# Evaluating the Effectiveness of YOLO Models in Different Sized Object Detection and Feature-Based Classification of Small Objects

Luyl-Da Quach *, Khang Nguyen Quoc, Anh Nguyen Quynh, and Hoang Tran Ngoc

Software Engineering Department, FPT University, Cantho City, Vietnam;
Email: KhangNQCE160710@fpt.edu.vn (K.N.Q.), anhnqce160515@fpt.edu.vn (A.N.Q.), hoang2531992@gmail.com (H.T.N.)
*Correspondence: luyldaquach@gmail.com (L.D.Q.)

***Abstract*—The YOLO tool has been increasingly developed to assist in object classification. However, the problem of object classification has many difficulties, including small objects, background effects, or noise loss of information. Therefore, to evaluate the objective of classifying small, information-losing objects, the research team installed and evaluated new YOLO models such as YOLOv5, YOLOv6, and YOLOv7. In addition, the study conducted a feature-based object classification test for comparison and evaluation. The study was conducted on the self-collected data set, divided into 2 datasets: a dataset used to evaluate object classification and a dataset used to classify by features. The evaluation results show certain advantages of the YOLOv7 model on parameters such as Precision, Recall, and a mAP threshold of 50%. The evaluation results show certain advantages of the YOLOv7 model on parameters such as Precision, Recall, and a mAP threshold of 50%. The study results show that YOLOv7 achieves specific effects when the accuracy of object recognition is over 90%, in which the feature-based classification also achieves an accuracy of over 70%. This issue may need different future studies in object recognition and object feature recognition.**

***Keywords*—YOLOv5, YOLOv6, YOLOv7, small object, rice seed, object detection, feature classification**

## I. INTRODUCTION

A small subject is part of a large image, which is why the proportion of small objects in the image is very small. This is why small objects will have low resolution, and very little information can be extracted efficiently. Detecting or processing information on small objects faces many challenges in deep learning, mainly with three main difficulties: 1) Small objects should lack information as a feature for CNN to create a layer; 2) The effect of a complex background, which causes the image to be noisy by other objects; 3) The noise and image quality that makes it difficult to detect small [1]. However, advancing new YOLO models has solved some complex problems in small object recognition. Especially in the problem of detecting diseases on plants, small objects such as diseases on leaves and fruits are in different positions on the tree. In addition, the rice grain disease dataset is often noisy by environmental factors such as light and shooting angle, making identifying and classifying small objects more difficult. To solve this problem, in this paper, the research team proposes to evaluate new models such as YOLOv5, YOLOv6, and YOLOv7 to identify and classify diseases in rice seeds. The research team's method focuses on processing small objects in the rice grain disease dataset by feature extraction techniques of the YOLO network and some advanced image processing techniques. The research team evaluates the effectiveness of its method on real data sets and compares it with other identification and classification methods. The results show that the research method has high accuracy and can be applied to solve the problem of disease identification and classification in rice in practical applications.

Rice is an essential food in Asian countries, essential in daily meals. But the supply is not only in Asia but also in the world, typically Africa contributes 2.95%, America 5.19%, Europe 0.67%, Oceania 0.15%, and the rest in Asia [2]. As expected, the demand for output will increase shortly. However, damage to rice plants due to diseases and environmental factors is urgent. However, early visual observation is the only technique to diagnose rice diseases. However, to detect disease problems in rice early, many studies have been based on deep learning and machine learning algorithms. In particular, studies using Convolutional Neural Networks (CNN) to detect diseases on leaves [3–7], image processing [8–11], machine learning [12–14], transfer learning [15, 16] and combine techniques [17–19]. However, identifying the disease in rice seeds is only done in the laboratory. Including sequential research in the use of chemical methods [20]. In addition, advanced artificial intelligence processing techniques to identify diseases from seeds, such as: using terahertz to identify rice grains resistant to blight [21], Optimal Hyperspectral Wavebands application to detect blight [22], classification of plant seeds [23], using cells made in the laboratory [24]. However, this method is quite

Manuscript received April 27, 2023; revised May 20, 2023; accepted June 2, 2023; published September 7, 2023.

doi: 10.12720/jait.14.5.907-917

expensive to perform because of the need for laboratory and high-end equipment, so it costs a lot.

Therefore, the study proposes a method to identify rice grain blight disease by the current new YOLO models, namely YOLOv5, YOLOv6, and YOLOv7. Due to the characteristics of rice grains with many grains on the image of a branch, the study also evaluates the accuracy of YOLO models in identifying and classifying small objects. Based on the data shown as illustrated in Fig. 1, the challenge of the research in processing: the image after rice grain separation has a very small resolution, and there is a lot of noisy data such as the body and branches on rice paddies.

Figure 1. Image showing the original unprocessed rice branch.

## II. Literature Review

This section will present the studies that used YOLOv5, YOLOv6, and YOLOv7 for the problem of small object recognition and disease classification. From this content, the research can evaluate related studies and determine the direction of applied research.

The beginning of the interest in developing a small object recognition system in 2021 is the YOLOv5 model. This model has been extensively studied by many studies focusing on exploiting related aspects affecting the image of small objects such as Unmanned Aerial Vehicle (UAV), and small components. Zhan *et al.* Mahaur and Mishra, Li *et al.* [25–27] proposed a method to improve the detection accuracy of small objects, and the model increases the accuracy significantly in the self-driving system. Benjumea *et al.* [28] and Liu *et al.* [29] improves the YOLOv5 model to YOLO-Z to detect small objects in remote sensing images. Liu *et al.* [30] proposes to use PANet and BiFPN to increase small object detection. Xuan *et al.* [31] and Li *et al.* [32] are used to detect small objects underwater. In general, current studies need to improve the YOLOv5 model to identify small objects accurately.

The YOLOv6 model has been developed since 2022, and its appearance has shown more improvement problems than YOLOv5 [33]. Azizah and Fatichah [34] uses YOLOv6 to detect Tajweed, in which the comparison of YOLOv5, YOLOv6, and YOLOv7 models shows that YOLOv7 is more effective. Weng *et al*. [35] used in the hardware identification of electronic circuits is a relatively small design and is needed in the identification process. John and Meva [36] compares YOLOv3 to YOLOv7 models for vehicle identification and counting. In general, studies have been conducted on the ability to recognize small objects and make comparisons and improvements, but currently are still using fixed images and have yet to lose information.

Finally, the YOLOv7 model [37] is also developed in 2022, YOLOv7 has shown outstanding progress and quite good hardware usability. Zhao *et al.* [38] build the YOLOv7-sea model to detect small maritime objects and noise caused by large sea surfaces. Zhao and Zhu [39] used the MS-YOLOv7 model to activate SoftNMS and Mish to improve the ability to identify many objects and be obscured when using UAVs. In order to improve the high omission detection rate of the YOLOv7 algorithm, research [40] has made improvements to the YOLOv7-RAR algorithm to detect vehicles on urban roads in terms of small targets such as perspective and Incomplete feature extraction, the accuracy of the algorithm is 95.1% higher than the original algorithm of 2.4%. Tang *et al.* [41] is used in detecting different objects at long distances. In these studies, YOLOv7 also needed specific improvements to recognize small and different-sized objects.

