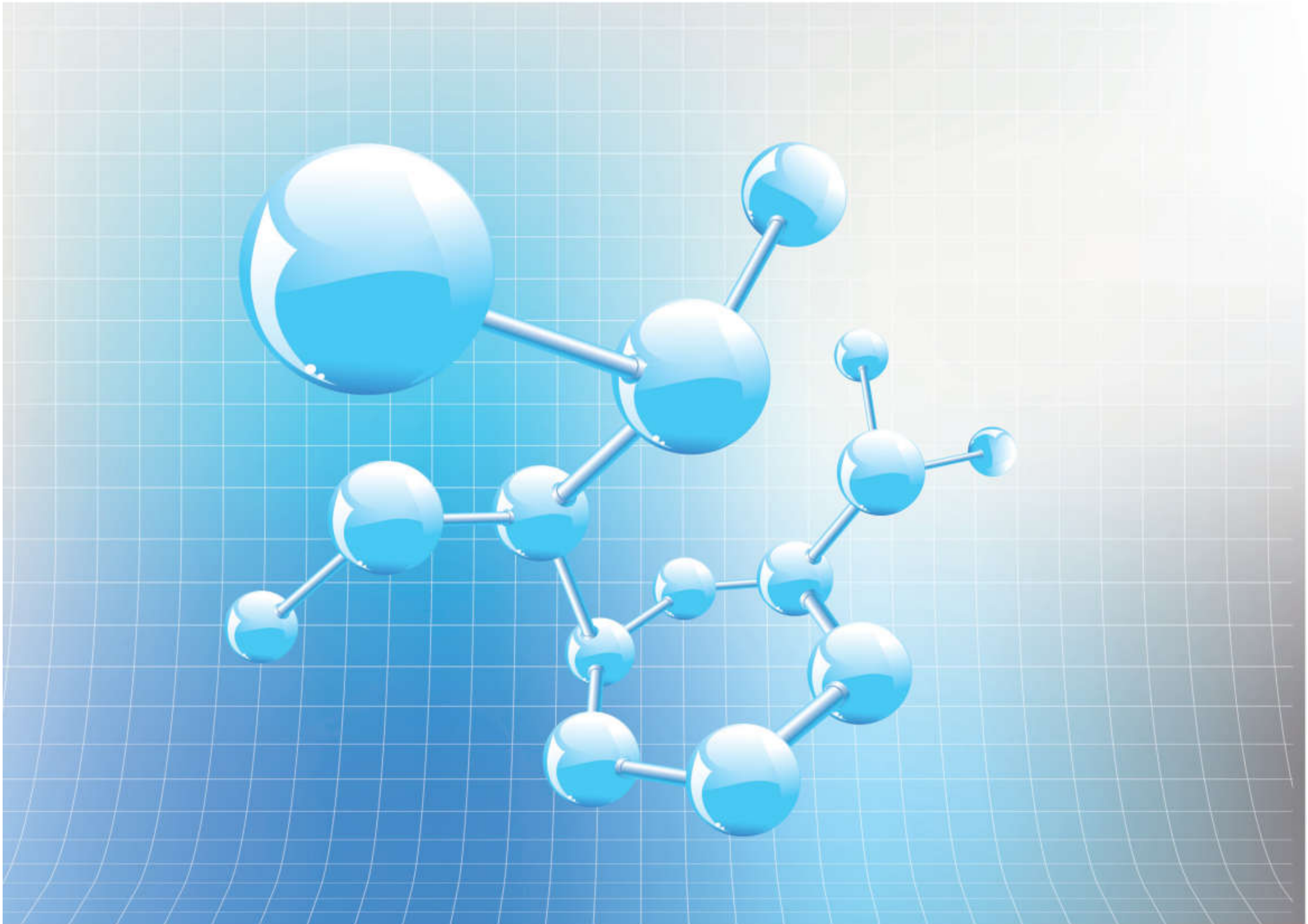
Tạp chí
NGHIÊN CỨU KHOA HỌC
ĐẠI HỌC SAO ĐỎ
SCIENTIFIC JOURNAL - SAO DO UNIVERSITY

