Journal of Informatics and Web Engineering

Vol. 3 No. 1 (February 2024) eISSN: 2821-370X

Plant Disease Detection and Classification Using Deep Learning Methods: A Comparison Study

Pei-Wern Chin[1], Kok-Why Ng[2*] and Naveen Palanichamy[3]

[1,2,3]Multimedia University, Persiaran Multimedia, 63100 Cyberjaya, Selangor, Malaysia.

**corresponding author:(kwng@mmu.edu.my, ORCiD:0000-0003-4516-4634)*

Abstract - The presence issue of inaccurate plant disease detection persists under real field conditions and most deep learning (DL) techniques still struggle to achieve real-time performance. Hence, challenges in choosing a suitable deep-learning technique to tackle the problem should be addressed. Plant diseases have a detrimental effect on agricultural yield, hence early detection is crucial to prevent food insecurity. To identify and categorise the indications of plant diseases, numerous developed or modified DL architectures are utilised. This paper aims to observe the performance of the YOLOv8 model, which has better performance than its predecessors, on a small-scale plant disease dataset. This paper also aims to improve the accuracy and efficiency of plant disease detection and classification methods by proposing an optimised and lightweight YOLOv8 architecture model. It trains the YOLOv8 model on a public dataset and optimises the YOLOv8 algorithm with the integration of the GhostNet module into the backbone architecture to cut down the number of parameters for a faster computational algorithm. In addition, the architecture incorporates a Coordinate Attention (CA) mechanism module, which further enhances the accuracy of the proposed algorithm. Our results demonstrate that the combination of YOLOv8s with CA mechanism and transfer learning obtained the best result, yielding $mAP_{0.5}$ score of 72.2% which surpassed the studies that utilised the same dataset. Without transfer learning, our best result is demonstrated by YOLOv8s with GhostNet and CA mechanism yielding a $mAP_{0.5}$ score of 69.3%.

Keywords—YOLOv8, GhostNet, Deep Learning, Plant Disease Detection, Object Detection

Received: 14 June 2023; Accepted: 15 September 2023; Published: 16 February 2024

I. INTRODUCTION

Farming is significant to support life on Earth by supplying the food that humans need for survival and the country's economic benefit. The rise of the human population on Earth is challenging agriculture to unparalleled levels. According to Lovrenciear [1], Malaysia's agriculture is facing enormous challenges in trying to produce a stable food supply. Citizens will suffer from higher living expenses on a basic level due to soaring global food costs when an undependable and insufficient food crop is available. Thus, it is significant to improve and transform the agricultural sector in Malaysia which will be very advantageous for the country's well-being and economic growth.

The possibility of disease transmission and pest damage is one of the biggest hazards to the growth of crops. According to Mbinda et al. [2], most farmers struggle to properly address this catastrophe because they are unable to accurately assess and identify the damage done to their crops. The productivity and quality of crops are greatly

Journal of Informatics and Web Engineering
https://doi.org/10.33093/jiwe.2024.3.1.10
© Universiti Telekom Sdn Bhd. This work is licensed under the Creative Commons BY-NC-ND 4.0 International License.
Published by MMU Press. URL: https://journals.mmupress.com/jiwe

influenced by plant diseases and pests. The traditional methods of disease detection are often done manually and have been shown to be inefficient, costly, and time-consuming.

In recent years, researchers have increasingly employed deep learning for automating plant disease detection and analysis through computer vision methods, a crucial step to guarantee both the quantity and quality of crop production. Deep learning has consistently demonstrated superior performance compared to traditional techniques in the realm of digital image processing. The convolutional neural network (CNN) family has made breakthroughs compared to most of the previous image processing techniques since the outbreak growth in data. The use of image processing methods in plantation fields is no longer novel. The You Only Look Once (YOLO) model has garnered the attention of many researchers since it was introduced in 2015 for object detection which employs CNN due to its speed, accuracy, ability to learn deep feature information of images and simple architecture [3][4]. The latest YOLOv8 model was just released in the latter half of 2022.

For these reasons, this research undertakes the study and development of both a traditional YOLOv8 model and an optimized lightweight deep learning object detection model. These models aim to effectively and accurately detect and classify plant diseases by using a neural network approach by implementing the YOLOv8 model with Ghost module and CA mechanism that overcomes the drawbacks of the traditional systems, warning farmers before any significant losses are incurred. This research will compare and analyze the performance of the proposed model and the conventional YOLOv8 model in the field of plant condition assessment to find out the difference in performance.

