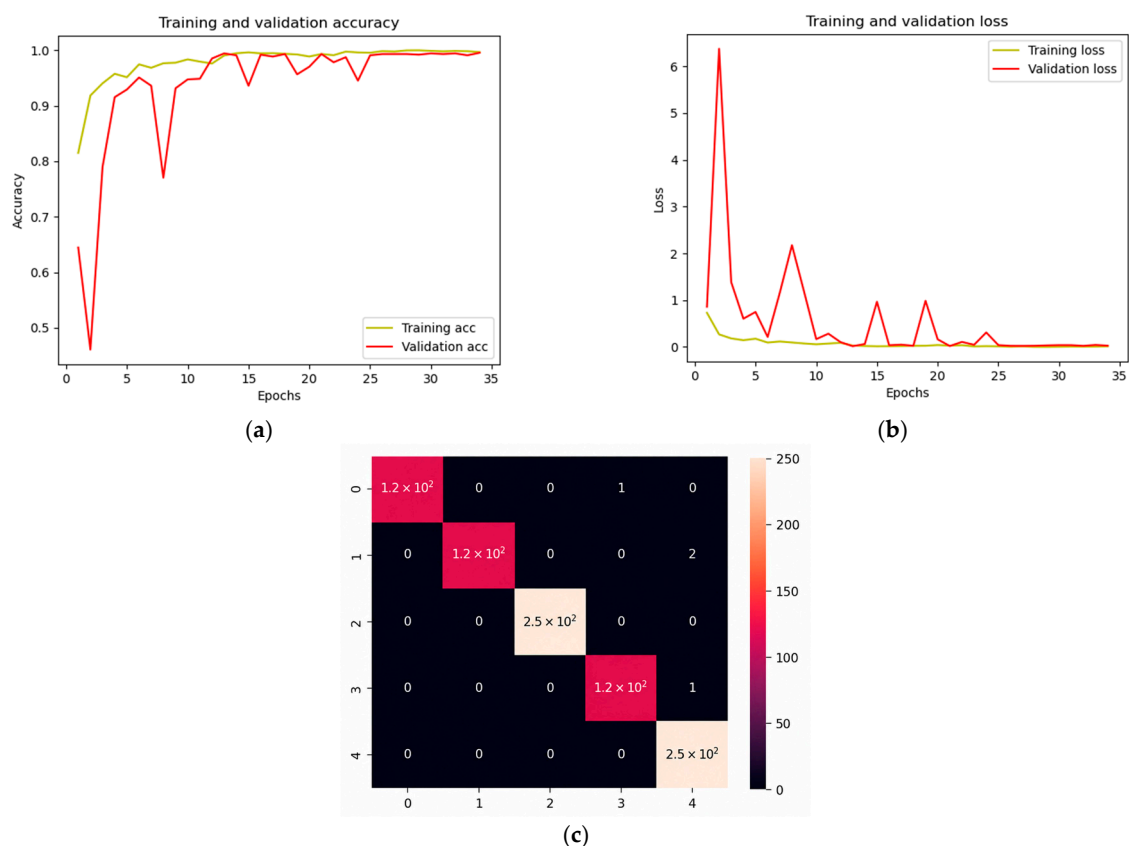


Figure 6. Classification learning curves and confusion matrix of the evaluated classifier under the adopted five-class UAV dataset protocol. (a) Accuracy curve, (b) loss curve, and (c) confusion matrix. These plots summarize within-dataset learning behavior and class-level prediction patterns; they should be interpreted together with the baseline comparison, branch-level ablation, and modality-controlled stability results.

4.8. Field-Scale Segmentation Mapping Results

Figure 7 Field-scale segmentation mapping results for the Water Stress 2025 dataset. The upper panel shows the index-guided reference mask, and the lower panel shows the predicted map. The classes are soil/background, low water stress, high water stress, and healthy maize. The visual agreement shows that the model can generate field-scale stress maps from UAV multispectral orthomosaics.

Figure 8 Field-scale segmentation mapping results for the Common Rust 2025 dataset. The upper panel shows the index-guided reference mask, and the lower panel shows the predicted map. The classes are soil/background, common rust, and healthy maize. The map shows that the proposed framework can support field-level disease monitoring under real UAV imaging conditions.

Index-Guided Reference Mask vs Predicted Segmentation Map

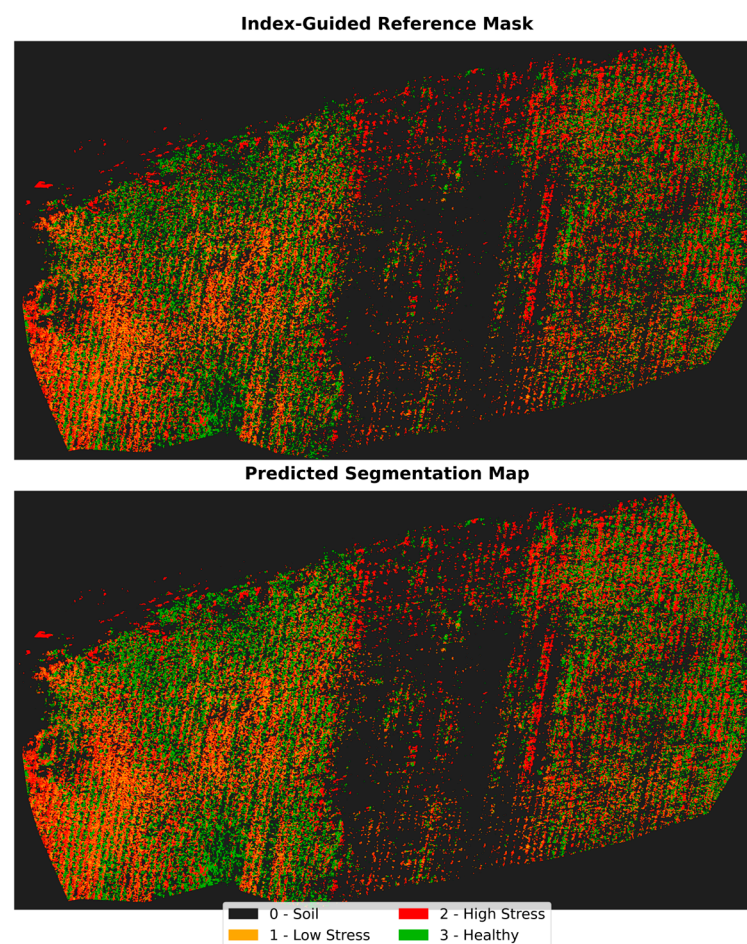


Figure 7. Field-scale segmentation mapping results for the Water Stress 2025 dataset.

