QUANTIZED ALPHA–WIOU YOLOV5 ALGORITHM FOR RICE LEAF DISEASE DETECTION

THUẬT TOÁN ALPHA – WIOU YOLOV5 LƯỢNG TỬ HÓA PHÁT HIỆN BỆNH TRÊN CÂY LÚA

Nguyen Thu Ha*, Vu Duc Long, Nguyen Hung Cuong, Mai Ngoc Tam, Bui Dang Thanh

School of Electrical and Electronic Engineering, Hanoi University of Science and Technology, Vietnam

*Corresponding author: ha.nguyenthu3@hust.edu.vn

(Received: April 25, 2025; Revised: August 09, 2025; Accepted: August 11, 2025)

DOI: 10.31130/ud-jst.2025.23(9A).235

Abstract - This paper presents an innovative method based on convolutional neural networks for early detection and diagnosis of diseases in rice plants. The YOLOv5n model is modified by replacing its original loss function with the alpha-WIoU function to enhance feature learning and increase accuracy. The model is then quantized to reduce its size and optimize it for low-end devices. Our improved model achieves an accuracy of 93%, which is 2.6% higher than the original YOLOv5n, while reducing the model size by 45%. Additionally, it processes each image in just 0.554 seconds on a CPU, nearly three times faster than the original model. The results show that the proposed model is more accurate, lighter and faster, suitable for practical implementation, contributing to the development of smart agriculture.

Key words - Rice leaf diseases; YOLOv5; CIoU; Alpha–WIoU; Quantization

Tóm tắt - Bài báo trình bày về đề xuất một phương pháp cải tiến dựa trên mạng nơ-ron tích chập nhằm nhận diện và chẩn đoán sớm bệnh trên cây lúa. Mô hình YOLOv5n được điều chỉnh bằng cách thay thế hàm tính sai số gốc bằng hàm alpha-WIoU, giúp cải thiện khả năng học đặc trưng và tăng độ chính xác. Sau đó, mô hình được lượng tử hóa để giảm kích thước và tối ưu cho thiết bị có cấu hình thấp. Mô hình cải tiến đạt độ chính xác 93%, tăng 2,6% so với mô hình YOLOv5n ban đầu, trong khi kích thước mô hình giảm 45%. Ngoài ra, thời gian xử lý mỗi ảnh là 0,554 giây trên CPU, nhanh gấp gần ba lần so với mô hình gốc. Kết quả cho thấy mô hình đề xuất chính xác hơn, nhẹ và nhanh hơn, phù hợp triển khai thực tế góp phần phát triển nông nghiệp thông minh.

Từ khóa - Bệnh trên cây lúa; YOLOv5; CIoU; Alpha–WIoU; Lượng tử hóa mô hình

1. Introduction

Rice is the staple food for approximately half of the global population, with an estimated 480 million tons of milled rice produced annually for global food supply, especially in Asia [1]. In South Asian countries, rice plays a pivotal role in the cultural life of residents as well as in the national economy [2]. For Vietnam, the second largest rice exporter worldwide (4.8 tons of milled rice were exported, earning the national budget 2.9 billion USD in 2008, for example [3], rice not only ensures national food security but also imposes a profound impact on Vietnamese economic sector: 30% of agricultural products come from rice and 70% of the national workforce is allocated to this sector [4]. However, foliar diseases in rice crops exert a substantial detrimental effect on overall yields and agronomic quality, posing a critical threat to agricultural productivity. Therefore, early prediction and treatment of diseases on rice plants is an urgent requirement to maintain crop quality.