P. ISSN 1859-4190
E. ISSN 2815-553X



Số 4 (87)
2024

P. ISSN 1859-4190
E. ISSN 2815-553X

Địa chỉ Tòa soạn:
Trường Đại học Sao Đỏ.
Số 76, Nguyễn Thị Duệ, Thái Học 2, phường Sao Đỏ, thành phố Chí Linh, tỉnh Hải Dương.
Điện thoại: (0220) 3587213, Fax: (0220) 3882 921, Hotline: 0912 107858/0936 847980.
Website: http://tapchikhcn.saodo.edu.vn/Email: tapchikhcn@saodo.edu.vn.

Giấy phép xuất bản số: 620/GP-BTTTT ngày 17/9/2021 của Bộ Thông tin và Truyền thông.
In 2.000 bản, khổ 21 × 29,7cm, tại Công ty TNHH in Tre Xanh, cấp ngày 17/02/2011.

TẠP CHÍ
NGHIÊN CỨU KHOA HỌC
ĐẠI HỌC SAO ĐỎ

TRONG SỐ NÀY

Số 4(87) 2024

LIÊN NGÀNH ĐIỆN - ĐIỆN TỬ - TỰ ĐỘNG HÓA

LIÊN NGÀNH CƠ KHÍ - ĐỘNG LỰC

TẠP CHÍ
NGHIÊN CỨU KHOA HỌC
ĐẠI HỌC SAO ĐỎ

TRONG SỐ NÀY

Số 4(87) 2024

NGÀNH KINH TẾ

NGÀNH TOÁN HỌC

NGÀNH HÓA HỌC - THỰC PHẨM

NGÀNH KHOA HỌC GIÁO DỤC

LIÊN NGÀNH TRIẾT HỌC - XÃ HỘI HỌC - CHÍNH TRỊ HỌC

SCIENTIFIC JOURNAL
SAO DO UNIVERSITY

CONTENTS

No 4(87) 2024

TITLE FOR ELECTRICITY - ELECTRONICS - AUTOMATION

TITLE FOR MECHANICAL AND DRIVING POWER ENGINEERING

SCIENTIFIC JOURNAL
SAO DO UNIVERSITY

CONTENTS

No 4(87) 2024

TITLE FOR ECONOMICS

TITLE FOR MATHEMATICS

TITLE FOR CHEMISTRY - FOOD

TITLE FOR EDUCATION SCIENCE

TITLE FOR PHILOSOPHY - SOCIOLOGY - POLITICAL SCIENCE

Deep learning-based Vietnamese rice leaf diseases detection using YOLOv10

Mô hình học sâu cho phát hiện bệnh trên cây lúa ở Việt Nam sử dụng YOLOv10

Han Hong Hanh[1], Can Vu Ha Son[1], Tran Van Kien[1], Do Le Tra My[1], Trinh Cong Dong[1], Vo Duc Nhan[3], Ngo Phuong Thuy[2], Bui Dang Thanh[1*]

**Corresponding author: thanh.buidang@hust.edu.vn*

[1]Hanoi University of Science and Technology

[2]Vietnam National University of Agriculture

[3]Ministry of Science and Technology

Date received: 28/9/2024
Date revised after review: 28/11/2024
Date accepted for publication: 29/11/2024

Abstract

Early detection of pests and diseases on rice plants is extremely essential to ensure growth, development and increase productivity, thereby promoting the development of rice production. A number of existing studies have used artificial intelligence for this purpose, famously including methods applying the YOLO model family and achieved significant results. In this study, we focus on YOLOv8 and YOLOv10. The study shows the effectiveness of the latest generation of YOLO, demonstrated in many aspects, especially the post-processing speed of each image improved by 8 times. Moreover, the accuracy of the new model also increased to 92.4% when tested on the database we collected and selected. A total of 3.558 images were used with 2.843 images for training set, 356 images for validation set and 359 images for testing set, covering common rice diseases such as blast, brown spot and leaf folder. Experimental results demonstrate that this approach has potential for practical application.