Index-Guided Reference Mask vs Predicted Segmentation Map

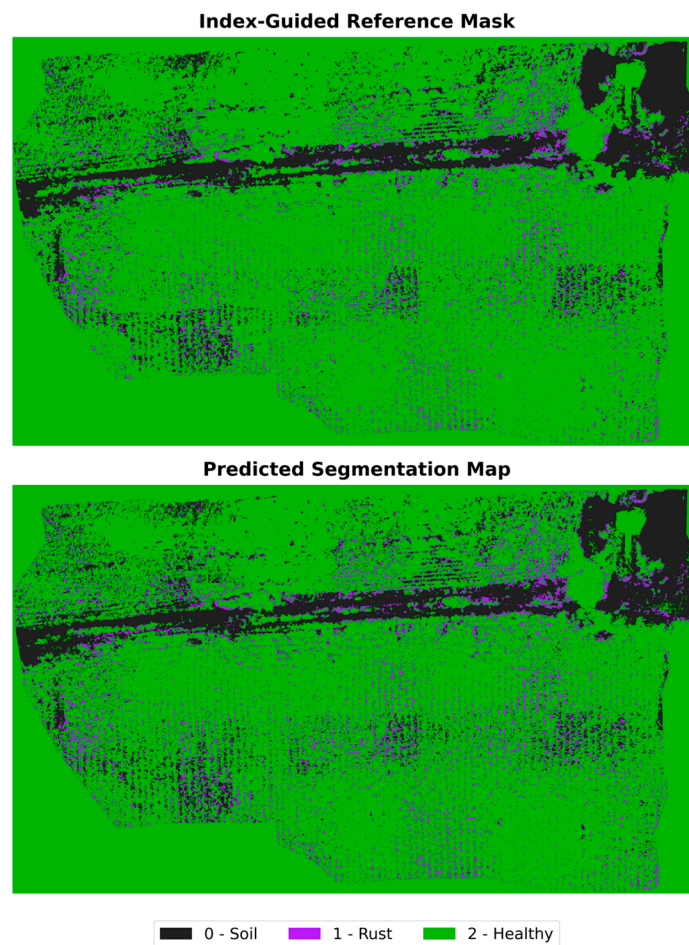
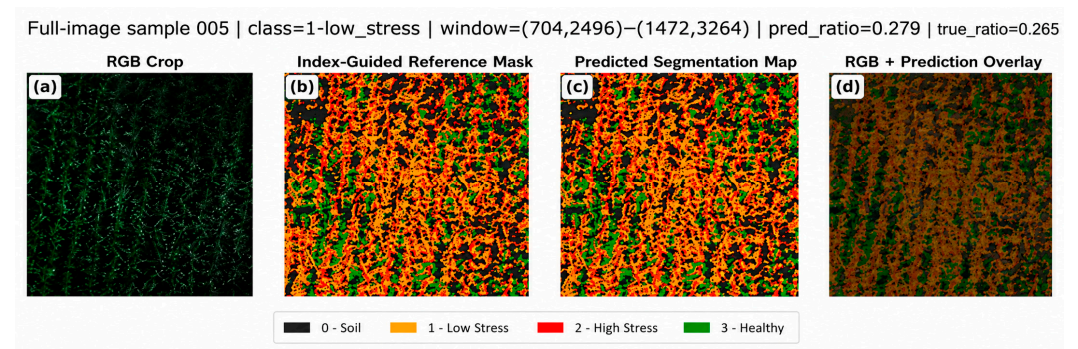


Figure 8. Field-scale segmentation mapping results for the Common Rust 2025 dataset.

4.9. Full-Image Sample-Based Qualitative Evaluation

Figure 9 Representative full-image sample windows extracted from the UAV ortho-mosaics. Each row includes the RGB crop, index-guided reference mask, predicted segmentation map, and RGB-prediction overlay. These examples demonstrate the local spatial consistency of the proposed framework and show how the predicted maps correspond to visible canopy patterns and field heterogeneity.



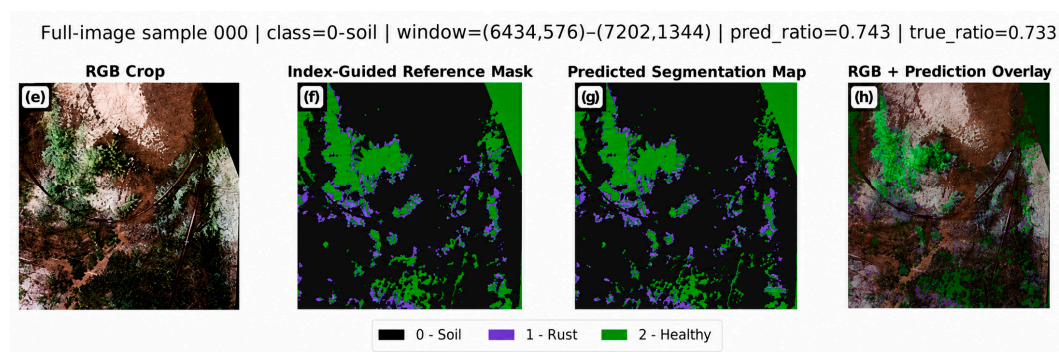


Figure 9. Representative full-image sample windows for Water Stress 2025 and Common Rust 2025. (a–d) Water Stress 2025 sample: RGB Crop, Index-Guided Reference Mask, Predicted Segmentation Map, RGB + Prediction Overlay, (e–h) Common Rust 2025 sample: RGB Crop, Index-Guided Reference Mask, Predicted Segmentation Map, RGB + Prediction Overlay.

4.10. Segmentation Confusion Matrix Analysis

As shown in Figure 10, The water-stress segmentation confusion matrix indicates that soil/background, low-stress, and high-stress regions were generally well separated. The remaining errors mainly occurred between healthy and stress-related vegetation classes, which is expected under real field conditions because mild water stress and healthy canopy regions may show partially overlapping spectral responses. The common-rust confusion matrix shows strong class-wise discrimination among soil/background, rust, and healthy vegetation. The main residual confusion occurred between rust and healthy vegetation, which can be attributed to early-stage symptoms, shadowed leaf regions, and heterogeneous canopy reflectance.