Currently, thanks to the tremendous innovation of science and technology in the era of the fourth industrial revolution, information technology and smart electronic devices are widely applied in the agricultural sector to optimize productivity, accuracy and minimize inputs such as: labor, fertilizers, and so on, leading to sustainable development [5]. Recently, deep learning models, especially convolutional neural network (CNN), are utilized for the early identification and classification of diseases in crops, optimizing and automating agricultural disease pre-treatment. For example, according to [6], YOLOv5, utilized for detecting five types of rice leaf disease, achieved 90%, 67%, 81%, and 76% for precision, recall, F1-score, and mAP@50, respectively. Following a previous study of our lab on autonomous rice leaf diseases detection, D. C. Trinh et al. [7] modified the original YOLOv8 by replacing Box Loss function from CIoU to alpha-EIoU, upgrading the performance of YOLOv8. According to another old research of our lab on this topic [8], YOLOv8 was enhanced by the utilization of Box Loss Function WIoU version 3 instead of the original CIoU. Consequently, the mAP@50 of the augmented YOLOv8 reached 89.2% which experienced a 4.5% increase compared to the original YOLOv8. For embedded system implementation, T. Zheng et al. [9] not only replaced Backbone of YOLOv5 with MobileNet version 3 but also applied a quantization technique to decrease the model size of YOLOv5 with minimal accuracy drop. In this research, although the model size dropped nearly 9 times, the accuracy of the model still reached 92.1%.

This paper proposes a modified YOLOv5n whose Box Loss function CIoU is replaced to alpha – WIoU; and our quantization in combination with precision calibration technique is applied for model size reduction with minimal accuracy drop and computational time, especially on embedded computers. The performance of our updated model will be evaluated based on the metrics evaluation values when detecting three types of diseases (Leaf folder, Leaf blast, Brown spot) of the dataset. The overall performance and inference time per image of this model will be evaluated and benchmarked against the original YOLOv5n and other state-of-the-art models from YOLO series when running on small-scale CPU Raspberry Pi 4 Model B (published in June 2019 by Raspberry Pi Holdings plc from UK).

2. Proposed method

2.1. Data preparation

2.1.1. Data collection

Our dataset comprises a total of 3,831 images, of which 1,831 were collected over a period of two months from rice fields of Vietnam National University of Agriculture (VNUA); while the remaining were from open-source database on the Internet. Three common rice foliar diseases such as: Leaf folder, Leaf blast, and Brown spot, were labeled as 0, 1, and 2, respectively. To ensure the consistency and accuracy of the annotation, expert consultation was sought from Mrs. Thu Hong (VNUA).

Figure 1. *An image of Leaf blast disease*

2.1.2. Data splitting

After the annotation process, our dataset was carefully reviewed to eliminate low-quality images, ensuring optimal conditions for the learning process. To maintain data balance and achieve the best performance, the dataset was subsequently partitioned into three independent subsets: a training set, a validation set, and a test set, with an 80% - 10% - 10% ratio, as outlined in [10]. Specifically, there are 3,174 images in the training set, 357 images in the validation set, and 300 images in the test set. Each subset contains three disease categories represented in each set.

2.1.3. Data augmentation

Data augmentation is an approach applied to raise the quantity of visual data through digital image processing algorithms without adding external images. Therefore, the model acquires a broader scope of information compared to the original dataset, upgrading the robustness and accuracy of deep learning model. In this study, we utilize the default hyper-parameters recommended by the developers, as presented inTable 1.

Table 1. *Hyper-parameters setting for data augmentation*

Lr0	0.01
Lr1	0.01
Hsv_h	0.015
Hsv_s	0.7
Hsv_v	0.4
Box	7.5
Cls	0.5
Dfl	1.5
Translate	0.1
Mosaic	1.0
Scale	0.5
Flipud	0.0
Fliplr	0.5

2.2. YOLOv5

2.2.1. Overview

YOLOv5 was officially released by Ultralytics in June 2020 with remarkable enhancement over its predecessors such as YOLOv3 and YOLOv4. There are five versions of YOLOv5 with increasing size: nano (n); small (s), medium (m); large (l); extra-large (x). Despite the release of YOLOv11 in 2024, YOLOv5n remains our preference due to its minimal size (1.9 million parameters) and fastest computational speed (4.1 GFLOPS) compared to other YOLO series [11].