In summary, through the study, we found that the YOLOv5 to YOLOv7 models have made significant improvements in the recognition of small and different-sized objects. However, the problem of the complexity of objects when obscured, lost information, etc., is still a complicated issue and needs to be raised in the comparison process. In addition, through the survey, there still needs to be research that goes into identifying and analyzing features when using YOLOv5, YOLOv6, and YOLOv7 models. Therefore, the study evaluated the efficiency of recognizing objects of different sizes and classifying them based on their characteristics, namely the diseased rice dataset.

## III. YOLO Algorithm and Version

The You Only Look One (YOLO) model is one of the famous models in computer vision, especially in the Object recognition task. The YOLO model uses deep learning techniques to quickly and accurately identify and classify objects in the image. This model can give information about the coordinates of the bounding box of those objects and also determine their class. This algorithm will divide the input images into $S \times S$ grids (where S is the number of cells). The bounding boxes will be determined according to 5 parameters, including creating degrees (x, y) as the center of the bounding box in the current cell, the width and height of the bounding box, and the probability that the bounding box contains the object. In the process, the model will apply the loss function optimization method by backpropagation and gradient descent. So, YOLO models use a neural network to learn and improve the model's accuracy so that the model can improve over time on large datasets. However, the YOLO model also has some limitations. In some cases, the accuracy is lower than that of its predecessors, especially for small and discriminant objects within a relatively close range (narrow). The basic model of YOLO's Neural network with 24 Convolutions layers is shown in Fig. 2.

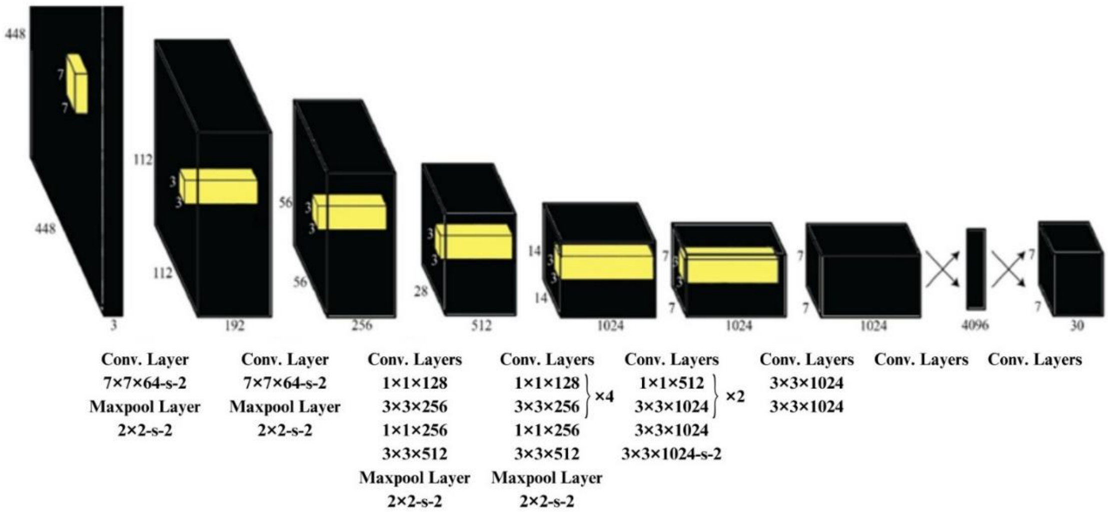
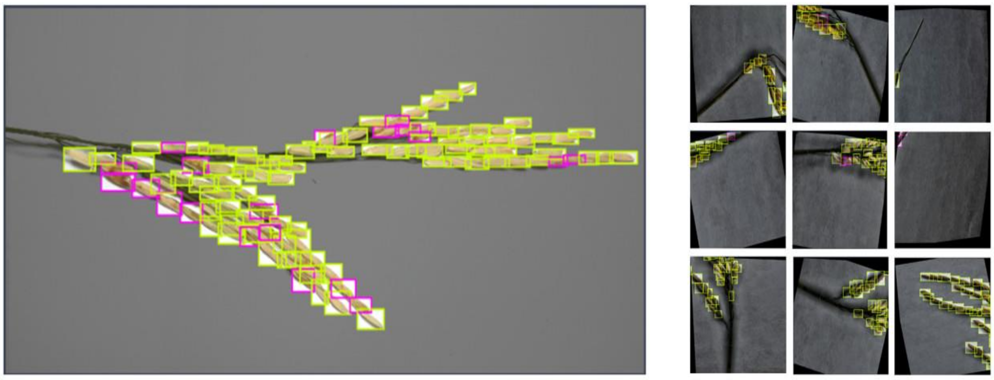



Figure 2. Convolution structure 24 layers convolution in YOLO model.



(a) (b)

Figure 3. Rice image sample in the dataset. (a) Image after annotated (b) Image with augmentation and color enhancement and contouring.

On the other hand, the 24-layer Convolution structure is widely used, so YOLO has undergone a tremendous development from V1 to V7 with many tasks in image recognition and the fields of robotics, UAV, and identification in real-time, especially the last three released versions, including YOLOv5, YOLOv6, and YOLOv7. The first is the version 5 model, which is widely applied in the identification of Objects and serves as the background structure for other recognition models. One of this versions' outstanding features is the ability to self-train, which the model automatically learns from unlabeled images. When the model detects new objects in these images, it automatically adds them to the training dataset to improve its detection. Since then, applying pre-train models and Transfer Learning techniques has become easy and highly effective since YOLOv5 models and later. YOLOv5's Self-training feature is essential as it improves the object detection and classification of the model without investing a lot of time and effort in creating a new training dataset. In addition, studies demonstrate that YOLOv5 improves the processing time of Deep Learning networks compared to previous versions. Yolov6 performs better than YOLOv5 when benchmarking against the MS Coco dataset [42–44]. This version model, which is applicable and highly effective for hardware, applies to real-time identification and drones. The author confirms that YOLOv7 has outperformed the object development models [37]. Although little known, the version 7 model has a very high processing time and results, especially on MS Coco sets, compared to one-stage or two-stage models. However, there are still a few reviews between versions, especially on objects of small size and almost similar shape, so we will create the most objective environment for three models through which to evaluate the effectiveness of data sets with complex characteristics.

## IV. Materials and Methods

The study proposes to evaluate three outstanding models of YOLO learning based on the recognition and classification of small-sized objects, similar images, and loss of information in object detection. The tasks to be evaluated include the classification and localization of objects. The proposed method of the research team is to use rice blight disease data to train the model to identify and separate each rice grain, then use the YOLO model to separate diseased and non-diseased rice seeds. Evaluating the effectiveness of each model will compare the efficiency of object detection and object classification of each different version of the dataset.

### A. Data Collection and Preprocessing

The data used for the study were provided by the Mekong Delta Rice Institute and were captured using a Samsung Galaxy Note 10 phone with a standard field size of 1816 × 4032. The rice image dataset used through selection includes 200 images. This image will create two data sets, Dataset Seed (DS) and Dataset Seed and Disease (DSD). Then, the image data will be processed through the following steps: 1) Label the data with each label (only rice seeds with DS; diseased rice seeds and good rice seeds with DSD); 2) push the project Data are labeled on Roboflow; 3) Split the dataset at a ratio of 6:2:2 into three sets of training, validation and testing sets; 4) Data argumentation to increase the number of images; 5) Export dataset according to the format of each version of YOLO used. The processed image is shown in Figs. 3 and 4.