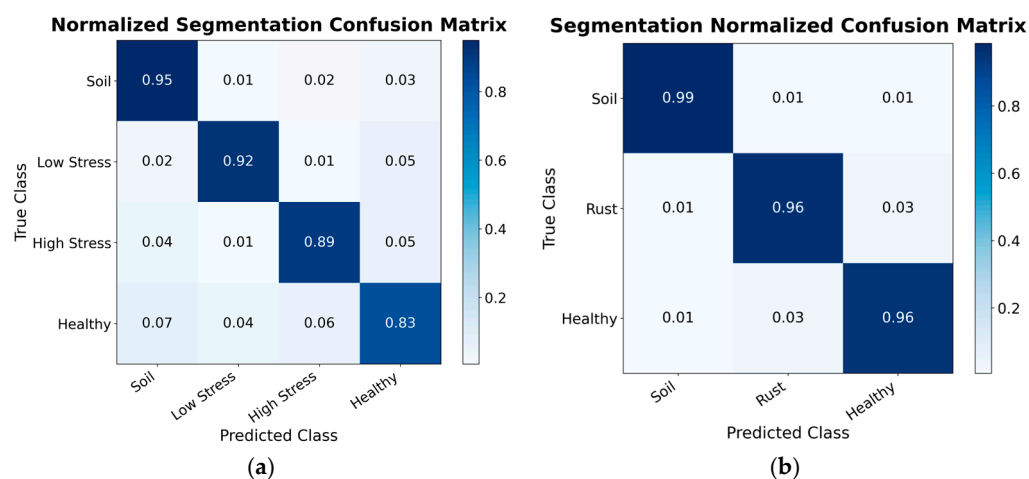


Figure 10. Normalized segmentation confusion matrices for (a) Water Stress 2025 and (b) Common Rust 2025.

4.11. Training Convergence and Stability Analysis

As shown in Figures 11 and 12, The convergence curves support the final segmentation results reported in Tables 3 and 4. For Water Stress 2025, the model reached its highest validation Mean IoU of 0.7723 at epoch 38, whereas for Common Rust 2025, the highest validation Mean IoU of 0.9164 was obtained at epoch 87. These results indicate stable learning behavior and show that the rust segmentation task benefited from longer optimization.

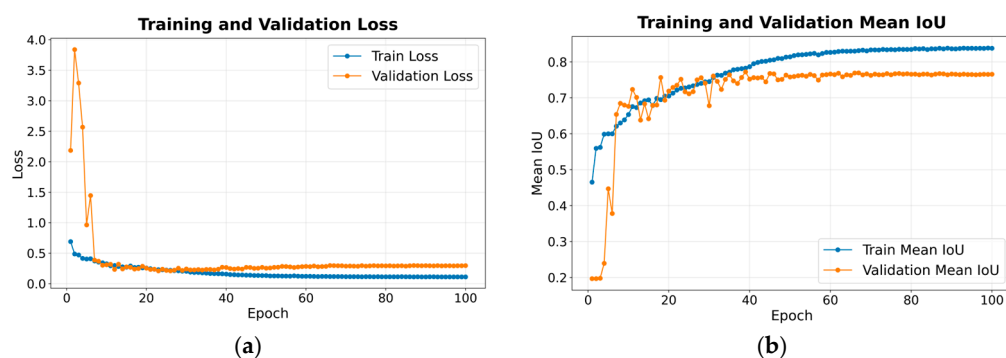


Figure 11. Training convergence curves for Water Stress 2025. (a) Training and validation loss (b) Training and validation Mean IoU.

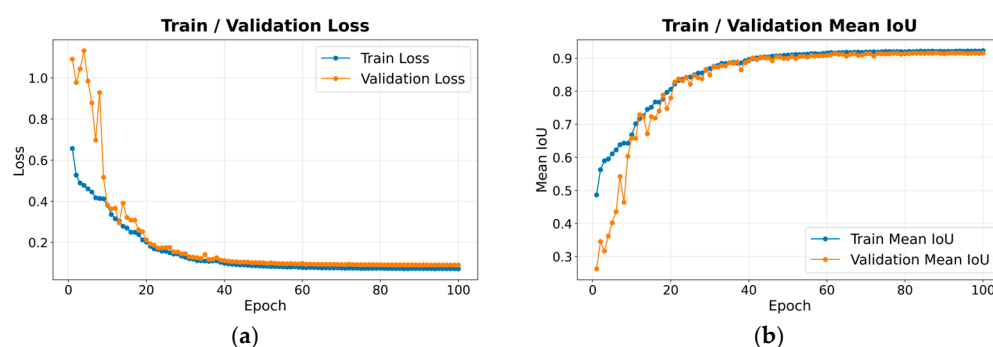


Figure 12. Training convergence curves for Common Rust 2025. (a) Training and validation loss (b) Training and validation Mean IoU.

4.12. Practical Implications of Segmentation-Based UAV Crop Monitoring

The field-scale segmentation maps show the practical value of the UAV-based segmentation part of the study. The segmentation workflow converts multispectral orthomosaics into spatial stress and disease maps. These maps help identify within-field variability, stressed areas, and infected regions. Therefore, the practical implications mainly relate to segmentation-based field mapping, not to the full classification branch.

For Water Stress 2025, the predicted map separates soil/background, low water stress, high water stress, and healthy maize. Low-stress areas may require continued monitoring. High-stress areas may guide irrigation assessment, soil moisture inspection, or targeted field observation. However, direct agronomic actions should also use field measurements and expert assessment.

For Common Rust 2025, the predicted map separates soil/background, rust-affected vegetation, and healthy maize. This can help identify candidate disease-prone areas, prioritize field scouting, and guide localized crop-protection assessment. However, disease-management decisions should be supported by field validation because the labels are adaptive index-guided masks, not direct physiological ground truth.

The full-image windows and RGB-prediction overlays improve interpretability. They show the segmentation outputs on the original UAV images. This allows visual comparison between predicted stress or disease regions, canopy structure, field texture, and exposed soil. These overlays support qualitative inspection, but they should be used together with quantitative evaluation and field verification.

The practical role of the classification branch is more limited. The classifier reached high accuracy, but the gain over DenseNet201 and the RGB + multispectral configuration was small. Also, Wavelet and GLCM mainly improved offline convergence, not final accuracy. Therefore, the full four-branch classifier should not be considered the default practical model for classification-only deployment. Simpler options such as DenseNet201 or

RGB + multispectral may be more suitable, depending on computing resources, preprocessing time, and deployment limits.

Overall, the practical contribution of the study mainly comes from UAV dataset construction, adaptive index-guided segmentation, and field-scale mapping. The classification part complements this workflow by analyzing modality contribution and convergence, but it does not prove that the complex classifier is practically superior.

