



ARTICLE

Border Sensitive Knowledge Distillation for Rice Panicle Detection in UAV Images

Anitha Ramachandran and Sendhil Kumar K.S.*

School of Computer Science and Engineering, VIT (Vellore Institute of Technology) Vellore, Vellore, 632014, India

*Corresponding Author: Sendhil Kumar K.S. Email: sendhilkumar.ks@vit.ac.in

Received: 06 June 2024 Accepted: 11 August 2024 Published: 15 October 2024

ABSTRACT

Research on panicle detection is one of the most important aspects of paddy phenotypic analysis. A phenotyping method that uses unmanned aerial vehicles can be an excellent alternative to field-based methods. Nevertheless, it entails many other challenges, including different illuminations, panicle sizes, shape distortions, partial occlusions, and complex backgrounds. Object detection algorithms are directly affected by these factors. This work proposes a model for detecting panicles called Border Sensitive Knowledge Distillation (BSKD). It is designed to prioritize the preservation of knowledge in border areas through the use of feature distillation. Our feature-based knowledge distillation method allows us to compress the model without sacrificing its effectiveness. An imitation mask is used to distinguish panicle-related foreground features from irrelevant background features. A significant improvement in Unmanned Aerial Vehicle (UAV) images is achieved when students imitate the teacher's features. On the UAV rice imagery dataset, the proposed BSKD model shows superior performance with 76.3% mAP, 88.3% precision, 90.1% recall and 92.6% F1 score.

KEYWORDS

Rice panicle detection; UAV; border sensitive method; deep learning; knowledge distillation

1 Introduction

The majority of the world's population consumes rice as a staple food. Globally, it is grown in more than 100 countries, but Asia contributes 90% of the production. In addition to meeting caloric needs, rice constituents have potential health benefits. It is, however, important to consider any benefits in the context of the changing environment. Agricultural practices and environmental change affect rice quality and production in a significant way. As a staple in the global food system, rice must survive [1]. Precision agriculture and plant phenotyping tools have increased rice yield monitoring and prediction. Farmers can thus make better crop management decisions as a result of making educated decisions. Furthermore, it reduces resource use, which results in higher yields and profitability. During rice field phenotyping, structural traits, root traits, water-related traits, carbon-based traits, and photosynthesis-associated traits are measured [2]. Several factors influence grain yield, including plant density, the number of tillers per plant, the number of grains per panicle, the approximation of panicle heading date, and grain size. Rice plant shoot habits and panicles are directly related to yield potential. It is



especially critical to grasp the importance of the panicle since its development directly impacts grain density [3]. Panicle distribution is essential for plant breeders and agronomists to understand paddy production. Therefore, panicle detection is key to paddy phenotyping. Paddy panicles are difficult to quantify in the field. In the field, manual sampling is often used for yield estimation, although it is inherently imprecise and requires a lot of effort. As a result, alternative automated methods are developed for estimating paddy density and yield. The advent of low-cost imaging devices opens up the potential of substituting manual inspection in the rice field with image-based panicle analysis [4]. A variety of remote sensing platforms are available for agricultural data collection, including ground-based sensors, Unmanned Aerial Vehicles (UAVs), and satellite sensors. UAVs offer several advantages over other remote sensing platforms. Less expensive, enhanced flight capabilities, and more effective operation are some of the benefits of this technology. Drones provide a reliable method of acquiring data about rice crops [5]. UAVs equipped with RGB, thermal, and multispectral cameras provide information on rice crop growth above agricultural fields, enabling large-scale breeding experiments to be conducted. It has been demonstrated that the combination of UAV RGB data and deep learning is more effective than the conventional regression model for multispectral imagery. The problem remains that extracting valuable information from images is still difficult, despite the ease and speed at which images can be acquired.

Recently, researchers have demonstrated significant interest in using machine learning methods to detect crop diseases [6,7], water stress, and crop yield prediction. Wu et al. [8] presented an image analysis method for counting grains in a panicle. Linear regression is used to count grains and has an accuracy of 96% for Japonica rice and 97% for Indica rice. Yang et al. [9] measured High throughput Leaf Scoring (HLS) through image analysis based on the Hue, Saturation and Illumination colour space method. Li et al. [10] attempted to detect rice seedlings in each hill. The implementation and analysis of several colour models is performed to differentiate rice seedlings from the rest of the field. Handcrafted features, on the other hand, require manual extraction of complicated features and are suited to a specific scene, limiting model generalization. Machine learning algorithms are not well suited to complicated field scenes with densely distributed paddy panicles and significant occlusion. Deep learning models automatically learn features from input images due to deep hidden layers. However, there are still several research challenges with automatic UAV-based panicle detection: (1) Complex field background including weeds, shades, and water reflection. (2) Rice panicles vary greatly in size, shape and colour depending on growth stages, lighting conditions and UAV camera angle. (3) Occlusion between panicles, stems and leaves. These considerations motivate the development of an efficient panicle detection algorithm based on UAV imagery. The main advantage of the proposed algorithm is that the panicles along the edge of the image border can be easily identified, regardless of the size of the panicle. Consequently, the performance of the system is enhanced as a result.

The main contributions are:

1. To improve panicle detection performance, Border Sensitive Mechanism-based Knowledge Distillation (BSKD) is proposed.
2. To distinguish between the foreground and background features, a mask is used. The student learns these aspects as an added benefit.
3. The border sensitive method is applied to ResNet50 to derive border features more effectively.

The structure of this paper is as follows: [Section 1](#) introduces rice phenotypic research and discusses existing machine learning techniques. The [Section 2](#) discusses prior research on rice panicle detection. The [Section 3](#) describes the dataset and the proposed BSKD method. The proposed model results are depicted and explained in [Section 4](#). The paper concludes with [Section 5](#).

2 Related Works

Previous crop phenotype researchers used machine learning to analyze phenotypic traits automatically. Guo et al. [11] analyzed time-series RGB images to study rice panicle flowering dynamics. These images were captured at 2 m altitude with a Canon DSLR camera. Their approach is a hybrid of algorithms including scale-invariant features, a support vector machine, and a bag of visual words. Yang et al. [12] developed a UAV-based dataset for rice seedling classification and detection. The images were taken from 20 m altitude with the DJI Phantom4 Pro and the DJI Zenmuse. The image has a resolution of 5472×3648 and is annotated with the LabelImg tool. Tseng et al. [13] refined the preceding model by providing a hybrid method that combined transfer learning and machine learning. The author emphasizes recognizing small objects and using previously unknown data. In the experiment, an open-source rice seedling dataset based on UAV is applied and obtained 0.87 mIoU. Yu et al. [14] created a rice panicle phenotyping approach based on the Faster RCNN model to assess spikelet and kernel traits. In the study, Qiu et al. [15] examined rice qualification at various growth stages. These images are processed using the ENVI5.3 tool and the Pix4D mapper tool. The author proposed an optimal vegetative index (VI) approach and an object-oriented segmentation method to analyze rice growth qualifications.