II. LITERATURE REVIEW

A. State-of-the-Art Object Detection Methods

Ozguven and Adem [5] proposed an improved Faster R-CNN architecture for the detection of leaf spot disease in sugar beetroot utilizing 155 images for training and testing. This architecture was created by changing the parameters of a CNN model and a Faster R-CNN architecture. The altering of parameters due to limited datasets in the updated Faster R-CNN allows similar success rates in disease detection compared to the Faster R-CNN model where the diseased areas cannot be segmented properly which shows the importance of adjusting the parameters of CNN architecture in the condition where datasets are limited. The platform's uneven lighting, where leaf photographs are taken, hurts the success rate of this study. The suggested approach successfully identified the disease in 111 out of the 117 photos of sugar beet leaf. The overall classification rate was found to be 95.48% on the leaf spot disease detection in sugar beet leaves. This research is done to detect photos taken in real-time.

Song et al. [6] presented a YOLOv4 algorithm to detect, identify and localize citrus diseases from the photos taken. Citrus leaves with two different kinds of diseases which are Citrus Canker, and Citrus Greening are provided in the dataset. They employed two data augmentation methods: pixel-wise changes, such as noising and rotations, to introduce distortions to the data or images, along with Mosaic data augmentation. Four training images were combined to create a new image. It achieves an accuracy of 95.4% and 30 fps in Frame Per Second (FPS).

Shill and Rahman [7] aimed to compare the performance of YOLOv3 and YOLOv4 in the study of 13 plant species with 17 various diseases using the PlantDoc dataset which makes a sum of 30 classes. All the photos were initially downsized to 416x416 such that the result would be 13x13x3 (5+number of classes). Data labelling is a crucial step that comes after resizing the image. They translated each image's label into a darknet format so that the model could be trained more easily. In the case of the YOLOv3 model, they employed standard augmentation techniques, including random flipping, random cropping, and random translating. However, for the YOLOv4 model, certain unique augmentation techniques such as Mosaic data increased and DropBlock regularization was applied. The YOLOv3 feature extractor is known as Darknet-53. In YOLOv4, features are extracted using the CSPDarkNet53 model. The overall performance of YOLOv4 is superior to that of YOLOv3, with a mean average precision of 55.45% for YOLOv4 and 53.08% for YOLOv3. They suggested that increasing the number of photos for every class, enhancing photo quality and using various augmentation techniques while training could aid in improving the model's accuracy as well as combining other feature extraction techniques and saliency map with the YOLO model.

Mathew and Mahesh [8] presented a YOLOv5 in identifying bell pepper plants with bacterial spot disease using the Plant Village dataset from Kaggle. The results achieved in this research show that the suggested model is more accurate than other models such as Region-based Fully Convolutional Networks (R-FCN), SSD, Region-based Convolutional Neural Networks (R-CNN) and previous YOLO. The model uses augmentation methods such as scaling, colour space adjustments, and mosaic augmentation. Mosaic augmentation helps to overcome the 'small object problem' where bacterial minor spots can be detected. The method implemented in this research outperforms earlier versions of YOLO in terms of speed and accuracy. Hardware implementation becomes simpler due to the weight file being only 27 MB in size, 90% smaller than YOLOv4. They suggested extending the disease detection to other diseases that may be experienced by the bell pepper plant.

Zhang et al. [9] integrated CA into YOLOv5 which highlights and focuses on the specific visual features associated with the grape downy mildew (GDM) disease to improve detection efficiency under natural environments. According to experimental data, the proposed YOLOv5-CA achieved a detection precision of 85.59%, a recall of 83.70%, and an mAP@0.5 of 89.55%. Notably, these results were unaffected by varying illness levels of GDM and illumination conditions.

Li et al. [10] introduced CA and Transformer structures were incorporated into YOLOv5n to capture global information in cucumber disease detection, effectively reducing the impact of complex background information. The resulting model achieved an impressive FPS of up to 143 and an mAP@0.5 of 84.9% on the test set, all within a compact model size of just 4.7 MB. To enhance the detection of tiny lesions in leaves, two multi-scale training approaches, anchor clustering, and random scaling of input picture resolution, are integrated with feature extraction structures tailored for multiple scale targets.