5. Discussion

This section discusses the findings beyond performance values. It focuses on computational cost, real-time use, generalization under different field conditions, and the agricultural value of early stress and disease mapping. Since the classification experiments showed only a limited gain over strong baselines, the discussion separates the segmentation-based UAV monitoring contribution from the more limited classification-analysis contribution. Therefore, the method is evaluated in terms of technical performance, accuracy–complexity trade-off, deployment limits, and future needs.

5.1. Computational Cost, Real-Time Applicability, and Classification Complexity

Although the evaluated workflow achieved high segmentation and classification performance, the two stages have different computational meanings. The segmentation stage is directly useful for UAV-based field mapping because it produces pixel-level stress and disease maps. These maps provide spatial information that image-level classification cannot provide. In contrast, the practical value of the full classification branch is more limited. Its accuracy gain over strong general-purpose baselines was small, while its architecture and preprocessing steps were more complex.

This is important for real-time UAV applications, where processing power, memory, energy, and preprocessing time are limited. In this study, training and inference were performed offline on a high-performance workstation. This setting is suitable for post-field analysis, stress mapping, and decision support. However, it does not show that the full four-branch classifier is suitable for low-latency deployment. Future studies should test quantization, pruning, knowledge distillation, lighter backbones, and edge-device deployment. These methods may reduce inference and preprocessing costs. Therefore, the current workflow is suitable for offline agricultural analysis, but the classification branch needs further simplification for real-time UAV use.

5.2. Interpretation of the Contribution Relative to General-Purpose and Simpler Models

The results of this study should be interpreted by separating two parts: segmentation-based field mapping and image-level classification. The practical evidence is stronger for UAV-based segmentation and field-scale stress/disease mapping than for the full classification branch. Although the classifier reached high accuracy, its improvement over strong baselines and the RGB + multispectral configuration was small.

The evaluated workflow does not claim to be superior to all deep learning models. Its contribution should be considered in the context of UAV-based maize stress and disease analysis. DenseNet201, Xception, HRNet, and UPerNet were strong competitors. In classification, simpler models such as DenseNet201 or Xception may be more suitable when the goal is only image-level classification with lower complexity. Also, the RGB + multispectral configuration may be preferred because it reached peak accuracy without extra Wavelet and GLCM preprocessing.

The main contribution of the study is the UAV-based dataset and segmentation-oriented monitoring workflow. This workflow includes adaptive index-guided mask generation, scar-U-Net++ DB segmentation, field-scale stress/disease mapping, ablation analysis, and repeated-run stability analysis. The classification stage is included as a comple-

mentary analysis of modality contribution and convergence behavior. In segmentation, the model improved Mean IoU, especially for the Common Rust dataset, and produced field-scale maps for spatial interpretation. In classification, the full classifier achieved the highest accuracy, but the gain over the strongest baseline and the RGB + multispectral configuration was limited.

The ablation results showed that RGB and multispectral branches were enough to reach peak accuracy. Wavelet and GLCM mainly improved offline convergence, not final accuracy. Therefore, the classifier should not be seen as a model that provides a large accuracy gain. Its main value is to analyze how color, spectral, texture, and frequency-domain features affect classification under the same UAV-based protocol. The stability tests also showed that the proposed classifier was not the best in all input settings.

Overall, this study provides stronger evidence for segmentation-based UAV monitoring than for a practically superior complex classifier. The full classification branch is kept because it helps analyze multimodal branch behavior and convergence. However, simpler classification-only models may be preferable when deployment simplicity, pre-processing time, or parameter efficiency is more important.

5.3. Generalizability and Environmental Limitations

Although the proposed model achieved high performance, its generalizability should be further tested. The datasets were collected from two maize production areas with specific climate, soil, and cultivation conditions. Therefore, the learned spectral and spatial patterns may partly reflect local field characteristics. In real agricultural fields, soil color, plant phenotype, growth stage, irrigation, humidity, temperature, and light conditions can vary widely. Plants may also be affected by multiple biotic and abiotic stresses at the same time. Since the current dataset does not cover all this diversity, future studies should use multi-regional and multi-seasonal datasets. Cross-regional validation will help improve the model's transferability and operational reliability.

5.4. Agricultural and Economic Importance of Segmentation-Based Early Detection

Early detection of water stress and common rust is agriculturally important because delayed intervention may increase yield loss, irrigation inefficiency, and disease spread. In this study, the main decision-support value is associated with UAV-based segmentation maps, which provide spatial information about soil/background, healthy maize, water-stressed regions, and rust-affected vegetation. These maps may help guide field scouting, irrigation assessment, and localized crop-protection planning when combined with field validation and expert agronomic judgment.

The classification component complements this workflow by assigning patch-level labels and analyzing modality behavior. However, because its gain over simpler classification-only alternatives was limited, it should not be considered the primary source of agricultural or economic value in this study. Therefore, the practical agricultural relevance of the study is mainly linked to segmentation-based field mapping and early spatial warning rather than to demonstrating a substantially superior complex classifier.

5.5. Limitations and Future Work

Label and ground-truth limitation: The segmentation masks were generated using adaptive vegetation-index rules, not direct physiological measurements or fully expert-drawn biological masks. Indices such as SAVI, NDVI, NDRE, and SR are useful for plant stress analysis, but they can change with illumination, soil background, growth stage, location, and season. Although Otsu thresholding, K-Means clustering, morphological filtering, and CRF refinement were used to improve mask consistency, some uncertainty may remain. Therefore, the masks should be interpreted as adaptive index-guided labels

rather than direct physiological ground truth. Future studies should include expert-validated plots, field measurements, physiological observations, and independent field scoring.

Mixed-patch classification limitation: Another limitation is the mixed-patch problem. Each 128×128 patch may include soil, healthy vegetation, water-stressed plants, and rust-infected regions at the same time. Assigning only one class label to such a patch simplifies its real spatial structure. In this study, the majority-class rule was used to convert pixel-level masks into patch-level labels. However, secondary classes in mixed patches may be partly ignored. Future studies should consider multi-label classification, soft labels, fractional labels, or class-proportion regression.

Classification complexity and deployment limitation: A major limitation is the practical use of the full classification branch. The full XGHost-SECBAM-r-QB classifier combines RGB, multispectral, Wavelet, and GLCM branches and reached faster threshold-based convergence during offline training. However, its final accuracy gain over strong baselines such as DenseNet201 and over the RGB + multispectral configuration was limited. Also, Wavelet and GLCM did not improve peak accuracy; they mainly affected offline convergence. Therefore, the full four-branch classifier is not presented as a deployment-ready or universally preferable classification-only model. For operational classification, simpler models such as DenseNet201, Xception, or the RGB + multispectral configuration may be more suitable. Future studies should test lighter fusion models, compression, pruning, quantization, knowledge distillation, and edge-device deployment.