Deep learning algorithms have recently been employed in agricultural phenotypic research to avoid the need for manual feature extraction. There are two types of deep learning-based object detection algorithms: one-stage approaches and two-stage approaches. Wang et al. [16] proposed a rice panicle detection and counting model for both offline and online. The input data is augmented and trained in the PanicleDetect model. This model employs the YOLO5X deep learning model. During the online stage, the input images are fed into the trained PanicleDetect model and the results are saved. This work was only evaluated at one planting density, which is a bit higher than the normal planting density utilized for rice production. Different one-stage and two-stage deep learning models for detecting faulty paddy seedlings are empirically examined by Anuar et al. [17]. The DJI Phantom 4 was used to capture images of paddy seedlings with a resolution of 4000×3000 from a height of 5 m. EfficientDet-D1, Faster RRCNN, SSD, ResNet50, EfficientNet, and MobileNetV2 are the models that were tested. These models' different combinations are examined, and it is found that EfficientDet-D1 with EfficientNet performs better than the other combinations. A general model for detecting panicles in many rice varieties was presented by Sun et al. [18]. These rice varieties are grown in various ecological zones and are members of the Indica family. The proposed approach is improved on YOLOv4 to produce better results. Experiments were conducted on three types of datasets namely, the middle, the leaf-above-spike, and the spike-above-leaf. It is observed that the full heading stage had a better detection result than the maturity stage. One-stage detector is characterized by their efficient architectures, while two-stage detectors continue to outperform them in terms of accuracy. According to Zhou et al. [19], rice panicle detection and counting could be accomplished by using a modified region-based fully convolutional network. The proposed model has a precision score of 0.86, which is higher than the other works evaluated. The study by Yang et al. [20] presented a novel method for estimating rice grain yield at the stage of ripening using high-resolution UAV images. The proposed Convolutional Neural Network (CNN) architecture includes two distinct branches for processing multispectral and RGB images. Compared to methods based on vegetative index, it precisely predicts paddy yield. It also indicates that paddy datasets nearing harvest contain more relevant spatial characteristics. RiceNet was proposed by Bai et al. [21] is comprised of a single feature extractor, plant size estimator, plant location detector, and density map estimator. To extract paddy plants from the background, an attention mechanism is used.

3 Materials and Methods

3.1 Dataset

In this research, the paddy rice imagery dataset created by Wang et al. [22] is used. The experimental location is situated within Hokkaido University in Japan. In August 2018, UAV images of paddy fields were acquired at 1.2 m altitude. The configuration leads to a ground sampling distance of 0.04 cm per pixel. The horizontal overlap ratio is 43% while the vertical overlap ratio is 70%. Kitaake and KoKusyokuto-2 are rice varieties. The observed cultivation densities ranged from 11.46 to 16.18 plants per square meter. A commercial UAV was used to gather aerial imagery of the rice field. The efficiency of the aforementioned technology surpasses that of a handheld data collection device. A larger UAV can accommodate and support heavier cameras and lenses, which facilitates the capture of images of enhanced quality. Nevertheless, rotor-induced downward wind exerts a significant force on rice stems, resulting in image quality degradation. Hence, considering the inherent trade-offs regarding flight altitude, image resolution, and user-friendliness, the Mavic Pro, by DJI Corporation, is used. It is a small, lightweight drone with an integrated 1/2.3-inch Complementary Metal Oxide Semiconductor (CMOS) sensor. From the acquired videos, frames are extracted without duplication, blurring, and overexposure. Each frame has 4096×2160 resolution. It encompasses the phenotype of rice panicles throughout the entire reproductive cycle of rice. This is during which rice witnesses the heading phase, flowering phase, and ripening phase. A few examples are illustrated in Fig. 1. The heading stage refers to the developmental phase during which the panicles become fully apparent. The flowering phase begins immediately after heading when tiny white spikes emerge from the panicles. Flowering lasts seven days. After flowering, the ripening phase lasts until harvest. At this stage, significant variations are observed in the colour, shape, and texture of the panicles. It is observed that as rice matures, the number of labels in each image increases constantly. Since the rice grows, it becomes impossible to observe the panicles complete due to their overlapping. The images with the resolution 4096×2160 are divided into three subsets of images with the resolution 128×128 , 256×256 , and 512×512 .

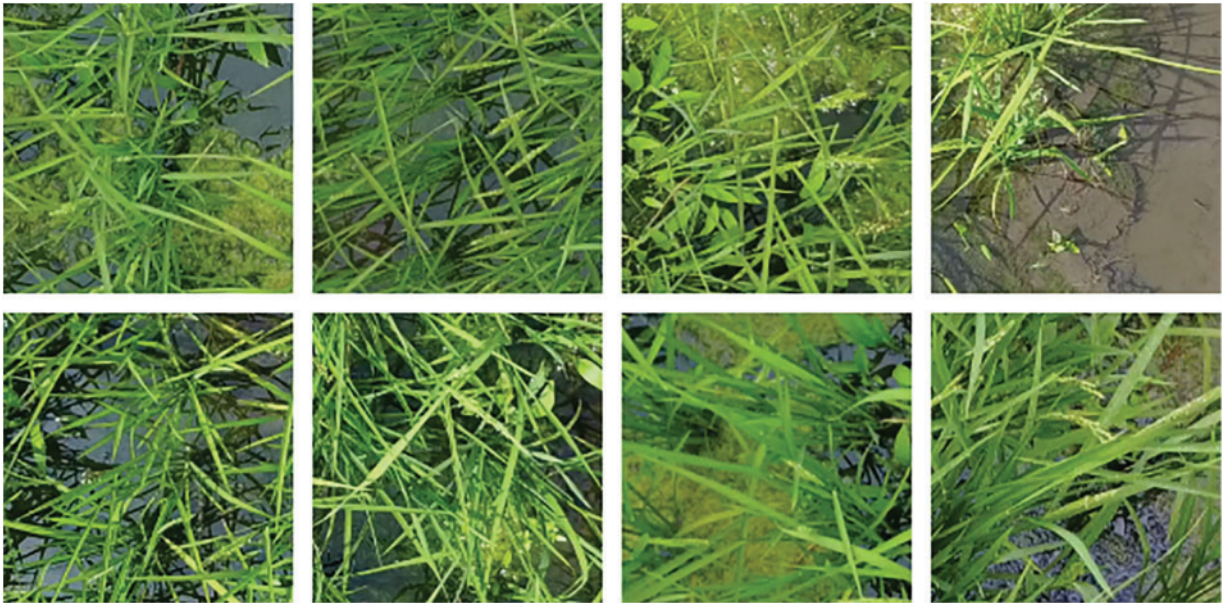


Figure 1: UAV paddy rice imagery dataset samples

Due to the non-overlapping sliding windows, it is unlikely that the same panicle appears in many images. With threshold and morphology filtering, we can refine the confidence map and obtain a refined dataset with manually labeled annotations. There may be some missed annotations since the annotations are manually labeled. It was, however, refined using a deep learning model with an improvement of 40.6%. A minimized receptive area of view cannot ensure panicle completion. Hence, in this study, a 512×512 subset is chosen to maintain a balance between positive and negative annotations. The positive label refers to the panicle, while the negative label refers to the background. The dataset consists of 7600 images. 70% of the dataset (5866) is used for training, 10% for validation (648), and 20% for testing (1086).