The data is labeled in YOLO format using a self-designed program that provides bounding boxes and labels coordinates *x*, *y*, height, and width to train the YOLO model. However, we use the open-source LabelImg tool and the YOLO format for efficient and fast training.

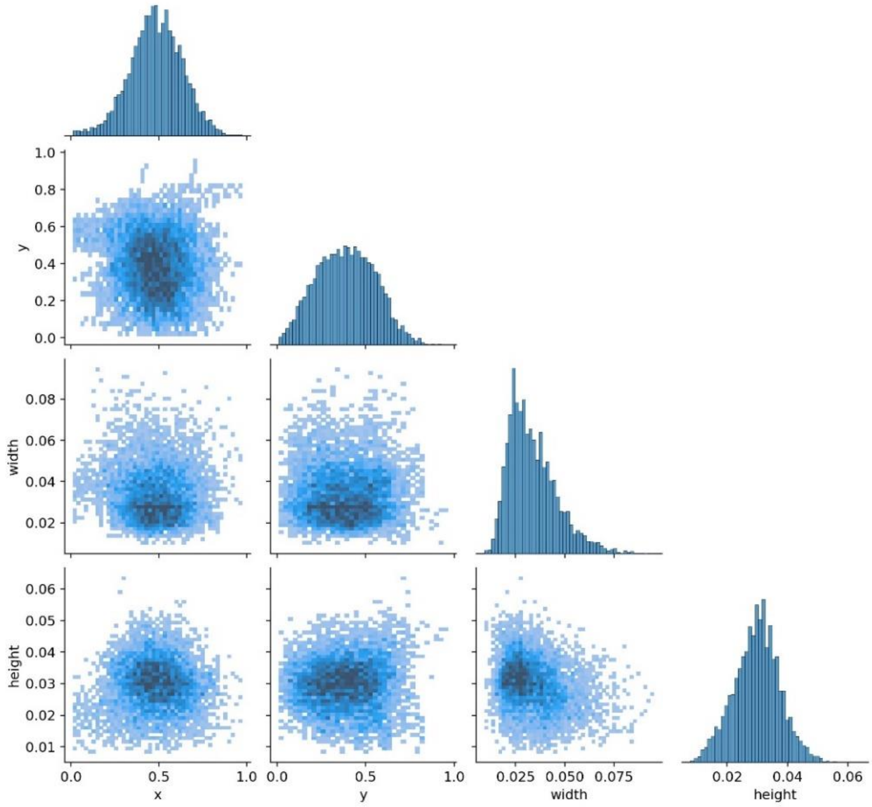



Figure 4. General features of the dataset.

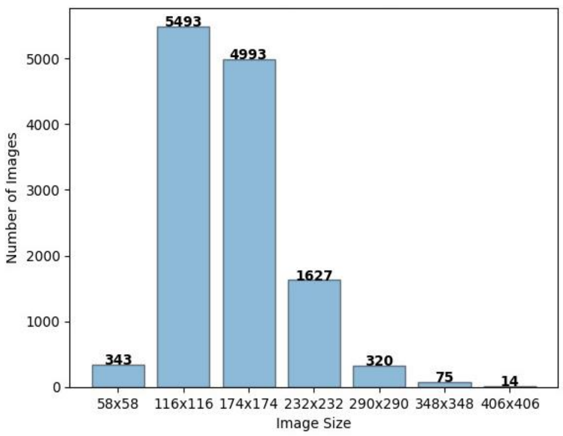



Figure 5. Statistics of rice grain image data by size.

The data parameters are visualized as the distribution of feature box occurrences and dimensions in the data set. In it, we can see that most of the data is uniformly distributed and, when normalized to a fairly standard normal distribution, has a bell shape. At the same time, the "width" and height graphs showing the objects are all similar in size and shape. The diagrams and illustrations are shown in Fig. 4. On the other hand, the data is enhanced as follows: the image will be randomly cropped to size 2×2, then transformed by Flip Horizontal argumentation operations, 90 deg Rotation, shear, Saturation with 50% enhanced image features in grain color and disease spots and highlights contoured objects. Since then, the data has been increased many times with 1,369 training images, 101 validation images, and 83 testing images (DS), and the total number of objects in each dataset is up to 25,734 objects and improving the efficiency of object detection. The processed image is illustrated in Fig. 3. Besides because the research objective will evaluate the problem of disease classification on rice seeds, the study labeled two types of seeds as 22,574 and diseased seeds as 3,169 (Called DSD) (see Table I).

Finally, the study's primary purpose is to conduct object recognition and classification on a small data set, so we analyze the number of rice grains used, as shown in Fig. 5. Statistically, the number of images of each particle varied in size from 58×58 pixels to 406×406 pixels, with the most excellent density distribution at sizes 116×116 and 174×174. Statistics show the complexity of the data made in the study.

TABLE I. STATISTICS OF RICE IMAGE DATA FOR EACH DATASET

| Dataset | DS | DSD | |
|---|---|---|---|
| Classes | Seed | Seed | Diseases |
| Total | 25,734 | 22,574 | 3,160 |

TABLE II. YOLOV5S ARCHITECTURE OF BACKBONE

| Backbone Layer | Output Layer |
|---|---|
| Input Shape | (640, 640, 3) |
| Conv 1 | (64, 6, 2, 2) |
| Conv 2 | (128, 3, 2) |
| C3 | (128) |
| Conv 3 | (256, 3, 2) |
| C3 | (256) |
| Conv 4 | (512, 3, 2) |
| C3 | (512) |
| Conv 5 | (1024, 3, 2) |
| C3 | (1024) |
| SPPF | (1024, 5) |

## B. Parameters for YOLO Models

In the YOLO architecture, the backbone is critical to help the model recognize and classify images well. Backbone plays the role of building high-level features from the features of the input image. These features are then fed into classification and prediction classes to produce the final result. The larger the backbone parameters, the more features are generated, helping the model learn more complex features. However, the use of extensive backbones also requires higher computational costs. Therefore, choosing the proper backbone is crucial to achieving the best performance with reasonable computational cost. In this study, the architectural parameters described specifically for each YOLO model are shown in Tables II–IV, and the architectural illustrations are shown in Figs. 6–8.

TABLE III. YOLOV6S ARCHITECTURE OF BACKBONE

| Backbone Layer | Setting Layer |
|---|---|
| EfficientRep | (1, 6, 12, 18, 6) (number layer repeats)<br>(64, 128, 256, 512, 1024) (Output channels)<br>For the output of each Conv |

TABLE IV. YOLOV7 ARCHITECTURE OF BACKBONE

| Backbone Layer | Output Layer |
|---|---|
| Conv 1 | (32, 3, 1) |
| Conv 2 | (64, 3, 2) |
| Conv 3 | (64, 3, 1) |
| Conv 4 | (128, 3, 2) |
| Conv 5-6 | (64, 1, 1) |
| Conv 7-8-9-10 | (64, 3, 1) |
| Conv 11 | (256, 1, 1) |
| Conv 12-13 | (128, 1, 1) |
| Conv 14 | (128, 3, 1) |
| Conv 15-16 | (128, 1, 1) |
| Conv 17-18-19-20 | (128, 3, 1) |
| Conv 21 | (512, 1, 1) |
| Conv 22-23 | (256, 1, 1) |
| Conv 24 | (256, 3, 1) |
| Conv 25-26 | (256, 1, 1) |
| Conv 27-28-29-30 | (256, 3, 1) |
| Conv 31 | (1024, 1, 1) |
| Conv 32-33 | (512, 1, 1) |
| Conv 34 | (512, 3, 1) |
| Conv 35-36 | (512, 1, 1) |
| Conv 37-38-39-40 | (512, 3, 1) |