Benchmarking and external validation limitation: Disease-specific methods from previous studies were not directly reimplemented because they used different crops, sensors, inputs, image resolutions, class definitions, and evaluation protocols. In many cases, code, trained weights, preprocessing steps, or dataset splits were also unavailable. Therefore, the benchmark in this study is a controlled internal comparison on the same UAV datasets, not a direct reproduction of all previous methods. Future studies should test the workflow on independent UAV datasets and compare it with reimplemented state-of-the-art methods under shared protocols.

Environmental and seasonal generalization limitation: The datasets were collected from two maize fields with specific climate, soil, irrigation, and cultivation conditions. Therefore, the learned spectral and spatial patterns may partly depend on local field conditions. In real fields, soil color, plant phenotype, growth stage, irrigation regime, humidity, temperature, illumination, and disease–stress interactions may vary widely. Future studies should use multi-region, multi-season, and multi-year UAV datasets to improve transferability and operational reliability.

Overall, these limitations show that the study should be interpreted as a real-field UAV dataset, segmentation, and modality-analysis contribution, not as final evidence of a universally deployable classification system. The most important future directions are expert-validated ground truth, independent external validation, and lighter deployment-oriented classification designs.

6. Conclusions

This study presented a UAV-based workflow for detecting maize water stress and common rust under real field conditions. The study introduced two real-field UAV datasets, Water Stress 2025 and Common Rust 2025. It also included adaptive index-guided mask generation, scar-U-Net++ DB segmentation, field-scale stress/disease mapping, and complementary multimodal classification analysis. The datasets included five classes: soil, healthy maize, low water stress, high water stress, and common rust.

The scar-U-Net++ DB segmentation model achieved Mean IoU values of 0.7723 for Water Stress 2025, 0.9164 for Common Rust 2025, and 0.9531 on the benchmark dataset. Compared with the strongest benchmark model, it improved Mean IoU by 0.1177 on Wa-

ter Stress 2025 and 0.2644 on Common Rust 2025. These results show that the main quantitative contribution of the study is the segmentation-based UAV mapping approach.

The full classifier reached 99.54% accuracy in the five-class task. However, this result should be interpreted carefully. The improvement over strong baselines such as DenseNet201 and over the RGB + multispectral configuration was limited. The ablation results showed that RGB and multispectral inputs were enough to reach peak accuracy. Wavelet and GLCM branches mainly improved offline convergence, not final accuracy. Therefore, the full four-branch classifier should not be considered a clearly superior or deployment-ready classification model. Its main role is to analyze modality contribution and convergence behavior.

The repeated-run experiments supported the reproducibility of the segmentation-based UAV monitoring workflow. For classification, stability depended on the input modality. The proposed classifier was strongest with RGB-only input, while DenseNet-based models were more stable with multispectral-only and RGB + multispectral inputs. Therefore, the classification results do not show universal superiority. Model selection should depend on input type, convergence behavior, and deployment constraints.

The practical value of the study mainly comes from UAV dataset construction, adaptive index-guided segmentation, and field-scale stress/disease mapping. The generated maps can support field scouting, irrigation assessment, and localized crop-protection planning. However, these outputs should be checked with field observations and physiological measurements because the masks are adaptive index-guided labels, not direct physiological ground truth.

Future studies should include expert-validated ground truth, direct physiological measurements, independent UAV datasets, and multi-region and multi-season validation. They should also test lighter classification designs, such as simplified RGB + multispectral models, lightweight fusion modules, pruning, quantization, knowledge distillation, and edge-device deployment.

Supplementary Materials: The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/app16146860/s1>, Table S1: Complete version of the literature comparison table. Table S2: Complete version of the dataset comparison table. Figure S1: Representative six-channel patch and orthomosaic examples from the Water Stress 2025 and Common Rust 2025 datasets. Figure S2: Additional dataset examples and comparison with public benchmark datasets. Figure S3: Additional 3D views of the proposed dual-branch UNet++ segmentation architecture. Figure S4: Detailed Xception-based classification branch architecture. Figure S5: Class distribution of the RGB and six-channel multispectral training/test datasets. Table S3a: Friedman ranking results for the Water Stress 2025 segmentation ablation study. Table S3b: Friedman ranking results for the Common Rust 2025 segmentation ablation study. Table S3c: Nemenyi post-hoc comparison between the selected final segmentation configuration and the best-ranked configuration. Table S4a–S4b: Stability analysis experiments for Water Stress 2025 segmentation. Table S5a–S5b: Stability analysis experiments for Common Rust 2025 segmentation. Table S6a–S6b: Summary SR, MEC, and ACD performance for Water Stress 2025 and Common Rust 2025 segmentation. Table S7: Stability analysis experiment with the benchmark segmentation dataset. Table S8: SR, MEC, and ACD performance for the benchmark segmentation dataset. Table S9a–S9c: Classification stability experiments under RGB-only, multispectral-only, and fused RGB + multispectral input settings. Table S10: SR, MEC, and ACD performance for classification under RGB, multispectral-only, and fused RGB + multispectral input settings. Figure S6: Representative test-sample visualization for the classification task. Supplementary Sections S15–S17: Detailed formulations of Otsu thresholding, K-Means clustering, and fully connected Conditional Random Field. Supplementary Section S18: Mathematical formulation of the stability metrics.

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

Funding: This research received no external funding.

Data Availability Statement: A subset of the UAV-derived datasets introduced in this study is publicly available at Zenodo: <https://zenodo.org/records/19108517> (accessed on 4 June 2026) This repository provides representative data related to the Water Stress 2025 and Common Rust 2025 datasets. The complete datasets will be released through Zenodo after the final acceptance of the manuscript. The source code of the proposed scar-U-Net++ DB segmentation model and XGhost-SECBAM-r-QB classification model will also be provided after final editorial approval.