Keywords: *Rice leaf diseases; deep Learning; YOLOv8; YOLOv10.*

Tóm tắt

Việc phát hiện sớm các loại sâu bệnh trên cây lúa là vô cùng cần thiết để đảm bảo sự sinh trưởng, phát triển và nâng cao năng suất, từ đó thúc đẩy nền sản xuất lúa gạo phát triển. Đã có nhiều nghiên cứu sử dụng trí tuệ nhân tạo cho mục đích này, nổi tiếng trong đó có những phương pháp ứng dụng họ mô hình YOLO và đã đạt được những kết quả đáng kể. Trong nghiên cứu này, chúng tôi tập trung vào hai dòng YOLOv8 và YOLOv10. Nghiên cứu cho thấy sự hiệu quả của thế hệ YOLOv10, thể hiện ở nhiều khía cạnh, đặc biệt tốc độ hậu xử lý mỗi hình ảnh cải thiện đến 8 lần. Hơn thế, độ chính xác của mô hình mới cũng tăng lên đến 92,4% khi thử nghiệm trên bộ cơ sở dữ liệu được chúng tôi thu thập và chọn lọc. Tổng cộng 3.558 ảnh đã được sử dụng với 2.843 ảnh cho tập huấn luyện, 356 ảnh xác thực và 359 ảnh của tập kiểm tra, bao gồm các bệnh phổ biến trên cây lúa là đạo ôn, đốm nâu và cuốn lá. Kết quả thí nghiệm chỉ ra rằng hướng tiếp cận này có triển vọng ứng dụng trong thực tế.

Từ khóa: *Bệnh trên cây lúa; học sâu; YOLOv8; YOLOv10.*

1. INTRODUCTION

Rice is not only a staple food but it also forms the foundation of global food security, especially in Asia where it is the main source of nutrition for billions of people. The world's population is expected to continue growing, which will result in a significant increase in rice demand, making its cultivation critical for sustaining food supplies. However, rice crops are confronted with numerous challenges, with diseases being one of the most significant threats to production [1], [2].

Diseases like Leaf Blast, Leaf Folder, and Brown Spot can severely impact rice yields, leading to substantial economic losses and threatening the livelihoods of millions of farmers. The urgency of early and accurate disease detection in rice crops cannot be overstated, as these diseases can spread rapidly under favorable conditions, resulting in widespread damage that is often difficult to control once it has reached advanced stages [3].

Given the critical importance of rice to global food

Reviewers: 1. Prof.Dr. Le Van Hung
2. Dr. Do Van Dinh

security and the potential for these diseases to cause catastrophic losses, the development of effective detection and management strategies is of utmost importance. Traditional methods of disease detection in rice fields, such as visual inspection, are labor-intensive, time-consuming, and often inaccurate, especially when multiple diseases are present. This is where advanced technologies, particularly deep learning models, can play a transformative role.

Deep learning techniques, and specifically the YOLO (You Only Look Once) family of models, have revolutionized the field of computer vision, offering real-time, accurate detection of objects in images. Some scientific research using versions of this family have given the results. For example, YOLOv3 is used in tomato leaves [4]. YOLOv5 is used for rice leaf [5]. YOLOv8, for instance, has been successfully applied in agricultural settings, including the detection of plant diseases [6]. With the introduction of YOLOv10, expectations are high that it will offer further improvements in accuracy, speed, and overall performance.

This paper seeks to compare the performance of YOLOv8 and YOLOv10 in detecting and classifying rice diseases, focusing on the aforementioned Leaf Blast, Leaf Folder and Brown Spot. By evaluating these models on a comprehensive dataset of rice leaf images, we aim to determine which version is better suited for this critical task, thereby contributing to the development of more effective tools for safeguarding rice production against the ever-present threat of disease.

1.1. Leaf blast

Leaf Blast is particularly dangerous because of its ability to attack all parts of the rice plant, from the roots to the panicles. This fungal disease thrives in warm, humid environments-conditions often found in rice-growing regions and can cause up to 50% yield loss in severe cases [7]. The lesions it forms on leaves weaken the plant, reducing photosynthesis and leading to the premature death of the affected parts.