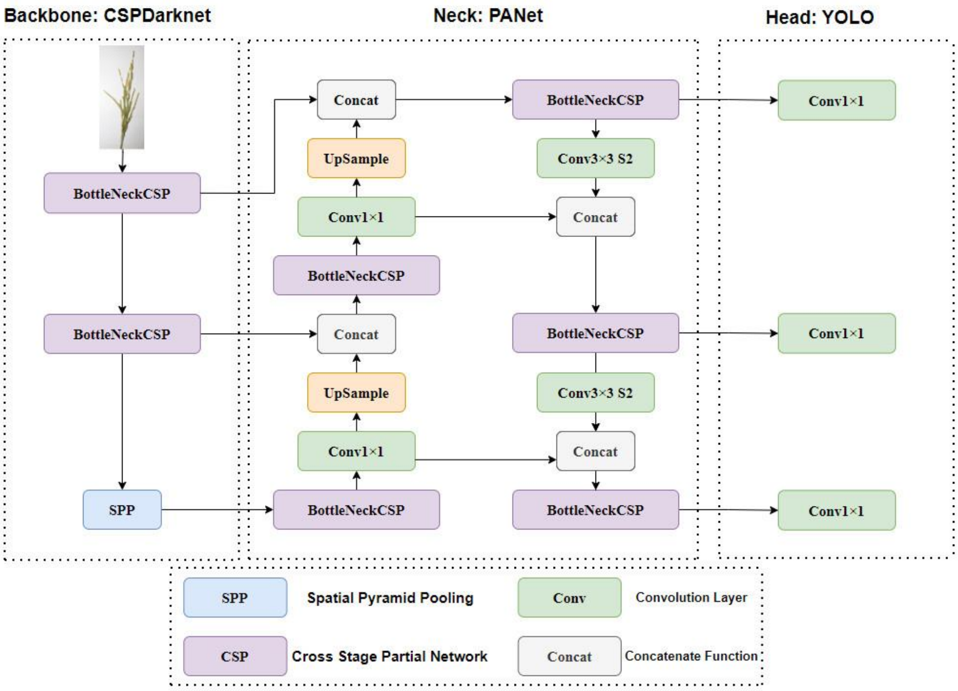



Figure 6. Network architecture for YOLOv5s.

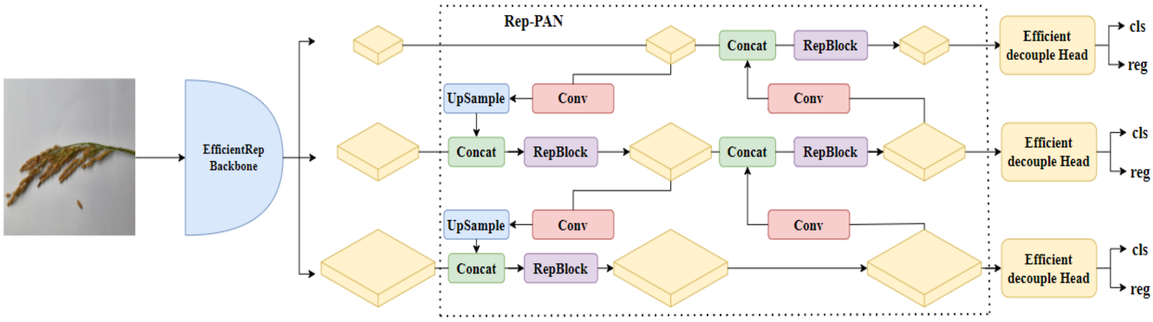



Figure 7. Network architecture for YOLOv6s.

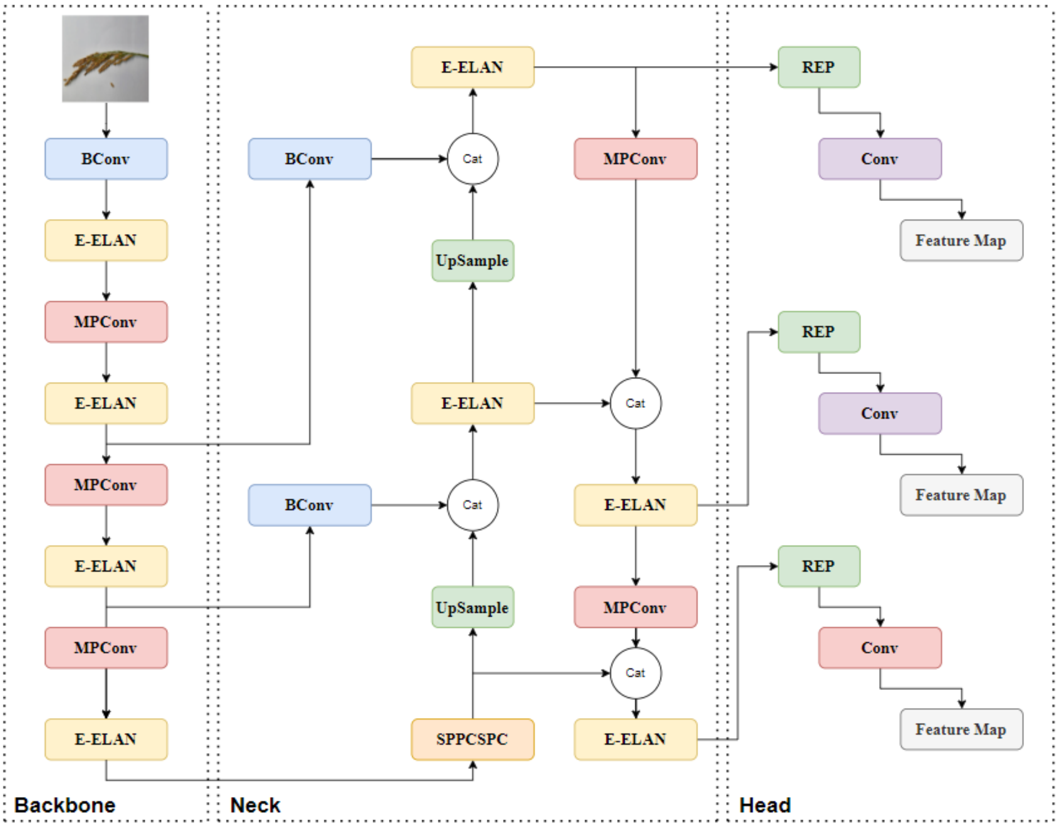



Figure 8. Network architecture for YOLOv7.

## C. *Experimental Environment*

The software and hardware parameters used for the model training in this document are shown in Table V.

TABLE V. CONFIGURATION PARAMETERS

| Device | Configuration |
|---|---|
| System | Windows 10 Home Single Language |
| CPU | Intel(R) Core (TM) i7-1065G7 |
| GPU | 16 GB GPU Tesla T4 |
| GPU accelerator | CUDA 11.2, Cudnn 11.0 |
| Frames | Pytorch |
| Compilers | Google. LLC. Collab and Anaconda |
| Python version | 3.8 |

## D. *Proposed Method*

After the data collection and processing are completed, they will be fed into YOLO models for object detection training. YOLO models are initialized to Pycharm Professional with the parameters mentioned in Table VI.