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

References

1. Erenstein, O.; Jaleta, M.; Sonder, K.; Mottaleb, K.; Prasanna, B.M. Global Maize Production, Consumption and Trade: Trends and R&D Implications. *Food Secur.* **2022**, *14*, 1295–1319.
2. Savary, S.; Willocquet, L.; Pethybridge, S.J.; Esker, P.; McRoberts, N.; Nelson, A. The Global Burden of Pathogens and Pests on Major Food Crops. *Nat. Ecol. Evol.* **2019**, *3*, 430–439.
3. Zhu, M.; Zhong, T.; Xu, L.; Guo, C.; Zhang, X.; Liu, Y.; Zhang, Y.; Li, Y.; Xie, Z.; Liu, T. The Zmcpk39–Zmdi19–Zmpr10 Immune Module Regulates Quantitative Resistance to Multiple Foliar Diseases in Maize. *Nat. Genet.* **2024**, *56*, 2815–2826.
4. Zheng, H.; Chen, J.; Mu, C.; Makumbi, D.; Xu, Y.; Mahuku, G. Combined Linkage and Association Mapping Reveal Qtl for Host Plant Resistance to Common Rust (*Puccinia Sorghi*) in Tropical Maize. *BMC Plant Biol.* **2018**, *18*, 310.
5. Leng, G.; Hall, J. Crop Yield Sensitivity of Global Major Agricultural Countries to Droughts and the Projected Changes in the Future. *Sci. Total Environ.* **2019**, *654*, 811–821.
6. Cairns, J.E.; Prasanna, B.M. Developing and Deploying Climate-Resilient Maize Varieties in the Developing World. *Curr. Opin. Plant Biol.* **2018**, *45*, 226–230.
7. Trivedi, P.; Leach, J.E.; Tringe, S.G.; Sa, T.; Singh, B.K. Plant–Microbiome Interactions: From Community Assembly to Plant Health. *Nat. Rev. Microbiol.* **2020**, *18*, 607–621.
8. Barbedo; Jayme, G.A. Factors Influencing the Use of Deep Learning for Plant Disease Recognition. *Biosyst. Eng.* **2018**, *172*, 84–91.
9. Upadhyay, A.; Chandel, N.S.; Singh, K.P.; Chakraborty, S.K.; Nandede, B.M.; Kumar, M.; Subeesh, A.; Upendar, K.; Salem, A.; Elbeltagi, A. Deep Learning and Computer Vision in Plant Disease Detection: A Comprehensive Review of Techniques, Models, and Trends in Precision Agriculture. *Artif. Intell. Rev.* **2025**, *58*, 92.
10. Chen, S.; Zhang, K.; Wu, S.; Tang, Z.; Zhao, Y.; Sun, Y.; Shi, Z. A Weakly Supervised Approach for Disease Segmentation of Maize Northern Leaf Blight from Uav Images. *Drones* **2023**, *7*, 173.
11. Pacal, I.; Işık, G. Utilizing Convolutional Neural Networks and Vision Transformers for Precise Corn Leaf Disease Identification. *Neural Comput. Appl.* **2025**, *37*, 2479–2496.
12. Keshavarzi, M.; Mesarich, C.; Bailey, D.; Johnson, M.; Sen Gupta, G. A Review of Semantic Segmentation Methods and Their Application in Apple Disease Detection. *Comput. Electron. Agric.* **2025**, *237*, 110531.
13. Huang, M.; Xu, G.; Li, J.; Huang, J. A Method for Segmenting Disease Lesions of Maize Leaves in Real Time Using Attention Yolact++. *Agriculture* **2021**, *11*, 1216.
14. Isinkaye; Olubusola, F.; Olusanya, M.O.; Akinyelu, A.A. A Multi-Class Hybrid Variational Autoencoder and Vision Transformer Model for Enhanced Plant Disease Identification. *Intell. Syst. With Appl.* **2025**, *26*, 200490.
15. Zhu, T.; Yan, F.; Lv, X.; Zhao, H.; Wang, Z.; Dong, K.; Fu, Z.; Jia, R.; Lv, C. A Deep Learning Model for Accurate Maize Disease Detection Based on State-Space Attention and Feature Fusion. *Plants* **2024**, *13*, 3151.
16. Zhou, H.; Su, Y.; Chen, J.; Li, J.; Ma, L.; Liu, X.; Lu, S.; Wu, Q. Maize Leaf Disease Recognition Based on Improved Convolutional Neural Network Shufflenetv2. *Plants* **2024**, *13*, 1621.