3.2 Border Sensitive Knowledge Distillation

In this section, we will elucidate the proposed Border Sensitive Knowledge Distillation (BSKD) model, as depicted in (Fig. 2). The first step involves an imitation mask. This is designed to separate the foreground features from the background characteristics present in the provided input images. Then, we proceed to explain the technique of feature-based distillation. This section provides a detailed explanation of the proposed teacher network, which utilizes the ResNet50 architecture along with a border sensitive mechanism. Additionally, the process by which the simple student model, ResNet18, distills knowledge from the teacher network is thoroughly described. The Border Sensitive Method is discussed in detail as an independent subsection. Finally, the overall loss calculation is shown.

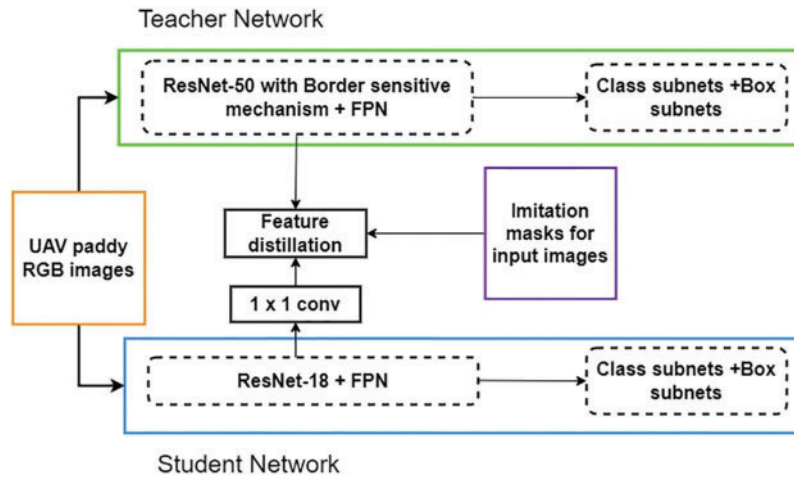


Figure 2: Proposed border sensitive knowledge distillation (BSKD)

3.2.1 Imitation Mask

The high-level features $f \in R^{C \times H \times W}$, which are generated in the final layer of backbones such as the C5 level features in ResNet50 backbone, are considered. As we employ a border sensitive mechanism at the C5 level in ResNet50, the high-level features f of each individual has strong representative ability and gather rich information in the global context, including border features. To obtain the vector representation f_1^K for the K instance, we compute the average of the feature vectors inside its corresponding RoI. The RoI area of the K instance is activated using a binary mask $M^K \in \{0, 1\}^{H \times W}$. The formal expression for f_1^K is as follows:

$$f_1^K = \frac{1}{n^K} \sum_{h=1}^H \sum_{w=1}^W M^K(h, w) \cdot f(h, w) \quad (1)$$

$$n^K = \sum_{h=1}^H \sum_{w=1}^W M^K(h, w) \quad (2)$$

Let $M^K(h, w)$ be equal to 1 and it is only considered when RoI of K instance surpasses the values of h and w . The entire number of feature vectors within the RoI is denoted as n^K . By comparing all of the pixel embeddings in the original features f with the class prototypes, we can create an imitation mask $M_{im} \in R^{H \times W}$. After min-max normalization, we obtain final imitation mask $M_{im} \in [0, 1]^{H \times W}$. The values within the imitation mask correspond to the probabilities assigned to pixel embeddings, indicating their likelihood of being classified as foreground instances. These probabilities are crucial for the student model to prioritise and focus on during the process of knowledge distillation. In order to effectively disentangle pixel embeddings into foreground and background, we employ a threshold factor φ within the range of $[0, 1]$. When $\varphi = 0$, all pixel embeddings are considered as foreground, while when $\varphi = 1$, all pixel embeddings are considered as background. This allows us to segment the pixel embeddings into two distinct categories: M_{fg} , which represents the foreground, and M_{bg} , which represents the background. M_{fg} and M_{bg} is applied on features extracted by Feature Pyramid Network (FPN). We employ the technique of interpolation to adjust the dimensions of the masks to align with the desired feature scales.

3.2.2 Feature Based Distillation

The Knowledge distillation process involves transferring knowledge from one machine learning model (teacher) to another (student). The teacher is a high-capacity, high-performance model, while the student is a compact, low-capacity model. The goal of knowledge transfer is to maximize the student's compactness without sacrificing performance. After each training step, the teacher model gives feedback to the student model, which is accomplished through a series of iterations. In response, the student model adjusts its parameters. Performance is measured until the desired level is reached. It is regarded as one of the most effective methods of model acceleration and compression. Knowledge distillation can be categorized into logit-based distillation, feature-based distillation, and relationship-based distillation. The feature-based distillation technique involves extracting intermediate features from the teacher network, resulting in a higher level of information than other distillation approaches. In the proposed border sensitive feature distillation approach, the ResNet50 architecture was adopted as the backbone network for the teacher model. The ResNet18 architecture was selected for the student model. The border sensitive mechanism is exclusively implemented for the teacher network at C5 level in ResNet50 (Fig. 3) to ensure that the features extracted are highly informative, even in the border region. The student network has the ability to reproduce the teacher's model behaviors with regards to the output features of the Feature Pyramid Network (FPN). To implement feature knowledge distillation, an adaptation layer is integrated into the student network. Specifically, a 1×1 convolution layer is selected and placed after each FPN output in the student model. This adaptation layer modifies the features' shape. The output features of the FPN can be represented as f_{FPN}^T for teacher network and f_{FPN}^S for student network, respectively. The feature-based distillation loss is calculated as follows:

$$L_{dist} = L_{mse}(f_{FPN}^T, \sigma(f_{FPN}^S)) \cdot (M_{fg} + \tau \cdot M_{bg}) \quad (3)$$

where L_{mse} denotes mean square error loss, σ denotes adaptation layer, and τ denotes a constant that balances the significance of background and foreground distillation. M_{fg} represents foreground imitation mask whereas M_{bg} represents background imitation mask.