TABLE VI. INVARIANT TRAINING HYPARAMETERS

| Hyperparameter | Value |
|---|---|
| Batch size | 4 |
| Image size | 640×640 px |
| Epochs | 300 |
| Epochs evaluation interval | 1 |
| Optimization algorithms | SGD |
| Learning rate | 0.001 |
| Early Stopping | Enable |

The model parameters YOLOv5, YOLOv6, and YOLOv7 to be trained were selected with the small version of low complexity according to the information mentioned in Table VII. The models used for training include YOLOv5s, YOLOv6s, and YOLOv7, and these models will be set to the appropriate parameters provided for the custom dataset. The provided models will be trained on parameters from pre-trained models on the COCO dataset consisting of 80 classes to achieve better results by getting features from the parameters from the pre-trained dataset. The complexity and evaluation results of each model on the COCO set are listed in Table VII.

The process of our method is illustrated through steps including data processing, model training, evaluating the model's effectiveness on the test set, and comparing and evaluating the results of each model based on metrics according to Fig. 9.

TABLE VII. MODEL PROPERTIES

| Model | Params (M) | $mAP^{val}$ (%) |
|---|---|---|
| YOLOv5s | 7.2 | 56.8 |
| YOLOv6s | 17.2 | 60.4 |
| YOLOv7 | 36.9 | 69.7 |

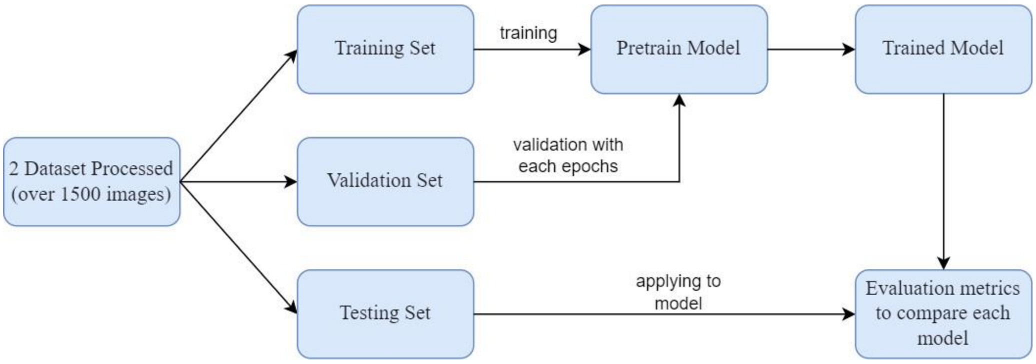



Figure 9. Illustrate the research process.

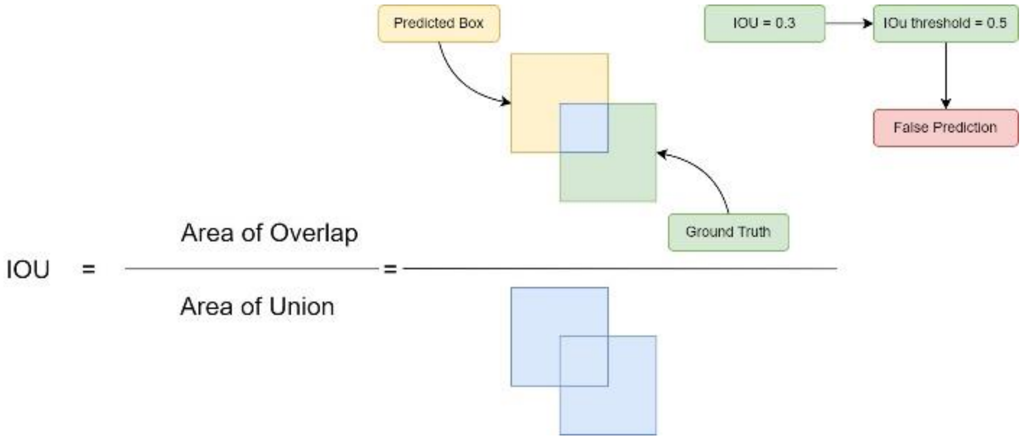



Figure 10. Illustrate the IoU formula and predictive evaluation based on the IoU Threshold.

## E. Evaluation Metrics

We use some basic metrics for Object Detection and Classification models to evaluate models and the effectiveness of recognition and classification on small objects. The primary scales are Average Precision (AP) and mean Average Precision (mAP) to evaluate the algorithm's ability to detect objects. They are calculated by comparing the algorithm's output with the actual object labels of the algorithm image and calculating the accuracy of the algorithm for each feature class according to Eqs. (1) and (2). In addition, Precision measures the ratio of the number of true predictions to the total number of optimistic predictions and recall measures the ratio of the number of true positives to the number of positive ground truths according to Eqs. (3) and (4). The F1 score considers both precision and recall by calculating their harmonic mean with Eq. (5). It helps in assessing the balance between precision and recall. Finally, we illustrate the change process of its value based on precision, recall and confidence score with F1-Score curve.

$$AP = \frac{1}{11}\sum_{R_i} PR_i \tag{1}$$

$$mAP = \frac{1}{classes}\sum_{i=1}^{classes} AP_i \tag{2}$$

$$P = \frac{TP}{TP + FP} \tag{3}$$

$$R = \frac{TP}{TP + FN} \tag{4}$$

$$F_1 - Score = \frac{(2 \times P \times R)}{(R + P)} \tag{5}$$

$$IoU(A,B) = (A \cap B)/(A \cup B);\ IoU(A,B) \in [0;1] \tag{6}$$

In addition, the above evaluations should be based on an Intersection of Union (IoU) threshold or ground truth ratio to confirm whether the recognized object is true or false. For this study, we used a ratio of 0.5 on the validation and testing set. For testing, we use more Confidence metrics with a value of 50% to identify all objects with a classification capacity greater than 50% to make an assessment. An illustration of the IoU and its identifiers is shown in Fig. 10 and Eq. (6).

## V. Result

We collected results and performed metrics on the testing set through training on two datasets. In which for only specific object recognition on DS, the YOLOv7 model achieved the best results in both training and testing, achieving more than 90% with Precision (91.13%), Recall (90.7%), and mAP threshold of 50% (91.23%). Compared to YOLOv5 and YOLOv6, version YOLOv7 gives stable results and continues to increase stability later. However, compared to version 6, it provides the most durable results in less than 200 epochs with mAP parameters on the validation set.

On the other hand, for simultaneous object recognition and classification in DSD, the version model of YOLOv5 and YOLOv7 achieved much lower results than YOLOv6, specifically for Recall and mAP (Yolov6 achieved 84.42% and 84.5%, respectively with 71.1% and 67.58% in YOLOv7). In the training environment, we added an early stopping mode to correct overfitting and limit the number of epochs. If the result becomes saturated (see Fig. 11), it stops at around 200 epochs due to no improvement too much efficiency improvement on the train set. However, we can see that the training result of YOLOv6 on the validation set is stable and much higher than that of the other two models. The parameter results and the training and testing process are specified in Fig. 12 and Table VIII.