17. Yang, T.; Wang, Y.; Lian, J. Plant Diseased Lesion Image Segmentation and Recognition Based on Improved Multi-Scale Attention Net. *Appl. Sci.* **2024**, *14*, 1716.
18. Zhao, Y.; Liu, D.; Wang, Z.; Yao, J.; Yu, J.; Wang, Z.; Jia, Z. Automated Phenotyping of Maize from 3d Point Clouds Using an Optimized Deep Learning Approach. *Agriculture* **2025**, *15*, 2430.
19. Zhu, Z.; Chen, Y.; Lu, C.; Yang, M.; Xia, Y.; Huang, D.; Lv, J. Research on Crop Classification Using U-Net Integrated with Multimodal Remote Sensing Temporal Features. *Sensors* **2025**, *25*, 5005.
20. Albahli, S. Agrifusionnet: A Lightweight Deep Learning Model for Multisource Plant Disease Diagnosis. *Agriculture* **2025**, *15*, 1523.
21. Zhao, C.; Li, C.; Wang, X.; Wu, X.; Du, Y.; Chai, H.; Cai, T.; Xiang, H.; Jiao, Y. Plant Disease Segmentation Networks for Fast Automatic Severity Estimation under Natural Field Scenarios. *Agriculture* **2025**, *15*, 583.
22. Osouli, S.; Bolourian Haghighi, B.; Sadrossadat, E. Mob-INC: An effective scheme for maize disease recognition based on deep networks. *Expert Syst. Appl.* **2025**, 129006. <https://doi.org/10.1016/j.eswa.2025.129006>.
23. Rani, R.; Sahoo, J.; Bellamkonda, S.; Kumar, S. Attention-Enhanced Corn Disease Diagnosis Using Few-Shot Learning and Vgg16. *MethodsX* **2025**, *14*, 103172.
24. Calderara-Cea, F.; Torres-Torriti, M.; Auat Cheein, F.; Delpiano, J. A Two-Stage Deep Learning Strategy for Weed Identification in Grassfields. *Comput. Electron. Agric.* **2024**, *225*, 109300.
25. Liu, Y.; Su, J.; Zheng, Z.; Liu, D.; Song, Y.; Fang, Y.; Yang, P.; Su, B. Gldcnet: A Novel Convolutional Neural Network for Grapevine Leafroll Disease Recognition Using Uav-Based Imagery. *Comput. Electron. Agric.* **2024**, *218*, 108668.
26. Hu, Y.; Tang, J.; Yang, J. Introducing Artificial Intelligence Technology to Plant Disease Management for Sustainable Agriculture. *Crop Prot.* **2024**, *184*, 106764.
27. Polly, R.; Devi, E.A. Semantic Segmentation for Plant Leaf Disease Classification and Damage Detection: A Deep Learning Approach. *Smart Agric. Technol.* **2024**, *9*, 100526.
28. Rajpal, A.; Mishra, R.; Rajpal, S.; Kavita; Bhatia, V.; Kumar, N. Explaining Deep Learning-Based Leaf Disease Identification. *Soft Comput.* **2024**, *28*, 12299–12322.
29. Lv, Z.; Xu, B.; Zhong, L.; Chen, G.; Huang, Z.; Sun, R.; Huang, W.; Zhao, F.; Meng, R. Improved Monitoring of Southern Corn Rust Using Uav-Based Multi-View Imagery and an Attention-Based Deep Learning Method. *Comput. Electron. Agric.* **2024**, *224*, 109232.
30. Khan, I.; Sohail, S.S.; Madsen, D.Ø.; Khare, B.K. Deep Transfer Learning for Fine-Grained Maize Leaf Disease Classification. *J. Agric. Food Res.* **2024**, *16*, 101148.
31. Aszkowski, P.; Kraft, M.; Drapikowski, P.; Pieczyński, D. Estimation of Corn Crop Damage Caused by Wildlife in Uav Images. *Precis. Agric.* **2024**, *25*, 2505–2530.
32. Aslan, I.; Polat, N. Deep Learning-Based Classification of Mature and Immature Lavender Plants Using Uav Orthophotos and a Hybrid Cnn Approach. *Earth Sci. Inform.* **2024**, *17*, 1713–1727.
33. Theerthagiri, P.; Ruby, A.U.; Chandran, J.G.C.; Sardar, T.H.; Ahamed Shafeeq, B.M. Deep Squeezenet Learning Model for Diagnosis and Prediction of Maize Leaf Diseases. *J. Big Data* **2024**, *11*, 112.
34. Zhang, F.; Bao, R.; Yan, B.; Wang, M.; Zhang, Y.; Fu, S. Lsannet: A Lightweight Convolutional Neural Network for Maize Leaf Disease Identification. *Biosyst. Eng.* **2024**, *248*, 97–107.
35. Sharma, V.; Tripathi, A.K.; Daga, P.; Nidhi, M.; Mittal, H. Clgannet: A Novel Method for Maize Leaf Disease Identification Using Clgan and Deep Cnn. *Signal Process. Image Commun.* **2024**, *120*, 117074.
36. Wang, H.; Pan, X.; Zhu, Y.; Li, S.; Zhu, R. Maize Leaf Disease Recognition Based on Tc-Mrsn Model in Sustainable Agriculture. *Comput. Electron. Agric.* **2024**, *221*, 108915.
37. Wang, P.; Xiong, Y.; Zhang, H. Maize Leaf Disease Recognition Based on Improved Msrrc and Oscrnet. *Crop Prot.* **2024**, *183*, 106757.
38. Pacal, I. Enhancing Crop Productivity and Sustainability through Disease Identification in Maize Leaves: Exploiting a Large Dataset with an Advanced Vision Transformer Model. *Expert Syst. With Appl.* **2024**, *238*, 122099.
39. Waheed, A.; Goyal, M.; Gupta, D.; Khanna, A.; Hassanien, A.E.; Mohan Pandey, H. An Optimized Dense Convolutional Neural Network Model for Disease Recognition and Classification in Corn Leaf. *Comput. Electron. Agric.* **2020**, *175*, 105456.
40. Bezabh; Agegnehu, Y.; Salau, A.O.; Abuhayi, B.M.; Mussa, A.A.; Ayalew, A.M. Cpd-Cnn: Classification of Pepper Disease Using a Concatenation of Convolutional Neural Network Models. *Sci. Rep.* **2023**, *13*, 15581.
41. Dash, A.; Sethy, P.K.; Behera, S.K. Maize Disease Identification Based on Optimized Support Vector Machine Using Deep Feature of Densenet201. *J. Agric. Food Res.* **2023**, *14*, 100824.