Figure 1. *Leaf blast*

1.2. Leaf folder

Leaf Folder infestations, though caused by a pest rather than a disease, are equally concerning. The larvae of this pest fold the leaves and feed inside, shielding themselves from natural predators and chemical treatments. The damage they cause reduces the photosynthetic area of the plant, leading to stunted growth and lower yields [8].

Figure 2. *Leaf folder*

1.3. Brown spot

Brown Spot has historically been a major problem in rice cultivation, especially in areas with poor soil fertility or adverse weather conditions. This fungal disease, which causes brown lesions on the leaves, can reduce yields by up to 90% if not properly managed [9]. Brown Spot is often associated with nutritional deficiencies, making it a particular threat in regions where soil health is already compromised.

Figure 3. *Brown spot*

2. PROPOSED METHOD

2.1. Overview

In this article, we focus on researching two deep learning object detection models YOLOv8 and YOLOv10, using the latest versions of the two models to detect three types of diseases of rice leaf: Blast, brown spot and leaf folder. After that, we will compare the research results of the two models and draw important conclusions.

2.2. Data preparation

2.2.1. Data collection

The dataset includes images of rice leaves in an infected state, collected directly in rice fields owned by Vietnam National University of Agriculture, from April to May 2024. Dataset includes images of three common diseases in Vietnam: Blast, brown spot, and leaf folder. Images were taken in ideal weather conditions: No rain, temperature 28°C÷30°C, warm sunshine, 70% humidity, wind speed below 10km/h, good lighting, suitable for data collection.

2.2.2. Data processing

The dataset includes 3558 images, divided into three parts: Training set includes 2843 images, validation set includes 356 images, test set includes 359 images. The number of samples used for each disease are 1782 for the blast, 1148 for the brown spot and 1225 for the leaf folder. Note that the images in the test set do not overlap with those from the training set and the validation set. The pictures are labeled on the Makesense.ai website, the results are saved as text files, compatible with the YOLO format.

Data augmentation methods are important in the training process. Different methods include flipping images horizontally and vertically, rotating images, translating images, etc. These are applied directly to the training process of each epoch, from which the model can learn more complex data and adapt to different conditions, leading to better performance without the need to create additional images. The table below shows the augmentation hyperparameters used in our model.

Table 1. *Hyperparameters*

No.	Name	Function
1	Lr0	0.01
2	Lrf	0.01
3	Hsv_h	0.015
4	Hsv_s	0.7
5	Hsv_v	0.4
6	Box	7.5
7	Cls	0.5
8	Dfl	1.5
9	Translate	0.1
10	Mosaic	1.0
11	Scale	0.5
12	Flipud	0.0
13	Fliplr	0.5

2.2.3. Training platform

A powerful computer system is indispensable to train a deep learning model. A model in deep learning can be trained using a CPU (Central Processing Unit), however, the time to complete the training process is unmeasurable, so this task must be performed by a system with a powerful GPU (Graphics Processor Unit) combined with huge memory space (RAM). Google Colab remains a great option to solve the above problem and is often used on computers with weak configurations, however, its hardware and time usage for users is limited.

In the project, we used an Acer laptop computer with the following specification:

Table 2. *System specs*

CPU	Intel® Core™ i7-12700H (up to 4.7 Ghz, 24MB cache)
GPU	NVIDIA GeForce RTX 3060 6GB GDDR6
RAM	16 GB DDR5 4800Mhz
Drive	SSD 512 GB

2.3. YOLOv8

2.3.1. Overview

YOLOv8, developed by Ultralytics, a powerful object detection model with many notable improvements over its predecessors. Some famous versions of YOLO are: YOLOv1 [10], YOLOv3 [11], YOLOv5 [12],… These models are already famous in the field of Computer Vision (CV) because they are both capable of providing relatively accurate predictions with trained layers and can learn new layers easily. YOLO models also have fast training speed, high accuracy and small model size, so they are more accessible to developers.