TABLE VIII. Assessment Table of Metrics of YOLO Models on Test Set of 2 Dataset

| **Dataset** | DS | | | DSD | | |
|---|---|---|---|---|---|---|
| **Model** | YOLOv5s | YOLOv6s | YOLOv7 | YOLOv5s | YOLOv6s | YOLOv7 |
| **Precision (%)** | 87.5% | 89.08% | 91.13% | 83.56% | 85.7% | 91.2% |
| **Recall (%)** | 88.75% | 89.52% | 90.7% | 62.68% | 84.42% | 71.1% |
| **mAP (%)** | 87.2% | 89.11% | 91.23% | 65% | 84.5% | 67.58% |
| **Best epoch** | 198 | 168 | 278 | 204 | 154 | 224 |

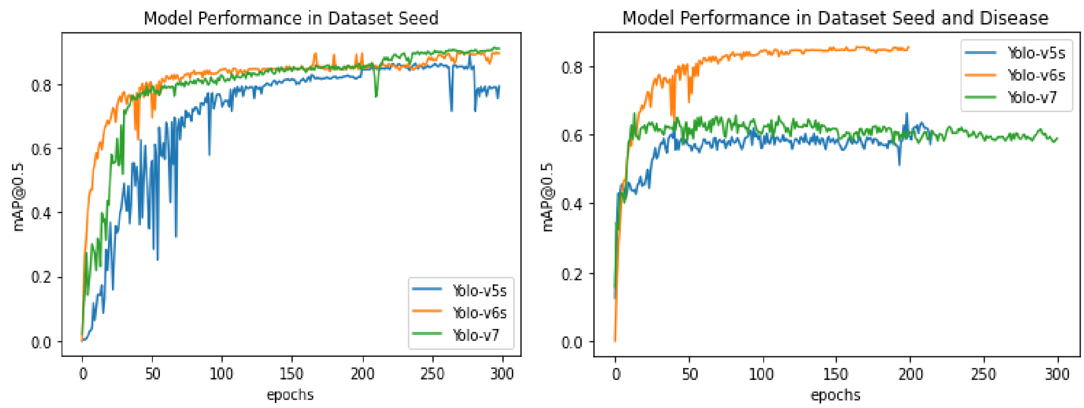



Figure 11. Training and validation results with mAP results per epoch.

In addition, in YOLOv5 and YOLOv7, with the F1-Score scale based on Confidence (metrics used to identify objects with a classification rate more significant than the Confidence index, Confidence has a value from 0 to 1. The line chart of YOLOv7 is better than YOLOv5. As in DS, the graph in YOLOv7 has a much higher shape than YOLOv5 and has an almost straight line at the top of the bell, indicating the F1-Score of the model in good action in almost Confidence from about 0.2 to 0.6, especially in YOLOv5, it shows a sharper peak of the bell. It achieves the best F1-Score (80%) at Confidence 0.459 compared to the version at Confidence 0.382 (with F1-Score reaching 86 %). In DSD, this dataset shows little difference between YOLOv5 and YOLOv7; with the class disease, we can see that version is more stable and has a flatter form than Yolov5; the estimated F1-Score of YOLOv7 is still higher (63% vs 61%). For YOLOv6, we use Confidence with a value of 0.25 and achieve the following results on two datasets: 89.05 % (DS) and 85.05 % (DDS). As a result, YOLOv6 showed quite good results, almost equal to YOLOv7 in DS and highest for DDS. An illustration of the F1-Score and Confidence on each dataset and model is shown in Fig. 12.

## VI. Discussion

Based on the training results and statistics on two tasks in three YOLO models for rice identification, YOLOv7 achieved quite good results. Moreover, for the published study, the YOLOv7 model showed more than 120% comparable results to other models of the same type. The model showed the ability to achieve outstanding results, with over 90% for different scales measured in the recognition task on the DS. The results are more stable than the v5 version. The precision achieved the highest rate with 91.2% in classification and recognition on the DSD, which shows the significantly improved recognition of the model structure and applied algorithms. For example, the ELAN block applied in the new structure of the model shows the efficiency of recognizing the features and the structure of the model (see Fig. 13).

On the other hand, during training and testing, we can see that YOLOv6 exhibits comprehensive capabilities on both data sets (see Table I). For DSD, the features of diseased and non-diseased rice grains are almost similar in size and shape, causing us challenges in classifying and recognizing images. Therefore, YOLOv5 and YOLOv7 do not achieve excellent results because the model structure needs to be simplified to extract essential features, which is overcome by using the EfficientRep Backbone network structure at the Backbone layer for feature extraction, as shown in Fig. 14. Hence, YOLOv6 achieves stable results over 85% in all testing parameters on two datasets. YOLOv6 is a reasonably stable model for identifying and classifying small objects with similar shape characteristics. The study only evaluates based on YOLO models with the simplest structure in that version that should still have limited guarantees.