42. Zhuang, S.; Wang, P.; Jiang, B.; Li, M. Learned Features of Leaf Phenotype to Monitor Maize Water Status in the Fields. *Comput. Electron. Agric.* **2020**, *172*, 105347.
43. Hanh; Thi, B.; Van Manh, H.; Nguyen, N.-V. Enhancing the Performance of Transferred Efficientnet Models in Leaf Image-Based Plant Disease Classification. *J. Plant Dis. Prot.* **2022**, *129*, 623–634.
44. Priyadarshini, A.; Arivazhagan, S.; Arun, M.; Mirmalini, A. Maize Leaf Disease Classification Using Deep Convolutional Neural Networks. *Neural Comput. Appl.* **2019**, *31*, 8887–8895.
45. Jiang, B.; Wang, P.; Zhuang, S.; Li, M.; Li, Z.; Gong, Z. Detection of Maize Drought Based on Texture and Morphological Features. *Comput. Electron. Agric.* **2018**, *151*, 50–60.
46. Zhuang, S.; Wang, P.; Jiang, B.; Li, M.; Gong, Z. Early Detection of Water Stress in Maize Based on Digital Images. *Comput. Electron. Agric.* **2017**, *140*, 461–468.
47. Wang, X.; Fan, L.; Li, H.; Bi, X.; Jiang, W.; Ma, X. Skip-Attseqnet: Leveraging Skip Connection and Attention-Driven Seq2seq Model to Enhance Eye Movement Event Detection in Parkinson's Disease. *Biomed. Signal Process. Control* **2025**, *99*, 106862.
48. Sun, L.; Li, W.; Xu, Y. Ghost-Unet: Lightweight Model for Underwater Image Enhancement. *Eng. Appl. Artif. Intell.* **2024**, *133*, 108585.
49. Wang, J.; Lv, P.; Wang, H.; Shi, C. Sar-U-Net: Squeeze-and-Excitation Block and Atrous Spatial Pyramid Pooling Based Residual U-Net for Automatic Liver Segmentation in Computed Tomography. *Comput. Methods Programs Biomed.* **2021**, *208*, 106268.
50. Zhang, J.; Kong, Y.; Chen, Z.; Han, T.; Han, Q.; Dong, M.; Chu, F. Cbam-Crlsgan: A Novel Fault Diagnosis Method for Planetary Transmission Systems under Small Samples Scenarios. *Measurement* **2024**, *234*, 114795.
51. Liu, H.; Cao, F. Improved Dual-Scale Residual Network for Image Super-Resolution. *Neural Netw.* **2020**, *132*, 84–95.
52. Zhu, R.; Yang, M.; Liu, W.; Song, R.; Yan, B.; Xiao, Z. Deepavo: Efficient Pose Refining with Feature Distilling for Deep Visual Odometry. *Neurocomputing* **2022**, *467*, 22–35.
53. Bao, W.; Huang, C.; Hu, G.; Su, B.; Yang, X. Detection of Fusarium Head Blight in Wheat Using Uav Remote Sensing Based on Parallel Channel Space Attention. *Comput. Electron. Agric.* **2024**, *217*, 108630.
54. Bai, Y.; Nie, C.; Yu, X.; Gou, M.; Liu, S.; Zhu, Y.; Jiang, T.; Jia, X.; Liu, Y.; Nan, F. Comprehensive Analysis of Hyperspectral Features for Monitoring Canopy Maize Leaf Spot Disease. *Comput. Electron. Agric.* **2024**, *225*, 109350.
55. Bhadra, S.; Sagan, V.; Sarkar, S.; Braud, M.; Mockler, T.C.; Eveland, A.L. Prosail-Net: A Transfer Learning-Based Dual Stream Neural Network to Estimate Leaf Chlorophyll and Leaf Angle of Crops from Uav Hyperspectral Images. *ISPRS J. Photogramm. Remote Sens.* **2024**, *210*, 1–24.
56. Cheng, Q.; Ding, F.; Xu, H.; Guo, S.; Li, Z.; Chen, Z. Quantifying Corn Lai Using Machine Learning and Uav Multispectral Imaging. *Precis. Agric.* **2024**, *25*, 1777–1799.
57. Ma, W.; Han, W.; Zhang, H.; Cui, X.; Zhai, X.; Zhang, L.; Shao, G.; Niu, Y.; Huang, S. Uav Multispectral Remote Sensing for the Estimation of Spad Values at Various Growth Stages of Maize under Different Irrigation Levels. *Comput. Electron. Agric.* **2024**, *227*, 109566.
58. Meiyang, S.; Qizhou, D.; ShuaiPeng, F.; Xiaohong, Y.; Jinyu, Z.; Lei, M.; Baoguo, L.; Yuntao, M. Improved Estimation of Canopy Water Status in Maize Using Uav-Based Digital and Hyperspectral Images. *Comput. Electron. Agric.* **2022**, *197*, 106982.
59. Zhang, L.; Han, W.; Niu, Y.; Chavez, J.L.; Shao, G.; Zhang, H. Evaluating the Sensitivity of Water Stressed Maize Chlorophyll and Structure Based on Uav Derived Vegetation Indices. *Comput. Electron. Agric.* **2021**, *185*, 106174.
60. Niu, Y.; Han, W.; Zhang, H.; Zhang, L.; Chen, H. Estimating Fractional Vegetation Cover of Maize under Water Stress from Uav Multispectral Imagery Using Machine Learning Algorithms. *Comput. Electron. Agric.* **2021**, *189*, 106414.
61. Deng, J.; Zhou, H.; Lv, X.; Yang, L.; Shang, J.; Sun, Q.; Zheng, X.; Zhou, C.; Zhao, B.; Wu, J. Applying Convolutional Neural Networks for Detecting Wheat Stripe Rust Transmission Centers under Complex Field Conditions Using Rgb-Based High Spatial Resolution Images from Uavs. *Comput. Electron. Agric.* **2022**, *200*, 107211.
62. Tiwari, L.; Raja, R.; Awasthi, V.; Miri, R.; Sinha, G.R.; Alkinani, M.H.; Polat, K. Detection of Lung Nodule and Cancer Using Novel Mask-3 Fcm and Twedlnn Algorithms. *Measurement* **2021**, *172*, 108882.
63. Biswas; Kr, T.; Giri, K.; Roy, S. Eckm: An Improved K-Means Clustering Based on Computational Geometry. *Expert Syst. With Appl.* **2023**, *212*, 118862.
64. Huang, J.; Zhang, G.; Ren, L.J.; Wang, N. A New Image Segmentation Method Based on the Yolo5 and Fully Connected Crf. *Int. J. Comput. Intell. Syst.* **2023**, *16*, 180.
65. Sahin; Mertkan, H.; Miftahushudur, T.; Grieve, B.; Yin, H. Segmentation of Weeds and Crops Using Multispectral Imaging and Crf-Enhanced U-Net. *Comput. Electron. Agric.* **2023**, *211*, 107956.

66. Chen, C.; Zhang, C.; Wang, J.; Li, D.; Li, Y.; Hong, J. Semantic Segmentation of Mechanical Assembly Using Selective Kernel Convolution Unet with Fully Connected Conditional Random Field. *Measurement* **2023**, *209*, 112499.
67. Zou, Y.; Chen, S.; Che, C.; Zhang, J.; Zhang, Q. Breast Cancer Histopathology Image Classification Based on Dual-Stream High-Order Network. *Biomed. Signal Process. Control* **2022**, *78*, 104007.
68. Zhou, C.; Chen, S.; Ding, C.; Tao, D. Learning Contextual and Attentive Information for Brain Tumor Segmentation. In Paper Presented at the International MICCAI Brainlesion Workshop, 2018.
69. Roy; Guha, A.; Siddiqui, S.; Pölsterl, S.; Navab, N.; Wachinger, C. ‘Squeeze & Excite’guided Few-Shot Segmentation of Volumetric Images. *Med. Image Anal.* **2020**, *59*, 101587.
70. Zhao, Y.; Liu, X.; Li, H.; Zhou, Y.; Zhang, J. Adaptive Skip-Review Attention-Based Networks for Intracranial Aneurysm Segmentation from Three-Dimensional Rotational Angiography. *Expert Syst. With Appl.* **2025**, *295*, 128775.
71. Ou, Y.; Chen, Y.; Zhang, Y.; Yang, S.; Zhang, X.; Zhou, Y.; Chen, S.; Peng, L. Enhanced Medical Image Segmentation Via Deep Dynamic Self-Adjusting U-Net with Multi-Scale Attention and Semantic Mitigation. *Vis. Comput.* **2025**, *41*, 8385–8401.
72. Shaheed, K.; Mao, A.; Qureshi, I.; Kumar, M.; Hussain, S.; Ullah, I.; Zhang, X. Ds-Cnn: A Pre-Trained Xception Model Based on Depth-Wise Separable Convolutional Neural Network for Finger Vein Recognition. *Expert Syst. With Appl.* **2022**, *191*, 116288.

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.