2.2.2. Backbone

In most CNN architecture models, Backbone is the first and most important component and is responsible for feature extraction. Backbone of YOLOv5 is a combination of DarkNet53 architecture with Cross stage partial network (CSPNet), forming CSPDarkNet53. Specifically, as illustrated in Figure 2, the CBS block, a building block of the Backbone, is responsible for feature extraction. The C3 block is created based on the CSPNet architecture, where the feature map is partly forwarded to the next CBS block and the remaining part is directed to the Neck. Consequently, compared to YOLOv4, YOLOv5 achieves superior accuracy and computational speed [11].

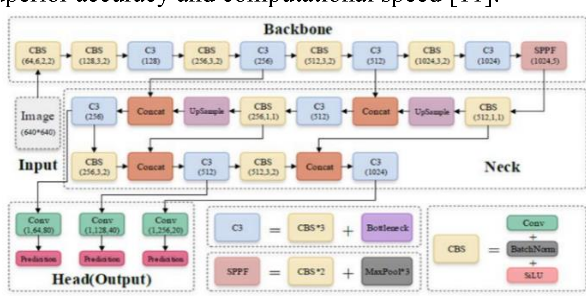

Figure 2. *YOLOv5 general architecture*

2.2.3. Neck

The Neck of YOLOv5 is a combination of the PANet architecture with Spatial pyramid pooling-fast (SPPF). This architecture not only serves as a bridge between Backbone and Head but also enhances the model's ability to analyze and extract unique features of feature maps from various sizes and flows from both deep and shallow layers, described in Figure 3 [11].

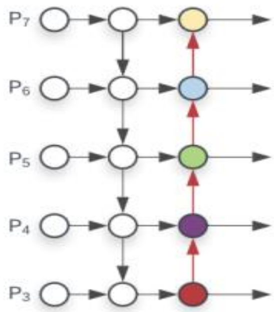

Figure 3. *Partial aggregation network*

2.2.4. Head

In YOLOv5 architecture, Head is the final component of the network responsible for producing the predictions after the feature extraction process in the Backbone and Neck. This component takes the processed feature maps and converts them into final outputs. This step is critical because it directly influences the accuracy and efficiency of the final prediction.

2.3. Loss function

2.3.1. Overview

In the YOLOv5 model, the total loss function called Classification loss is composed of three main subfunctions: Class loss $\mathcal{L}_{cls}$ (BCE with logits loss), Object loss $\mathcal{L}_{dfl}$ (BCE with logits loss), and Box loss $\mathcal{L}_{box}$ (CIoU loss). The relationship between these functions is expressed in:

$$Loss = l_1\mathcal{L}_{cls} + l_2\mathcal{L}_{box} + l_3\mathcal{L}_{dfl} \quad (1)$$

In this research, we replaced the box loss function from CIoU to our function called alpha-WIoU.

2.3.2. Complete IoU

Complete IoU loss function, abbreviated CIoU, is a box loss function updated from older models of YOLO series such as GIoU, DIoU for providing more precise bounding box regression and achieving better performance in object detection problems. The formula for CIoU is described specifically below:

$$\mathcal{L}_{CIoU} = 1 - IoU + \frac{p^2(b, b^{gt})}{c^2} + \alpha v \quad (2)$$

The operator p indicates the Euclidean distance, while α and v are operators applied to calculate the discrepancy in the width-to-height ratio (w as width, and h as height), and described in:

$$\alpha = \frac{v}{(1 - IoU) + v} \quad (3)$$

$$v = \frac{4}{\pi^2}(\arctan\frac{w^{gt}}{h^{gt}} + \arctan\frac{w}{h})^2 \quad (4)$$

2.3.3. Wise - IoU

In fact, some datasets such as leaf diseases inevitably contain a huge number of low-quality images which are compatibly learnt by the model, so WIoU, a dynamic non-monotonic focusing mechanism loss function belonging to IoU-based loss family, is introduced as an ideal replacement for CIoU. This box loss function was designed to eliminate the penalty of geometric factors when predicting box overlaps the targeting one, enhancing the generalization of the model during training process. WIoU has three versions, presented in Equation (5), and in this paper, we modify the third and state-of-the-art version of WIoU loss function ($\mathcal{L}_{WIoUv3}$) because of remarkable results of our previous study.