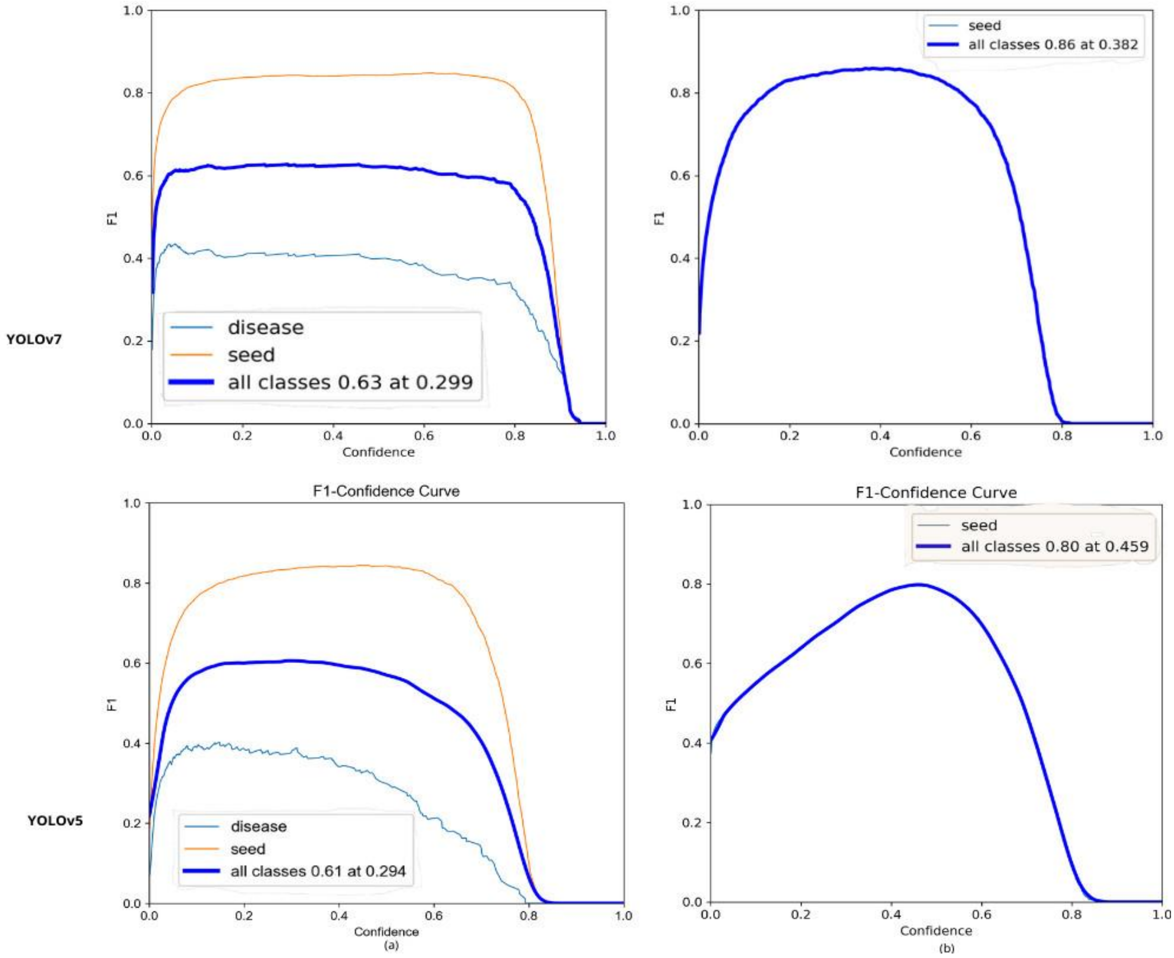



Figure 12. Illustrate the diagram of F1-Score according to the Confidence parameter on Yolov5 and Yolov7 in DSD(a) and DS(b).

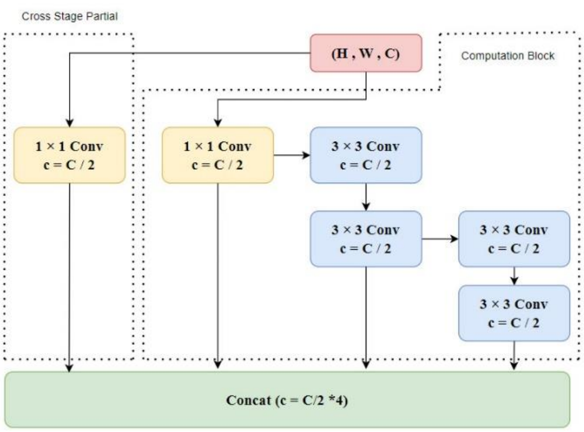



Figure 13. The architecture of ELAN block.

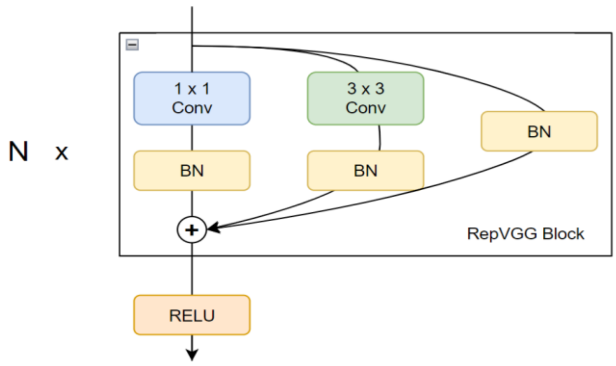



Figure 14. The architecture of RepVGG block.

Based on the advantages and disadvantages of each YOLO model in each structural component, the research contributes: 1) In-depth assessment of the architecture, each model will be suitable for different types of objects. 2) Provide data on diseases in rice seeds. This is the premise for developing more profound algorithms in classifying diseases of seeds.

## VII. Conclusion

In this study, we compared popular YOLO models for two main tasks in computer vision, object detection, and classification, on a dataset containing small objects with similar shape features. The results showed that YOLOv6 and YOLOv7 outperformed YOLOv5 in detecting and classifying objects, specifically diseased and non-diseased rice grains. YOLOv6 performed well and was not significantly affected when detecting and classifying complex objects due to its efficient image extraction structure from Efficient Rep Block. In future studies, improving and combining network structures that exhibit good performance in YOLOv6 and YOLOv7 can increase the model's reliability in detecting real-world objects.

## Conflict of Interest

The authors declare no conflict of interest.

## Author Contributions

Luyl-Da Quach and Khang Nguyen Quoc conducted the research to test the models; Anh Nguyen Quynh and Hoang Tran Ngoc conducted data processing; All conducted analysis of relevant data and suggested treatment methods during data processing as well as parameters during implementation; all authors approved the final version.

## Acknowledgment

To obtain the data set for research, we would like to thank the Mekong Delta Rice Institute for technical support and advice on rice diseases.