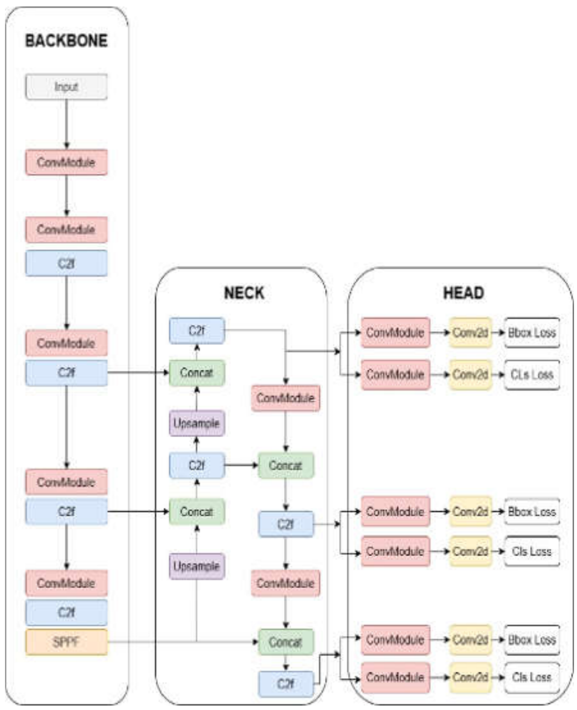


Figure 4. *YOLOv8 model*

In YOLOv8, we have different model sizes such as YOLOv8- n – nano, s – small, m – medium, l – large and x – extra large. The amount of feature extractions and convolution kernels make the difference among the versions.

YOLOv8 has architectural similarities with YOLOv5 due to the same developer, however, unique improvements make YOLOv8 superior and used in different applications. These include anchor-free detection, mosaic augmentation, a C2f module, a decoupled Head and a modified loss function. The general architecture of YOLOv8 is demonstrated below:

2.3.2. Backbone

In the Backbone, the C3 module in YOLOv5 is replaced in YOLOv8 by C2f, one of the two main blocks besides the Conv block. The C2f block is an aggregation of the C3 module from YOLOv5 and the ELAN model [13]. The C2f block consists of a Split block, several DarknetBottleneck blocks, a Concatenate layer and a ConvModule block. The place of the C2f block along with the scale of the model decides the number of DarknetBottleneck blocks. In C2f blocks, the number of bottleneck layers is for the first and the last one, for those in the middle. In addition, the feature maps of the second and third blocks are concatenated directly with the Head. The Conv block is fused from a Conv2d layer, a BatchNorm2d module, and a Sigmoid Linear Unit function.

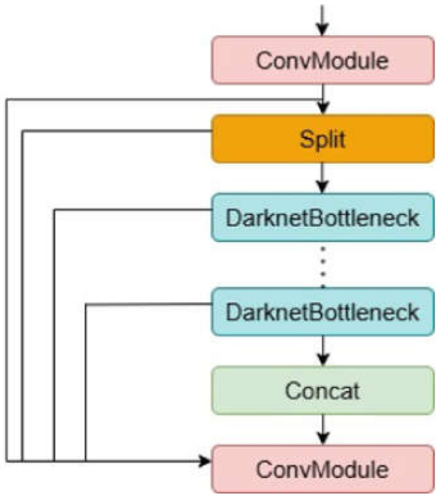


Figure 5. *C2f*

2.3.3. Neck

The main task of the Neck is to perform feature fusion, combining the features from the Backbone and forwarding them to the Head. YOLOv8 uses SPPF (Spatial Pyramid Pooling - Fast), an update to SPP, which allows efficient capturing and effective information encoding of an object in an image, regardless of its size and spatial location. The kernel size of the Max Pooling layer distinguishes SPPF from SPP. SPPF uses the same kernel size for each layer whereas SPP employs various kernel sizes to create feature maps of different sizes. Next, the max pooling layers in SPPF are placed in series, not in parallel like the predecessor. All of this makes SPPF significantly faster.