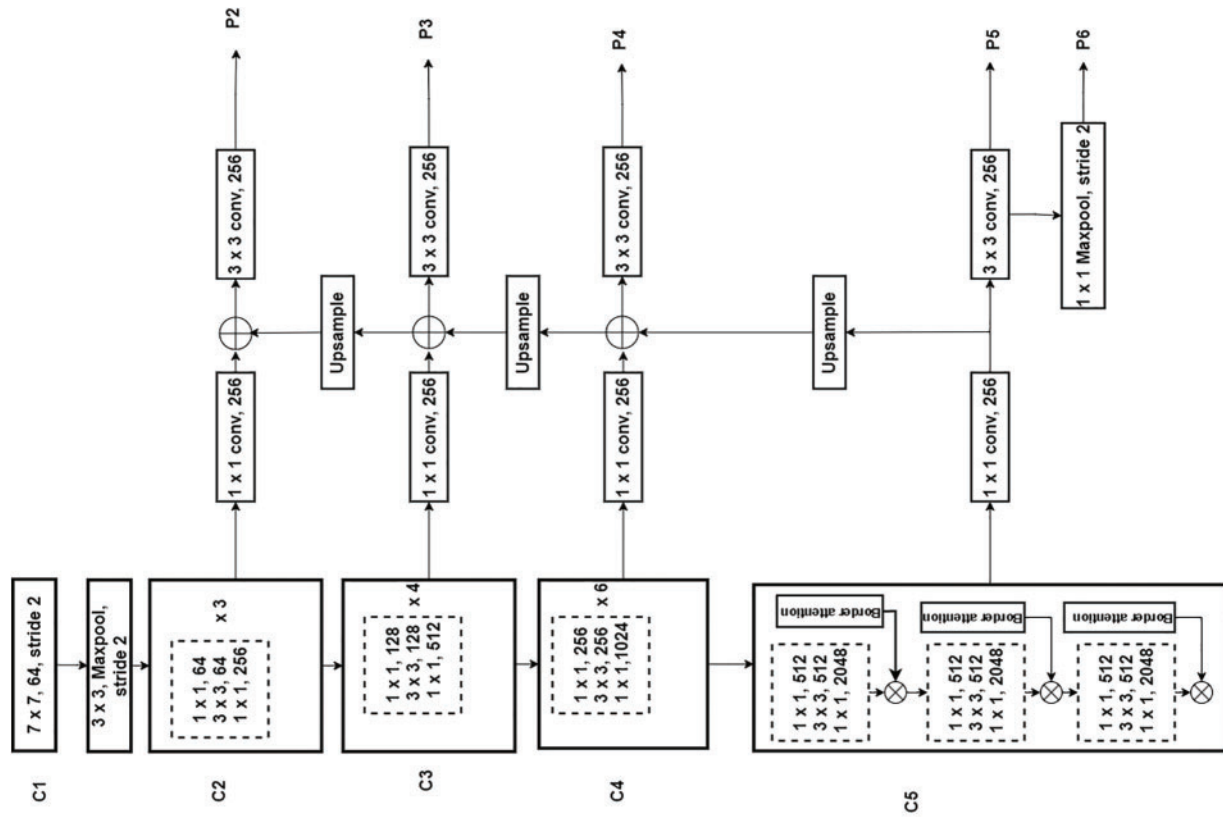


Figure 3: Teacher network–ResNet50+FPN+border sensitive method

3.2.3 Border Sensitive Method

Song et al. [23] proposed border sensitive mechanism inspired by human cortex activation mechanism. Based on a band shape kernel, a hierarchical activation function is applied for a more effective pooling strategy. Moreover, a further extension of strip max pooling has also been introduced to enhance the border characterization of prominent features. The fundamental limitations of existing activation functions necessitate implementing cortex activation in border-sensitive methods. Without a proper learning rate, even for a large gradient, neurons will not be activated after entering the ReLU module and altering their parameters, resulting in “neuron death”. In this study, the intrinsic advantage of the exponential function in establishing the activation mechanism of the cortex is capitalized on the problems of gradient vanishing. The overview of the proposed border sensitive method is illustrated in Fig. 4, where C represents the number of channels. Strip hierarchical pooling and strip max pooling are parallel branches, each consisting of horizontal and vertical strips. In the following step, a one-dimensional convolutional layer with a kernel size of three is applied. The next step involves modulating the current position by its neighboring features, which produces four strip features as a result. The cortex hierarchical activation function is capable of extracting a wide range of information during the process of downsampling. It can be formulated as:

$$\bar{X} = e^{\alpha \cdot x + \beta} \tag{4}$$

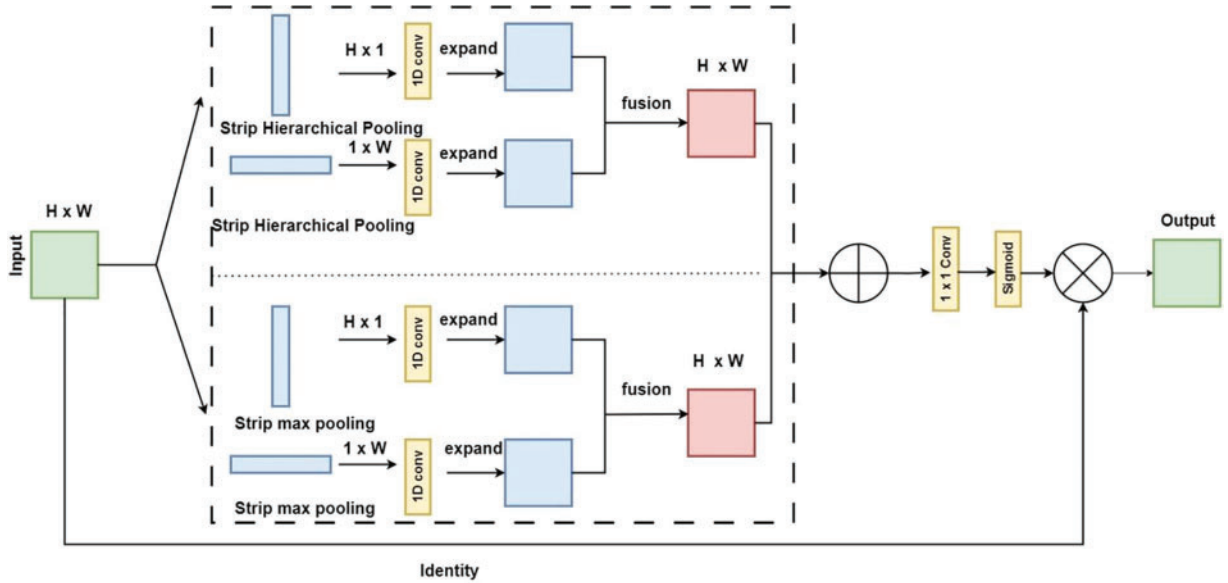


Figure 4: Border sensitive method

Here, α and β are hyperparameters and \bar{X} is the cortex hierarchical activation by input X . The final output \hat{A}_h of hierarchical pooling operation can be calculated by aggregating all weighted activations based on the kernelized neighborhood R ,

$$\hat{A}_h = \sum_{i \in R} w_i * a_i \quad (5)$$

where w_i refers to weighted activation that is computed as the ratio of the hierarchical activation of the i^{th} cortex to the sum of all its activations in R ,

$$w_i = \frac{\bar{a}_i}{\sum_{j \in R} \bar{a}_j} \quad (6)$$

In this work, we determined the hyperparameters value $\alpha = 1$ and $\beta = 0$ due to the fact that deep neural networks normalize their output using softmax functions, and even though we have an attention operation embedded within the layer, we are not required to manually limit the active function of the network to a particular range. The weights w_i , along with the activation values a_i , are implemented as nonlinear transformations in a kernel region, in contrast to average pooling.