## References

[1] Y. Tang, B. Gu, and B. Jiang, "Research on feature enhancement for small object detection," *J. Phys.: Conf. Ser*., no. 1, 012036, 2006.
[2] P. Varma, "Enhancing rice productivity and food security: A study of the adoption of the System of Rice Intensification (SRI) in selected states of India," *Centre for Management in Agriculture Indian Institute of Management Ahmedabad*, vol. 74, no. 1, pp. 37–39, 2016.
[3] C. R. Rahman *et al.*, "Identification and recognition of rice diseases and pests using convolutional neural networks," *Biosystems Engineering*, vol. 194, pp. 112–120, Jun. 2020.
[4] S. Ghosal and K. Sarkar, "Riceleaf diseases classification using CNN with transfer learning," in *Proc. 2020 IEEE Calcutta Conference (CALCON)*, IEEE, 2020, pp. 230–236.
[5] G. Latif, S. E. Abdelhamid, R. E. Mallouhy, J. Alghazo, and Z. A. Kazimi, "Deep learning utilization in agriculture: Detection of rice plant diseases using an improved CNN model," *Plants*, vol. 11, no. 17, 2230, Aug. 2022.
[6] L.-D. Quach, A. N. Quynh, K. N. Quoc, and N. N. Thai, "Using optimization algorithm to improve the accuracy of the CNN model on the rice leaf disease dataset," in *Proc. Information Systems for Intelligent Systems*, Singapore: Springer Nature Singapore, 2023, pp. 535–544.
[7] L.-D. Quach, N. Q. Khang, A. N. Quynh, and T. N. Hoang, "Evaluation of the efficiency of the optimization algorithms for transfer learning on the rice leaf disease dataset," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 10, 2022.
[8] I. V. Arinichev, S. V. Polyanskikh, G. V. Volkova, and I. V. Arinicheva, "Rice fungal diseases recognition using modern computer vision techniques," *The International Journal of Fuzzy Logic and Intelligent Systems*, vol. 21, no. 1, pp. 1–11, Mar. 2021.
[9] Y. A. Nanehkaran, D. Zhang, J. Chen, Y. Tian, and N. A. Nabhan, "Recognition of plant leaf diseases based on computer vision," *J. Ambient Intell. Human Comput*, Sep. 2020.
[10] J. Panigrahi, P. Pattnaik, B. B. Dash, and S. R. Dash, "Ricequality prediction using computer vision," in *Proc. 2020 International Conference on Computer Science, Engineering and Applications (ICCSEA)*, IEEE, 2020, pp. 1–5.
[11] L.-D. Quach, P. N. Trong, K. N. Hoang, and N. N. Chi, "Combining contrast limited adaptive histogram equalization and Canny's algorithm for the problem of counting seeds on rice," in *Proc. Information Systems for Intelligent Systems*, Singapore: Springer Nature Singapore, 2023, pp. 567–577.
[12] H. Pallathadka *et al.*, "Application of machine learning techniques in rice leaf disease detection," *Materials Today: Proceedings*, vol. 51, pp. 2277–2280, 2022.
[13] K. Ahmed, T. R. Shahidi, S. M. I. Alam, and S. Momen, "Rice leaf disease detection using machine learning techniques," in *Proc. 2019 International Conference on Sustainable Technologies for Industry 4.0 (STI)*, IEEE, 2019, pp. 1–5.
[14] M. Sharma, C. J. Kumar, and A. Deka, "Early diagnosis of rice plant disease using machine learning techniques," *Archives of Phytopathology and Plant Protection*, vol. 55, no. 3, pp. 259–283, Feb. 2022.
[15] J. Chen, D. Zhang, Y. A. Nanehkaran, and D. Li, "Detection of rice plant diseases based on deep transfer learning," *J. Sci. Food Agric.*, vol. 100, no. 7, pp. 3246–3256, May 2020.
[16] S. S. Rautaray *et al.*, "Paddy crop disease prediction—A transfer learning technique," *International Journal of Recent Technology and Engineering*, vol. 8, no. 6, pp. 1490–1495, Mar. 2020.
[17] K. N. L. V. Prasad *et al.*, "Rice leaf diseases prediction using deep neural networks with transfer learning," *Environmental Research*, vol. 198, 111275, Jul. 2021.
[18] J. Chen, D. Zhang, M. Suzauddola, Y. A. Nanehkaran, and Y. Sun, "Identification of plant disease images via a squeeze-and-excitation MobileNet model and twice transfer learning," *IET Image Processing*, vol. 15, no. 5, pp. 1115–1127, Apr. 2021.
[19] K. Thenmozhi and U. Reddy, "Crop pest classification based on deep convolutional neural network and transfer learning," *Computers and Electronics in Agriculture*, vol. 164, 104906, Sep. 2019.
[20] R. Kumar *et al.*, "Diagnosis and detection of seed-borne fungal phytopathogens," *Seed-Borne Diseases of Agricultural Crops: Detection, Diagnosis and Management*, pp. 107–142, 2020.
[21] J. Zhang, Y. Yang, X. Feng, H. Xu, J. Chen, and Y. He, "Identification of bacterial blight resistant rice seeds using terahertz imaging and hyperspectral imaging combined with a convolutional neural network," *Front. Plant Sci.*, vol. 11, 821, Jun. 2020.
[22] I. Baek *et al.*, "Selection of optimal hyperspectral wavebands for detection of discolored, diseased rice seeds," *Applied Sciences*, vol. 9, no. 5, 1027, Mar. 2019.
[23] Y. Today *et al.*, "Training instance segmentation neural network with synthetic datasets for crop seed phenotyping," *Commun. Biol.*, vol. 3, no. 1, 173, Apr. 2020.
[24] N. Yang, Y. Qian, H. S. El-Mesery, R. Zhang, A. Wang, and J. Tang, "Rapid detection of rice disease using microscopy image identification based on the synergistic judgment of texture and shape features and decision tree–confusion matrix method," *J. Sci. Food Agric.*, vol. 99, no. 14, pp. 6589–6600, Nov. 2019.
[25] W. Zhan *et al.*, "An improved YOLOv5 real-time detection method for small objects captured by UAV," *Soft Comput.*, vol. 26, no. 1, pp. 361–373, Jan. 2022.
[26] B. Mahaur and K. K. Mishra, "Small-object detection based on YOLOv5 in autonomous driving systems," *Pattern Recognition Letters*, vol. 168, pp. 115–122, Apr. 2023.
[27] S. Li, Y. Li, Y. Li, M. Li, and X. Xu, "YOLO-FIRI: Improved YOLOv5 for infrared image object detection," *IEEE Access*, vol. 9, pp. 141861–141875, 2021.
[28] A. Benjumea, I. Teeti, F. Cuzzolin, and A. Bradley, "YOLO-Z: Improving small object detection in YOLOv5 for autonomous vehicles," arXiv Preprint, arXiv:2112.11798, 2021.

[29] Z. Liu, X. Gao, Y. Wan, J. Wang, and H. Lyu, "An improved YOLOv5 method for small object detection in UAV capture scenes," *IEEE Access*, vol. 11, pp. 14365–14374, 2023.
[30] H. Liu, F. Sun, J. Gu, and L. Deng, "SF- YOLOv5: A lightweight small object detection algorithm based on improved feature fusion mode," *Sensors*, vol. 22, no. 15, 5817, Aug. 2022.
[31] K. Xuan, L. Deng, Y. Xiao, P. Wang, and J. Li, "So-YOLOv5: Small object recognition algorithm for sea cucumber in complex seabed environment," *Fisheries Research*, vol. 264, 106710, Aug. 2023.
[32] J. Li, C. Liu, X. Lu, and B. Wu, "Cme-YOLOv5: An efficient object detection network for densely spaced fish and small targets," *Water*, vol. 14, no. 15, 2412, Aug. 2022.
[33] C. Li *et al*., "YOLOv6: A single-stage object detection framework for industrial applications," arXiv Preprint, arXiv:2209.02976, 2022.
[34] A. N. Azizah and C. Fatichah, "Tajweed-YOLO: Object detection method for Tajweed by applying HSV color model augmentation on Mushaf images," *J. RESTI (Rekayasa Sist. Teknol. Inf.)*, vol. 7, no. 2, pp. 236–245, Mar. 2023.
[35] K. Weng, X. Chu, X. Xu, J. Huang, and X. Wei, "EfficientRep: An efficient REPVGG-style convents with hardware-aware neural network design," arXiv Preprint, arXiv:2302.00386, 2023.
[36] A. John and D. Meva, Comparative study of various algorithms for vehicle detection and counting in traffic," *Advancements in Smart Computing and Information Security*, vol. 1760, pp. 271–286, 2022.
[37] C. Y. Wang, A. Bochkovskiy, and H. Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," arXiv Preprint, arXiv:2207.02696, 2022.
[38] H. Zhao, H. Zhang, and Y. Zhao, "YOLOv7-sea: Object detection of maritime UAV images based on improved YOLOv7," in *Proc. 2023 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW)*, pp. 233–238, 2023.
[39] L. Zhao and M. Zhu, "MS-YOLOv7: YOLOv7 based on multi-scale for object detection on UAV aerial photography," *Drones*, vol. 7, no. 3, 188, Mar. 2023.
[40] Y. Zhang, Y. Sun, Z. Wang, and Y. Jiang, "YOLOv7-rar for urban vehicle detection," *Sensors*, vol. 23, no. 4, 1801, Feb. 2023.
[41] F. Tang, F. Yang, and X. Tian, "Long-distance person detection based on YOLOv7," *Electronics*, vol. 12, no. 6, 1502, Mar. 2023.
[42] M. Horvat and G. Gledec, "A comparative study of YOLOv5 models performance for image localization and classification," in *Proc. the Central European Conference on Information and Intelligent*, 2022.
[43] N. Govil, and A. Sharma, "Estimation of cost and development effort in scrum-based software projects considering dimensional success factors," *Advances in Engineering Software*, vol. 172, 103209, 2022.
[44] S. G. Jindal and A. Kaur, "Information retrieval from software bug ontology exploiting formal concept analysis," *Computación y Sistemas*, vol. 24, no. 2, pp. 413–428, 2020.