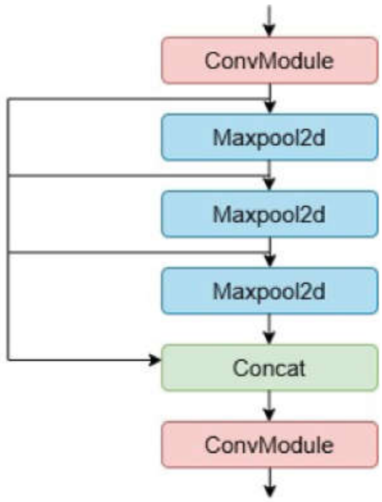


Figure 6. *SPPF*

2.3.4. Head

The Head includes several components such as ConvModule blocks, C2f blocks, Upsample functions and Concat blocks. A new feature of YOLOv8 compared to other versions is the Anchor-free detection, replacing the Anchor-based one. Next is the use of a decoupled Head. In previous models, object localization and classification tasks were handled on the same branch of the Head. This process often raises some task conflicts as the two tasks crave different features. The authors of YOLOv8 separated the two tasks and applied additional adjustments to get the best results.

2.3.5. Non - maximum suppression

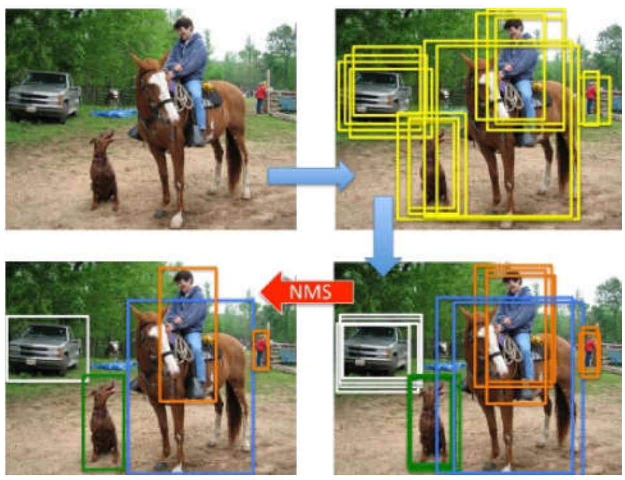


Figure 7. *Non - max suppression*

YOLOv8 uses non-maximum suppression (NMS). This is a technique used in object detection and computer vision to filter out duplicate or overlapping bounding box predictions. This technique is often applied after object detection algorithms generate multiple predicted bounding boxes for the same object [14].

The goal of non-max suppression is to eliminate duplicate detections and retain only the most accurate and representative bounding box for each object in an image. This technique reduces false positives and improves object detection accuracy.

2.4. YOLOv10

2.4.1. Overview

YOLOv10 is the latest version of the YOLO family, known for real-time end-to-end object detection.

The most notable development of YOLOv10 is related to the NMS technique. The drawback of this technique is the computational cost and inference time, especially when the number of boundings increases in the post-processing step, leading to inefficiencies in the architecture. This version of YOLO introduces consistent dual tasks for training without NMS and a comprehensive efficiency-based model design strategy.

2.4.2. Changes of YOLOv10 compared to YOLOv8

The architecture of YOLOv10 has a change in the Head, which is divided into One-to-many Head that creates many predictions for each object during training and One-to-one Head that creates the best prediction for each object during inference. As mentioned above, the biggest improvement of YOLOv10 is to eliminate the NMS post-processing step. The author proposes to use the Dual labels assignment and the Consistency matching metric [15].

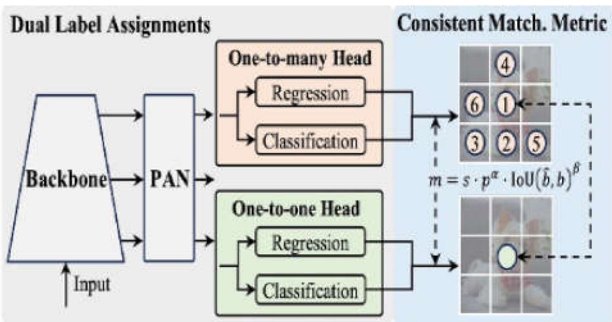


Figure 8. *YOLOv10 news*

Dual labels assignment takes advantage of the advantages of the two methods One-to-many label assignment and One-to-one label assignment, helping the model learn better, converge faster, simplify and speed up model deployment. In the training process, it will use 2 heads to take advantage of the strengths of the 2 methods, but when inferring, only 1 head, one-to-one, will be used to predict, avoiding the need to process NMS. Consistent math metrics is a metric to evaluate predictions and actual labels.