As the activation level increases, there is a corresponding increase in border feature dominance. The efficacy of this activation highlighting mechanism surpasses that of the traditional direct averaging pooling approach. The effective use of a band-shaped kernel is implemented to execute a strip pooling operation in either the vertical or horizontal dimension. The pooling operation computes the average value of all the feature elements within a given row or column, based on spatial dimensions of $(1, W)$ or $(1, H)$. Following vertical strip pooling, the output $Y^W \in R^W$ is:

$$Y_i^W = \frac{1}{H} \sum_{0 \leq j \leq H} X_{i,j} \quad (7)$$

As a result of, horizontal strip pooling, the output $Y^H \in R^H$ is:

$$Y_i^H = \frac{1}{W} \sum_{0 \leq j \leq W} X_{i,j} \quad (8)$$

In addition to establishing long-term dependencies between the distributed regions, the pooling operation described above encapsulates the regions in a banded shape. This allows the whole downsampling process to emphasize more on the differences between border features, rather than preserving the overall feature information. This method encapsulates strip pooling into a border-sensitive mechanism aimed at preventing model inflation. Strip hierarchical pooling becomes more effective when we implement cortex hierarchical activation into strip pooling. Therefore, the kernel size becomes $(1, W)$ and $(H, 1)$, accordingly. Along with the addition of strip hierarchical pooling, a branch of strip max pooling is also introduced. This branch is to conserve texture information on the border as much as possible to improve boundary feature extraction. The function of max pooling is calculated as follows:

$$\hat{a}_m = \max_{i \in R} a_i \quad (9)$$

Strip max pooling and strip hierarchical pooling combine Y^W and Y^H to derive $Y \in R^{CxHxW}$, which captures more relevant global information.

$$Y_{c,i,j}^S = Y_{c,i}^H + Y_{c,j}^W \quad (10)$$

Consequently, we can combine strip hierarchical pooling and strip max pooling by performing the following:

$$Y_{c,i,j}^{OverAll} = Y_{c,i,j}^{SHP} + Y_{c,i,j}^{SMP} \quad (11)$$

The final output for parallel joint pooling can be calculated as:

$$Output\ y = scale(X, \sigma(F(Y_{c,i,j}^{OverAll}))) \quad (12)$$

Here, *scale* () operation refers to pixel wise multiplication, *F* refers 1×1 convolution and σ refers to a sigmoid function. The border sensitive method is lightweight, effective, and efficient. The method avoids a lot of computation required to establish relationships between the two locations. It is easy to integrate this approach into a wide range of backbone networks.

3.2.4 Total Loss Function

The total training loss L is:

$$L = L_{OD} + \alpha L_{FDistill} \quad (13)$$

where L_{OD} denotes the original training loss of object detectors, $L_{FDistill}$ refers to the feature distillation loss and α is a hyperparameter which balances both detection and distillation losses.

3.3 Experimental Setup

The proposed work is implemented on Windows 11OS (CPU: AMD Ryzen 7 6800HS; RAM: 32GB; GPU: NVIDIA GeForce RTX 3050 Ti). Open-source deep learning frameworks MMDetection [24] and PyTorch are used to implement deep learning models. ResNet50 and ResNet18 are the backbones of teacher and student networks. The initial learning rate is set to 0.002, the weight decay is set to 0.0001, and the momentum is adjusted to 0.9 for all experiments. The value of α is assigned

as 5×10^{-4} , whereas the parameter φ is adjusted to 0.6. Pre-trained weights from ImageNet are used to initialize the backbone network.

Algorithm 1: Border Sensitive Method

Input: $X \in R^{C \times H \times W}$

Output: $y \in R^{C \times H \times W}$

- 1: Calculate vertical strip hierarchical pooling using Eq. (7) and Eq. (5).
 - 2: Increase the dimension to $X_1 = H \times W$.
 - 3: Calculate horizontal strip hierarchical pooling using Eq. (8) and Eq. (5).
 - 4: Increase the dimension to $X_2 = H \times W$
 - 5: $Y_{SHP} = \text{relu}(X_1 + X_2)$
 - 6: Calculate vertical strip max pooling using Eq. (7) and Eq. (9)
 - 7: Increase the dimension to $X_3 = H \times W$.
 - 8: Calculate horizontal strip max pooling using Eq. (8) and Eq. (9)
 - 9: Increase the dimension to $X_4 = H \times W$.
 - 10: $Y_{SMP} = \text{relu}(X_3 + X_4)$
 - 11: Calculate the parallel joint pooling using Eq. (11)
 - 12: Perform element-wise multiplication using Eq. (12)
 - 13: **return** Output y
-

4 Results and Discussion

4.1 Ablation Study

4.1.1 Hyper Parameter Sensitivity

The proposed BSKD model is subjected to a sensitivity study regarding the two most relevant hyperparameters, α and φ . It is evident from the model that α impacts the imitation mask distribution. Similarly, the value of φ impacts the amount of distillation loss. Therefore, distillation losses can be controlled. In our experiments, we varied α and φ values from 0 to 10, and observed changes in mAP. In Fig. 5, the results of BSKD are depicted on UAV paddy rice imagery to illustrate the detection of panicles. In Fig. 5a, we observe that the student model gains less from knowledge distillation when α is too low. Similarly, when the α value increases, the student's mean average precision increases as well. Thus, the α value has a direct impact on the mean average precision value. In contrast, when α is excessively large (more than 8), performance suffers significantly. It could be due to the fact that a substantial distillation loss would interfere with the training process. As we can see from Fig. 5b, we can almost observe the same kind of φ pattern. There is a possibility that the reason for this could be that background information has been highlighted to a significant extent, while panicle-related information has been overlooked to a significant extent. It is especially true in the case of small panicles, when the φ is too large.