$$\mathcal{L}_{WIoUv3} = d\mathcal{L}_{WIoUv1} \quad (5)$$

Operator d is defined in:

$$d = \frac{\beta}{\delta - \alpha^{\beta-\delta}} \quad (6)$$

In Equation (6), α and δ are hyper-parameters and are chosen as 1.9 and 3, respectively following [12]; while β is represented in:

$$\beta = \frac{\mathcal{L}^*_{\text{IoU}}}{\overline{\mathcal{L}_{\text{IoU}}}} \quad (7)$$

$\mathcal{L}_{WIoUv1}$ is WIoU loss function version 1 which is defined below:

$$\mathcal{L}_{WIoUv1} = \mathcal{R}_{WIoU}\mathcal{L}_{IoU} \quad (8)$$

In Equation (8), $\mathcal{L}_{WIoUv1}$ incorporates two layers of attention $\mathcal{R}_{WIoU}$ and $\mathcal{L}_{IoU}$. $\mathcal{R}_{WIoU}$, presented in Equation (9), is a distance attention and effective to remove factor challenging convergence, while $\mathcal{L}_{IoU}$ is intersection over unit and is defined as (1 – IoU).

$$\mathcal{R}_{WIoU} = \exp(\frac{(x - x_{gt})^2 + (y - y_{gt})^2}{{W_g}^2 + {H_g}^2}) \quad (9)$$

2.3.4. Alpha – Wise IoU

Alpha – IoU loss is a new family of box loss functions based on the fusion of IoU loss function and power parameter alpha. The advantage of this type of box loss function is: when alpha > 1, the reweighting factor increases monotonically with the increase in IoU. In other words, with alpha > 1, the model will focus more on one high IoU object [13]. Consequently, our modified box loss function alpha-WIoU is presented in:

$$\mathcal{L}_{alpha-WIoU} = d\mathcal{L}_{alpha-WIoUv1} \quad (10)$$

$\mathcal{L}_{alpha-WIoUv1}$ is an updated of $\mathcal{L}_{WIoUv1}$ with the combination of $\mathcal{R}_{alpha-WIoU}$ and $\mathcal{L}_{alpha-IoU}$ which is a replacement of $\mathcal{R}_{WIoU}$ and $\mathcal{L}_{IoU}$ as presented in:

$$\mathcal{L}_{alpha-IoU} = 1 - IoU^{\propto} \quad (11)$$

$$\mathcal{R}_{alpha-WIoU} = \exp(\frac{(x - xgt)^{2\propto} + (y - y_{gt})^{2\propto}}{Wg^{2\propto} + Hg^{2\propto}}) \quad (12)$$

$\propto$ is chosen as 3 for the recommendation of [7].

2.4. Post training quantization

Quantization is a technique that converts learning parameters from a high-precision 32-bit floating-point (FP32) to an 8-bit integer (int8) format, reducing model size and computational time. Because the quantization technique is applied after the training process, it is called Post training quantization (PTQ). The trained YOLOv5 model is typically saved in Pytorch format; however, this format requires significant computational resources, so it is not suitable for running on CPU. To enable the model to run efficiently on a CPU with a smaller size, YOLOv5 is stored in TensorFlow Lite format (TFLite), which is designed for microcontrollers or CPUs deployment and

PTQ is applied during this process. Although PTQ can cause a huge accuracy drop, we use a small sample from the training set as a calibration dataset for the PTQ process to minimize the drop [14].

3. Results and Discussion

3.1. Hyper-parameter settings

Hyper-parameters are typically predefined by users prior to the training process based on experimental experience. In this study, we choose the Stochastic Gradient Descent as our optimization algorithm with default hyper-parameters recommended by Ultralytics, as outlined in Table 2, with 100 training epochs.