In addition, YOLOv10 also applies strategies such as Spatial-channel decoupled downsampling to minimize information loss and computational cost, Rank-guided block design to ensure optimal parameter utilization,…

The efficiency, accuracy, and light weight of YOLOv10 make it suitable for a wide range of applications and can replace previous YOLO models in most real-time detection applications. For this study, the model shows its effectiveness in agricultural applications, specifically in identifying diseases on crops.

3. RESULT AND DISCUSSION

A number of different metrics can be used to evaluate the effectiveness of a model. In this paper, we focus on collecting and comparing the results of YOLOv8 and YOLOv10 through the following metrics: *m*AP50, Parameters, GFLOPS and the duration of postprocess per image. Mean Average Precision (*m*AP) is used to measure the performance of computer vision models. *m*AP is equal to the average of the Average Precision metric across all classes in a model. *m*AP50 measures the mean average precision at an intersection over union (IoU) threshold of 0.5.

$$mAP = \frac{1}{C}\sum_{k=1}^{N} P(k)R(k) \tag{1}$$

C and N means is the number of classes and the number of IOU thresholds, k is the IOU threshold, P(k) is the precision, and R(k) is the recall.

With available technical and computational conditions, we selected two small-sized models, YOLOv8s and YOLOv10s, for this research. The input size of the images were all resized to 640 pixels × 640 pixels and the Adam optimizer was selected. Finally, we trained the model with a range of 300 epochs, the batch size was set to 16.

The results of both experiments on YOLOv8 and YOLOv10 can be seen in Table 2.

Table 2. *Result of the method*

		YOLOv8	YOLOv10
mAP50	Blast	91.8	92.4
	Brown spot	92.8	90.2
	Folder	90.0	92.1
	All	91.5	91.6
Parameters (M)		11,12	8,03
GFLOPS		28.4	24.5
Postprocess/image		1.6 ms	0.2 ms

As can be seen in the table, the results of two models are quite impressive with more than 91% accuracy. Compared with the algorithms of YOLOv5 and YOLOv8 in the previous applications, the expected numbers have been significantly increased. For example, the accuracy on blast has reached 91.8% and 92.4%

respectively, much better than the one giving 81.6% used in [5]. In another case, with the same generation of YOLOv8, our method shows better performance as compared to those of [6], for each type of disease and for all three diseases. The above results truly demonstrate the improvement when using state-of-the-art models combined with a carefully collected, fine-tuned dataset.

Figure 9. *Detection by YOLOv8*

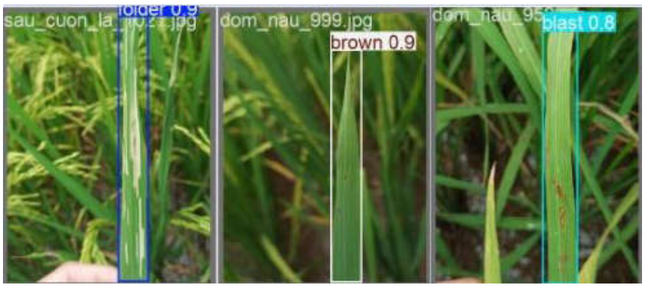


Figure 10. *Detection by YOLOv10*

In addition, focusing on YOLOv8 and YOLOv10 alone, we see that the newer version achieves similar results, even slightly better than its predecessor, while using 28% fewer parameters, faster computation with a lower number of flops. More specifically, the post-processing time per image is 8 times faster, providing great support during the model training process.

4. CONCLUSION

Our paper introduces a new approach to detect diseases on rice plants, using the latest YOLOv10 and, at the same time, upgrading YOLOv8's results using a new quality data set. Overall, the accuracy is up to 91.6%, which is a greater achievement as compared to 78.5% in [5] and 89.9% in [6]. YOLOv10 itself represents a robust architecture, demonstrated by its superior computational speed compared to older models. Next, the existing software will be combined with powerful hardware, creating economical, fast and effective solutions for farmers in crop management. All suggest an infinite potential for technological development. In the future, if we continue to enhance the model and adjust the data, we will produce more complete products, contributing more to smart agriculture in particular and artificial intelligence in general.