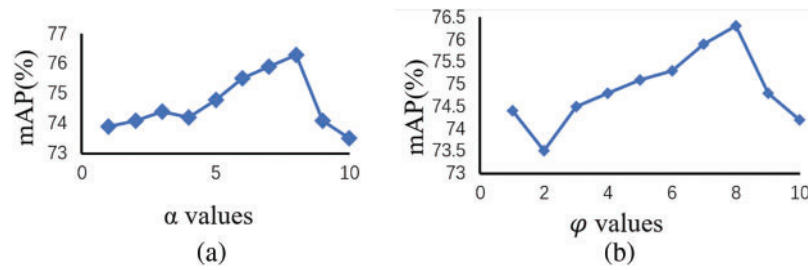


Figure 5: Study of α and φ 's hyperparameter sensitivity (a) α 's influence (b) φ 's influence

4.1.2 Impact of Various Adaptation Layers

The performance of various adaptation layer types on the UAV paddy dataset with BSKD is shown in Table 1. There are two networks that form the backbone of the teacher and student networks: ResNet50 and ResNet18. In this paper, we analyze four different types of adaptation layers: 1×1 conv +BN, 1×1 conv, 3×3 conv +BN, and 3×3 conv. It is made clear that “conv” stands for a layer of convolution, while “+BN” will be used to refer to the batch normalization that goes after the convolutional layer. According to the results of this study, the difference in mean Average Precision (mAP) between the adverse outcome (3×3 conv +BN) and the highest result (1×1 conv) is only 1.1. This is a clear indication that the proposed BSKD is not overly dependent on the selection of the adaptation layer.

Table 1: Result of various adaptation layers

Adaptation layer	3×3 conv+BN	3×3 conv	1×1 conv+BN	1×1 conv
mAP	75.2	75.5	75.9	76.3

4.1.3 Impact of Different Teachers and ResNet18 Student

In the context of image classification, it has been observed that the student model's performance tends to deteriorate when there is a significant disparity in performance between the teacher and the student models. In this section, we assess the validity of this observation in the context of UAV-based paddy panicle detection using the proposed BSKD model. ResNet18 is employed as the backbone network for the student model, and knowledge distillation is performed using various teacher sizes. Table 2 presents the findings. There exists a positive link in performance between the teacher and the student networks. The presence of a highly effective teacher is positively correlated with a greater improvement in performance for the respective student. This observation is different from the image classification findings. The rationale for this observation can be attributed to the fact that the distillation process in image classification primarily focuses on optimizing the final classification logits. In contrast, our approach to detecting paddy panicles using drones focuses on manipulating intermediate high-level features. When using one-hot labels as the target, the clean logits would not adequately capture the upgraded teachers' expanded knowledge. It may be inferred that the presence of a highly qualified teacher is imperative for knowledge distillation implementation.

Table 2: Result of various teachers and ResNet18 student

	ResNet18	ResNet34	ResNet50	ResNet101
Teacher	72.8	75.7	80.4	82.2
Student	72.8			
BSKD	74.1	75.4	76.3	76.9

4.2 Quantitative and Qualitative Detection Comparison

A quantitative comparison of the proposed detection method with various State-Of-The-Art (SOTA) detectors is shown in Table 3. According to the results derived from the developed BSKD algorithm, the performance of the AP_{50} algorithm is improved over that of the AP_{75} algorithm. The results of this observation provide us with valuable insight and suggest that a threshold for IoU of 0.50 might be appropriate to use. Similarly, the Average Precision (AP) value increases as the object size increases from small (s) to large (l). AP_s represents the average precision for small objects with the size less than 32^2 pixels. The value of AP_m indicates the average precision for medium objects with a size greater than 32^2 pixels but less than 96^2 pixels, while AP_l indicates the average precision of large objects with a size greater than 96^2 pixels. As a result of these findings, we can conclude that the proposed model doesn't perform well when implemented on small panicles. It appears that the proposed detection method is capable of exhibiting superior performance to other SOTA detectors, including both the one-stage and two-stage models (Fig. 6). Yolov8, Yolov3 and SSD were chosen as representatives of one-stage models, and Faster RCNN, mask RCNN and cascade RCNN were selected as representatives of two-stage models. The two-stage models exhibited superior performance to the one-stage SSD, yielding improved results in terms of mean average precision (mAP) values. Yolov8 demonstrates superior performance than two-stage detectors. However, the BSKD model is a single-stage model and shows better accuracy to other detectors. For instance, the mean average precision (mAP) of BSKD is 76.3%. In contrast, the mAP values for Yolov8, Yolov3, SSD, Faster RCNN, mask RCNN, and the cascaded RCNN stand at 65.3%, 64.2%, 53.7%, 54.2%, 54.5% and 63.6%, respectively. The mean average precision (mAP) of BSKD showed substantial improvements of 11%, 12.1%, 22.6%, 22.1%, 21.8% and 12.7% when compared to Yolov8, Yolov3, SSD, Faster RCNN, mask RCNN, and cascaded RCNN, respectively. The visualization of bounding box regression using the proposed method and SOTA detectors cascaded RCNN and Yolov8 is presented in Fig. 7. Cascaded RCNN is selected from other two-stage detectors and Yolov8 is selected from other one-stage detectors for visualization based on accuracy. According to observations, other SOTA detectors cascaded RCNN and Yolov8 cannot detect panicles around the edge. However, if the panicles are larger, they can detect them. In addition, the proposed method demonstrates a reduced convergence of the bounding box regression training loss compared to Yolov8 and cascaded RCNN.

It is possible that the primary reason for the outperforming results observed with the proposed model BSKD has to do with a border sensitive approach used for identifying panicles around the border and this in turn leads to an improvement in the total mean average precision. Considering that the UAV rice imagery dataset included samples with different illuminations, panicle sizes, shape distortions, partial occlusions, and complex backgrounds, the increased accuracy of the proposed model may indicate that it is capable of resolving these problems better than SOTA detectors. The main limitation of the proposed model is the fact that due to its border-sensitive nature; it overlooks objects

that are located in the vicinity of the border. Thus, it is possible for other objects to be mistakenly detected as panicles around the border, and the performance as a result will degrade.

Table 3: Comparison with the SOTA detectors

Model	Type	AP	AP ₅₀	AP ₇₅	AP _s	AP _m	AP _l
Yolov3	One stage	38.8	62.6	39.9	26.7	44.3	50.7
SSD	One stage	39.6	63.1	40.4	27.2	45.1	52.8
Yolov8	One stage	40.8	64.2	41.5	27.9	46.4	53.6
Faster RCNN	Two-stage	40.2	63.4	40.9	27.4	45.3	53.1
Mask RCNN	Two-stage	40.4	63.9	41.3	27.8	45.6	53.2
Cascade RCNN	Two-stage	41.2	64.5	41.9	28.1	46.5	53.7
BSKD (proposed)	One stage	42.7	65.2	42.4	29.5	47.2	56.4