Table 2. *Training hyper-parameters setting*

Name	Value
Batch size	16
Image size	640
Optimizer	SGD
Initial learning rate	0.01
Final learning rate	0.0001
Momentum	0.937
Training epoch	100

3.2. Experimental results

We utilize common metrics for object detection evaluation, including Precision, Recall, F1 – score, and mean average precision (mAP). The training and evaluation process was conducted on our computational systems of the laboratory with corresponding details presented in Table 3.

Table 3. *Technical details of computational system*

CPU	Intel® Core™ i7-12700H
GPU	NVIDIA GeForce RTX 3060 6GB GDDR6
RAM	16 GB DDR5 4800Mhz
DRIVE	SSD 512 GB

Table 4 presents a comparative analysis between the original YOLOv5n model and its variants incorporating different loss function change: YOLOv5n with WIoU (version 3), YOLOv5n with alpha-WIoU, and quantized YOLOv5n with alpha-WIoU. Based onTable 4, although the recall metric of the modified model decreased by 0.6% compared to the original one, other metrics demonstrated significant improvements: Precision increased by 2.6%, F1-score improved by 1%, and mAP@50 rose by 1.5%. The performance of our augmented YOLOv5n was also compared to other variants from YOLO series in order to demonstrate its superior performance despite its compact size, which is detailed specifically in Table 5.

Table 4. *Comparison of different modified models*

	P	R	F1	mAP@50
CIoU	90.4	91.1	90.8	92.5
WIoU	90.6	89.8	90.2	93.3
Alpha – WIoU	92.3	91.3	91.8	94
Proposed	**93**	90.5	**91.8**	**94**

Table 5. *Comparison of the proposed model with other YOLO models*

	P	R	F1	mAP@50	Size
YOLOv5n	90.4	91.1	90.8	92.5	3.5 MB
YOLOv8n	91.3	91.6	91.5	92.6	5.95 MB
YOLOv10n	91.4	91.8	91.6	93.1	5.2 MB
YOLOv11n	91.5	91.9	91.7	93.2	5.5 MB
Proposed	**93**	90.5	**91.8**	**94**	**1.9 MB**

After that, our quantized YOLOv5n, along with other models from YOLO series, were also tested on Raspberry Pi 4 Model B, equipped with 4GB RAM. As shown Table 6, our model achieves the lowest processing time, demonstrating its suitability in real-time application.

Table 6. *Experimental results on Raspberry Pi 4*

Model	Inference time
YOLOv5n	1.55 s
YOLOv8n	2.74 s
YOLOv10n	2.25 s
YOLOv11n	1.8 s
Proposed	**0.554 s**

The performance of the improved YOLOv5n on the entire test set is detailed in Table 7.

Table 7. *Overall performance of proposed model*

	P	R	F1	mAP@50
Leaf folder	92.1	91	92.6	95.2
Leaf blast	92.9	90.5	92	96.9
Brown spot	94	90	90.8	90
Overall	93	90.5	91.8	94

We visualized predictions in Figure 4 to demonstrate the detection ability and robustness of our model when applied to real images, captured in VNUA, outside our dataset. Each image highlights the bounding boxes around detected infectious areas, illustrating the capability of proposed model to precisely locate and detect the exact diseases.

Figure 4. *Output visualization*

4. Conclusion

In this research, we improved YOLOv5n for early-stage disease detection on rice leaf. By replacing the original Box Loss function to the alpha-WIoU which is an advanced formulation that integrates the strengths of previously published Alpha-IoU and WIoU Box Loss functions, and utilizing PTQ with precision calibration technique, our proposed model not only achieves enhancement in accuracy but also reduces computational resource requirements, making it feasible for deployment on small-scale CPUs instead of relying on expensive GPUs.

Compared to our previous work [8] which had mAP@50 reaching at 89.2%, the model of this research demonstrates overwhelming accuracy by 3.8% despite taking advantage of the small YOLOv5n instead of larger compact size model in YOLO series like YOLOv8n. In addition, despite compact size, our model exhibits superior performance in comparison to other up-to-date deep learning architectures among nano versions of YOLO series such as YOLOv8n, YOLOv10n, and YOLOv11n. Nevertheless, we recognize several limitations of the current model and aim to enhance it further in future research by expanding the dataset, providing standard deviation for evaluation metrics to stabilize the performance of our model, as well as optimizing the model for deployment on embedded microcontrollers, reducing costs and improving accessibility.