REFERENCES

[1]. Khush, G. S. (2005), *What it will take to feed 5.0 billion rice consumers in 2030*, *Plant Molecular Biology*, 59(1), 1-6.

[2]. Maclean, J. L., Dawe, D. C., Hardy, B., & Hettel, G. P. (2002), *Rice Almanac: Source Book for the Most Important Economic Activity on Earth*, IRRI, WARDA, CIAT.

[3]. Savary, S., Willocquet, L., Elazegui, F. A., Castilla, N., & Teng, P. S. (2000), *Rice pest constraints in tropical Asia: quantification of yield losses due to rice pests in a range of production situations, Plant Disease*, 84(3), 357-369.

[4]. Jun Liu and Xuewei Wang (2020), *Tomato Diseases and Pests Detection Based on Improved YOLOv3 Convolutional Neural Network*, Technical Advances in Plant Science, Vol.11, Article 898.

[5]. Trinh Cong Dong; Mac Tuan Anh; Giap Dang Khanh; Nguyen Thanh Huong; Nguyen Trong Cac; Bui Dang Thanh (2023), *Using deep learning for rice leaf diseases detection using YOLOv5*, Scientific Journal - Sao Do University 2023, 19 - 23.

[6]. Trinh, D.C.; Mac, A.T.; Dang, K.G.; Nguyen, H.T.; Nguyen, H.T.; Bui, T.D (2024), *Alpha-EIOU-YOLOv8: An Improved Algorithm for Rice Leaf Disease Detection*, AgriEngineering 2024, 6, 302 - 317, https://doi.org/10/3390/agriengineering6010018

[7]. Talbot, N. J. (2003), *On the trail of a cereal killer: investigating the biology of Magnaporthe grisea, Annual Review of Microbiology*, 57(1), 177-202.

[8]. Pathak, M. D., & Khan, Z. R. (1994), I*nsect Pests of Rice, International Rice Research Institute (IRRI).*

[9]. Ou, S. H. (1985), *Rice Diseases*, Commonwealth Mycological Institute.

[10]. Redmon, J.; Divvla, S; Girshick, R; Farhadi, A (2016), *You Only Look Once: Unified, Real-Time Object Detection*, *arXiv* 2016, arXiv:1506.02640.

[11]. Redmon, J.; Farhadi, A (2018), *Yolov3: An incremental improvement*, *arXiv* 2018, arXiv:1804.02767.

[12]. Ultralytics (2021), *YOLOv5: A state-of-the-art real-time object detection system*, https://docs.ultralytics.com/yolov5/ (updated on September 23, 2024).

[13]. Wang, C.Y.; Liao, H.Y.M.; Yeh, I.H (2022), *Designing network design strategies through gradient path analysis*, arXiv 2022, arXiv:2211.04800.

[14]. Gavin (2023), *One Important Concept You Must Know in Object Detection: Non-Maximum Supression(NMS)*, https://medium.com/@gavin_xyw/a-thorough-explanation-of-non-maximum-supression-nms-7bc8a997c729 (updated on July 10, 2023).

[15]. Petru Potrimba (2024), *What is YOLOv10? An Architecture Deep Dive...* Roboflow Blog: https://blog.roboflow.com/what-is-yolov10/ (updated on Jun 14, 2024).

THÔNG TIN TÁC GIẢ

Hàn Hồng Hạnh[1], Cấn Vũ Sơn Hà[1], Trần Văn Kiên[1], Đỗ Lê Trà My[1], Trịnh Công Đồng[1], Võ Đức Nhân[3], Ngô Phương Thủy[2], Bùi Đăng Thảnh[1*]

**Tác giả liên hệ: thanh.buidang@hust.edu.vn*

[1]Đại học Bách khoa Hà Nội;
[2]Học viện Nông nghiệp Việt Nam;
[3]Bộ Khoa học và Công nghệ.