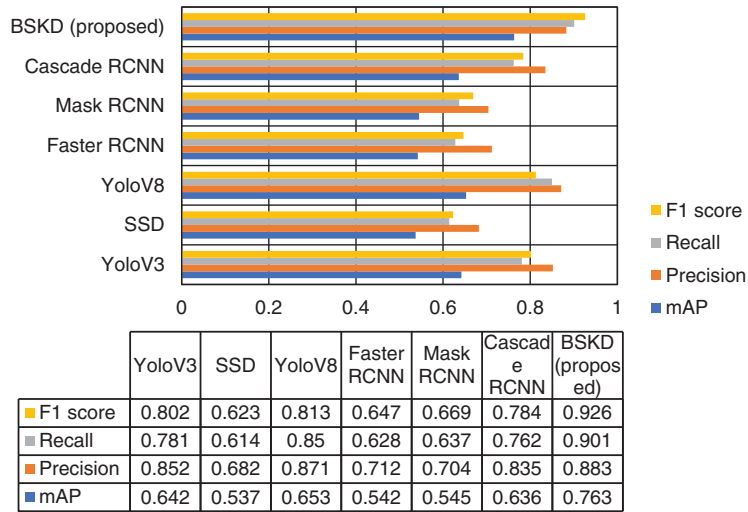


Figure 6: Overall performance comparison with the SOTA detectors

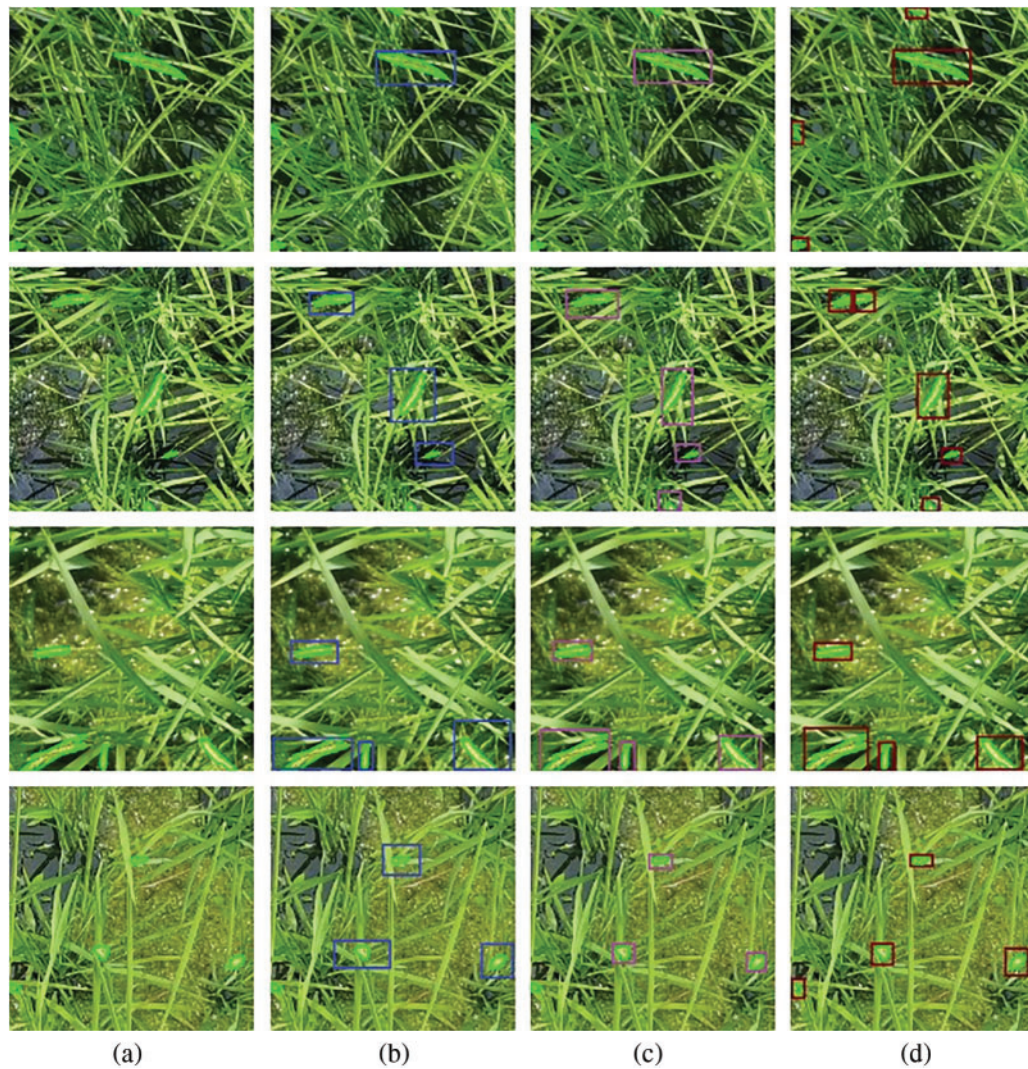


Figure 7: Qualitative comparison of proposed BSKD with SOTA detectors (a) Ground truth (b) Cascaded RCNN prediction (c) YOLOv8 prediction (d) Proposed BSKD prediction

5 Conclusion

This study introduces a novel approach called Border Sensitive Knowledge Distillation (BSKD) to detect rice panicles in images captured by Unmanned Aerial Vehicles (UAVs). This research focuses on the use of panicles located along borders to improve detection capabilities. Initially, the input images are acquired through the complicated teacher network, which consists of ResNet50, border sensitive method (BSM), and feature pyramid network (FPN). The generation of imitation masks extracts both foreground and background elements. These masks and intermediate features acquired by the teacher model are distilled by the simple student model, specifically the ResNet18+FPN architecture. The aggregate loss is determined through the summation of both detection and distillation losses. The proposed BSKD model demonstrates improved performance compared to state-of-the-art (SOTA) object detectors when evaluated on the UAV paddy dataset. Further research will focus on improving precision by analyzing images taken by UAVs at high altitudes and extremely small panicles.

Acknowledgement: The authors are grateful to all the editors and anonymous reviewers for their comments and suggestions and thank all the members who have contributed to this work with us.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Anitha Ramachandran; data collection: Anitha Ramachandran; analysis and interpretation of results: Anitha Ramachandran, Sendhil Kumar K.S.; draft manuscript preparation: Anitha Ramachandran. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are openly available at <https://zenodo.org/records/4444741> (accessed on 06 August 2024).