REFERENCES

[1] S. Muthayya, J. D. Sugimoto, S. Montgomery, and G. F. Maberly, "An overview of global rice production, supply, trade, and, consumption", *Annals of the New York academy of sciences*, vol. 1324, no. 1 pp. 7-14, 2014. https://doi: 10.1111/nya s.12540.

[2] E. Crystal and P. Whittlesey, "The role of rice in Southeast Asia", *Education about Asia,* vol. 9, no. 3, pp. 50-55, 2004.

[3] S. Shigetomi, K. Kubo, and K. Tsukada, "*The World Food Crisis and the Strategies of Asian Rice Exporters*", Institute of Developing Economies (IDE-JETRO), Chiba, Japan, Spot survey 32, 2011.

[4] K. Maitah, L. Smutka, J. Sahatqija, M. Maitah, and N. P. Anh, "Rice as a Determinant of Vietnamese", *sustainability*, vol. 12, p. 5123, 2020. https://doi.org /10.3390/su12125123.

[5] T. Popović, N. Latinović, A. Pešić, Z. Zečević, B. Krstajić, and S. Djukanović, "Architecting an IoT-enabled platform for precision agriculture and ecological monitoring: A case study", *Computers and Electronics in Agriculture*, vol. 140, pp. 255-265, 2017. https://doi.org/10.1016/j.compag.2017.0 008.

[6] M. E. Haque, A. Rahman, I. Junaeid, S. U. Hoque, and M. Paul, "Rice Leaf Disease Classification and Detection Using YOLOv5", *arXiv preprint arXiv:2209.01579*, 2022.

[7] D. C. Trinh, A. T. Mac, K. G. Dang, H. N. Thanh, H. N. Thai, and T. B. Dang, "Alpha-EIOU-YOLOv8: An Improved Algorithm for Rice Leaf Disease Detection", *AgriEngineering*, vol. 6, pp. 302-317, 2024. https://doi.org/10.3390/agrienginee ring6010018.

[8] C. D. Trinh, T. M. D. Le, T. H. Do, N. M. Bui, T. H. Nguyen, Q. U. Ngo, "Improving YOLOv8 Deep leaning model in rice disease detection by using Wise-IoU loss function", *Journal of Measurement, Control and Automation,* vol. 29. no. 1, pp. 1-6, 2025.

[9] T. Zeng, S. Li, Q. Song, F. Zhong, and X. Wei, "Lightweight tomato real-time detection method based on improved YOLO", *Computer and Electronics in Agriculture,* vol. 205, p. 107625, 2023. https://doi.org/10.1016/j.compag.2023.1076 25.

[10] I. O. Muraina, "Ideal dataset splitting ratios in machine learning algorithms: General concerns for data scientists and data analysts", *in 7th International Mardin Artuklu scientific researches conference*, 2021, pp. 496-504.

[11] R. Khanam and M. Hussain, "What is YOLOv5: A deep look into the internal features of the popular object detector", *arXiv preprint arXiv: 2407.20892*, 2024.

[12] Z. Tong, Y. Chen, Z. Xu, and R. Yu, "Wise-IoU: Bounding Box Regression Loss", *arXiv preprint arXiv:2301.10051*, 2023.

[13] J. He, S. Erfani, X. Ma, J. Bailey, Y. Chi, and X. S. Hua, "Alpha-IoU: A Family of Power Intersection over unit", *in 35th Conference on Neural Information Processing Systems,* 2021. https://doi.o rg/10.48550/arXiv.2110.13675.

[14] I. Hubara, Y. Nahshan, Y. Hanani, R. Banner, and D. Soudry, "Accurate Post Training Quantization With Small Calibration Sets", *in 38th International Conference on Machine Learning*, 2021.