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] N. K. Fukagawa and L. H. Ziska, "Rice: Importance for global nutrition," *J. Nutr. Sci. Vitaminol.*, vol. 65, pp. S2–S3, Oct. 2019. doi: [10.3177/jnsv.65.S2](https://doi.org/10.3177/jnsv.65.S2).
- [2] G. K. Dash, S. K. Sahoo, M. Barik, S. Parida, M. J. Baig and P. Swain, "Perspectives and challenges of phenotyping in rice research," in *Applications of Bioinformatics in Rice Research*, M. K. Gupta, L. Behera Eds. Singapore: Springer Singapore, 2021, pp. 171–195, doi: [10.1007/978-981-16-3997-5_9](https://doi.org/10.1007/978-981-16-3997-5_9).
- [3] M. Ikeda *et al.*, "Analysis of rice panicle traits and detection of QTLs using an image analyzing method," *Breed. Sci.*, vol. 60, no. 1, pp. 55–64, 2010. doi: [10.1270/jsbbs.60.55](https://doi.org/10.1270/jsbbs.60.55).
- [4] M. K. Vishal *et al.*, "Image-based phenotyping of diverse rice (*Oryza sativa* L.) Genotypes," 2020, *arXiv: 2004.02498*.
- [5] C. A. Devia *et al.*, "Aerial monitoring of rice crop variables using an UAV robotic system," in *ICINCO 2019–Int. Conf. Inform. Control, Autom. Robot.*, Prague, Czech Republic, Jul. 29–31, 2019. doi: [10.5220/0007909900970103](https://doi.org/10.5220/0007909900970103).
- [6] A. K. Sangaiah, F. -N. Yu, Y. -B. Lin, W. -C. Shen, and A. Sharma, "UAV T-YOLO-Rice: An enhanced tiny yolo networks for rice leaves diseases detection in paddy agronomy," *IEEE Trans. Netw. Sci. Eng.*, pp. 1–16, 2024. doi: [10.1109/TNSE.2024.3350640](https://doi.org/10.1109/TNSE.2024.3350640).
- [7] F. -N. Yu, W. -C. Shen, A. K. Sangaiah, and Y. -B. Lin, "DSC-T-Yolo-Rice: A sand clock yolo model for rice leaves diseases detection," in *2024 IEEE Int. Conf. Consum. Electron. (ICCE)*, Las Vegas, NV, USA, IEEE, Jan. 2024, pp. 1–6, doi: [10.1109/ICCE59016.2024.10444270](https://doi.org/10.1109/ICCE59016.2024.10444270).
- [8] W. Wu *et al.*, "Image analysis-based recognition and quantification of grain number per panicle in rice," *Plant Methods*, vol. 15, no. 1, Dec. 2019, Art. no. 122. doi: [10.1186/s13007-019-0510-0](https://doi.org/10.1186/s13007-019-0510-0).
- [9] W. Yang *et al.*, "Genome-wide association study of rice (*Oryza sativa* L.) leaf traits with a high-throughput leaf scorer," *J. Exp. Bot.*, vol. 66, no. 18, pp. 5605–5615, Sep. 2015. doi: [10.1093/jxb/erv100](https://doi.org/10.1093/jxb/erv100).
- [10] H. Li *et al.*, "An automatic approach for detecting seedlings per hill of machine-transplanted hybrid rice utilizing machine vision," *Comput. Electron. Agric.*, vol. 185, no. 06, Jun. 2021, Art. no. 106178. doi: [10.1016/j.compag.2021.106178](https://doi.org/10.1016/j.compag.2021.106178).
- [11] W. Guo, T. Fukatsu, and S. Ninomiya, "Automated characterization of flowering dynamics in rice using field-acquired time-series RGB images," *Plant Methods*, vol. 11, no. 1, Dec. 2015, Art. no. 7. doi: [10.1186/s13007-015-0047-9](https://doi.org/10.1186/s13007-015-0047-9).

- [12] M. -D. Yang, H. -H. Tseng, Y. -C. Hsu, C. -Y. Yang, M. -H. Lai and D. -H. Wu, "A UAV open dataset of rice paddies for deep learning practice," *Remote Sens.*, vol. 13, no. 7, Apr. 2021, Art. no. 1358. doi: [10.3390/rs13071358](https://doi.org/10.3390/rs13071358).
- [13] H. -H. Tseng, M. -D. Yang, R. Saminathan, Y. -C. Hsu, C. -Y. Yang and D. -H. Wu, "Rice seedling detection in UAV images using transfer learning and machine learning," *Remote Sens.*, vol. 14, no. 12, Jun. 2022, Art. no. 2837. doi: [10.3390/rs14122837](https://doi.org/10.3390/rs14122837).
- [14] L. Yu *et al.*, "An integrated rice panicle phenotyping method based on X-ray and RGB scanning and deep learning," *Crop J.*, vol. 9, no. 1, pp. 42–56, Feb. 2021. doi: [10.1016/j.cj.2020.06.009](https://doi.org/10.1016/j.cj.2020.06.009).
- [15] Z. Qiu, H. Xiang, F. Ma, and C. Du, "Qualifications of rice growth indicators optimized at different growth stages using unmanned aerial vehicle digital imagery," *Remote Sens.*, vol. 12, no. 19, Oct. 2020, Art. no. 3228. doi: [10.3390/rs12193228](https://doi.org/10.3390/rs12193228).
- [16] X. Wang *et al.*, "Field rice panicle detection and counting based on deep learning," *Front Plant Sci.*, vol. 13, Aug. 2022, Art. no. 966495. doi: [10.3389/fpls.2022.966495](https://doi.org/10.3389/fpls.2022.966495).
- [17] M. M. Anuar, A. A. Halin, T. Perumal, and B. Kalantar, "Aerial imagery paddy seedlings inspection using deep learning," *Remote Sens.*, vol. 14, no. 2, Jan. 2022, Art. no. 274. doi: [10.3390/rs14020274](https://doi.org/10.3390/rs14020274).
- [18] B. Sun *et al.*, "Universal detection of curved rice panicles in complex environments using aerial images and improved YOLOv4 model," *Front Plant Sci.*, vol. 13, Nov. 2022, Art. no. 1021398. doi: [10.3389/fpls.2022.1021398](https://doi.org/10.3389/fpls.2022.1021398).
- [19] C. Zhou *et al.*, "Automated counting of rice panicle by applying deep learning model to images from unmanned aerial vehicle platform," *Sensors*, vol. 19, no. 14, Jul. 2019, Art. no. 3106. doi: [10.3390/s19143106](https://doi.org/10.3390/s19143106).
- [20] Q. Yang, L. Shi, J. Han, Y. Zha, and P. Zhu, "Deep convolutional neural networks for rice grain yield estimation at the ripening stage using UAV-based remotely sensed images," *Field Crops Res.*, vol. 235, pp. 142–153, 2019. doi: [10.1016/j.fcr.2019.02.022](https://doi.org/10.1016/j.fcr.2019.02.022).
- [21] X. Bai *et al.*, "Locating and sizing method based on high-throughput UAV RGB images," *Plant Phenomics*, vol. 5, 2023, Art. no. 0020. doi: [10.34133/plantphenomics.0020](https://doi.org/10.34133/plantphenomics.0020).
- [22] H. Wang, S. Lyu, and Y. Ren, "Paddy rice imagery dataset for panicle segmentation," *Agronomy*, vol. 11, no. 8, 2021, Art. no. 1542. doi: [10.3390/agronomy11081542](https://doi.org/10.3390/agronomy11081542).
- [23] Y. Song, P. Zhang, W. Huang, Y. Zha, T. You and Y. Zhang, "Object detection based on cortex hierarchical activation in border sensitive mechanism and classification-GIoU joint representation," *Pattern Recognit.*, vol. 137, no. 2, May 2023, Art. no. 109278. doi: [10.1016/j.patcog.2022.109278](https://doi.org/10.1016/j.patcog.2022.109278).
- [24] K. Chen *et al.*, "MMDetection: Open MMLab detection toolbox and benchmark," Jun. 17, 2019, *arXiv.1906.07155*.