Wang et al. [11] proposed an optimised lightweight YOLOv5 model for plant disease detection and classification using the PlantVillage and a self-made peanut diseases dataset. To improve the model's precision and effectiveness, an improved attention submodule (IASM) mechanism was used. Weighted boxes fusion (WBF) structures and Ghostnet were employed to reduce model weight as well as combining bidirectional feature pyramid network (BiFPN) and rapid normalisation fusion for weighted feature fusion to hasten the learning rate of each feature layer. The optimised YOLOv5 model for disease classification had accuracy, recall, and F1 scores of 93.73%, 92.94%, and 92.97%, correspondingly.

B. Classification Methods

Bedi and Gole [12] incorporated the Squeeze-and-Excitation Module alongside the Ghost Module to enhance the detection of plant diseases, all while reducing the number of trainable parameters. During training and validation, the PlantGhostNet model achieved impressive accuracies of 99.75% and 99.51%, respectively, particularly excelling in identifying the bacterial spot disease that affects peach plants. Given its high accuracy and efficient parameter usage, the PlantGhostNet model is well-suited for deployment on low-processing-power devices, such as smartphones, tablets, and other mobile devices.

A lightweight CNN with 20 layers and fewer trainable parameters was proposed by Bhujel et al. [13]. According to the findings, CNN's attention mechanism increased interclass recall and precision, which raised overall accuracy (>1.1%). In addition, when compared to the normal ResNet50 model, the lightweight model dramatically reduced network parameters (by about 16 times) and complexity (by about 23 times). In comparison to the model without attention modules, the model with the attention mechanism nominally enhanced the network complexity and parameters, leading to higher detection accuracy. Self-attention (SA) mechanism (99.34%) was second to Convolutional block attention module (CBAM) in terms of average accuracy (99.69%), even though all of the attention modules enhanced CNN's performance.

Table 1 shows the pros and cons of the methods discussed in the literature review. The majority of today's cutting-edge models rely on datasets like Plant Village, which are made up of leaf images obtained in a controlled lab environment and do not function as reliable representations of real-world scenarios. To achieve higher outcomes in the diagnosis of plant disease, DL approaches demand a larger amount of data. This is a disadvantage because currently available datasets are typically tiny and lack sufficient images, which are required for high-quality decisions. A comprehensive collection should include images shot in as many diverse circumstances as feasible. This is because the lack of sufficient training images of varieties of plant leaves with diseases is the challenging problem of performing deep learning tasks in the domain of plant disease detection. The present success of deep learning models is typically due to classification models and large datasets that consist of lab-controlled images. It is still difficult to deploy plant disease object detection models on resource-constrained devices or in real-time applications. An important research direction is the development of efficient and lightweight models that can perform inference on edge devices without sacrificing accuracy and utilising the least number of resources while completing the most throughput.

Table 1. Pros and Cons of Plant Disease Detection Methods

Proposed Method	Pros	Cons
CNN	• Reduce computational cost • Reduce network complexity • Superior feature extraction task • Optimal for object classification task	• Require a large amount of dataset • Bad at handling rotation and scale invariance without data augmentation • Not optimal for object detection
CNN + GhostNet	• High accuracy • Lower number of trainable parameters • Lighter than CNN	
CNN + Attention mechanisms	• Higher accuracy than CNN	
YOLO	• Frame processing is superior to real-time • Better image generalisation • Fast speed	• Difficulty detecting small objects • Trouble detecting close objects • Lower accuracy than Faster R-CNN
YOLOv5 + GhostNet	• Lightweight model due to a lower number of parameters	
YOLOv5 + CA	• Higher detection accuracy in natural environments	
Faster R-CNN	• High accuracy and object detection rate • Use one CNN for regional proposal	• Slower than one-stage detector due to multistage pipeline • Detect the whole image by RPN with a slow speed
Mask R-CNN	• Simple to train • Very efficient • Easy to generalise to other tasks	

III. RESEARCH METHODOLOGY

A. Modified YOLOv8 Algorithm

YOLOv8 is a high-performance object detection method that is particularly good at recognising huge targets. However, due to detecting scale limitations, it is easy to miss detection for small targets when several targets overlap or are arranged compactly.

The improved YOLOv8 is proposed below to enhance the model's detection performance and make it more responsive to leaf detection, particularly for small leaf targets as well as reduce the computational resources required. YOLOv8n/s is utilised as the benchmark model in this study to integrate the Ghost module and the coordinate attention mechanism in YOLOv8 architecture to enhance the accuracy and efficiency of the model. Figure 1 shows the network design of the improved YOLOv8 algorithm.

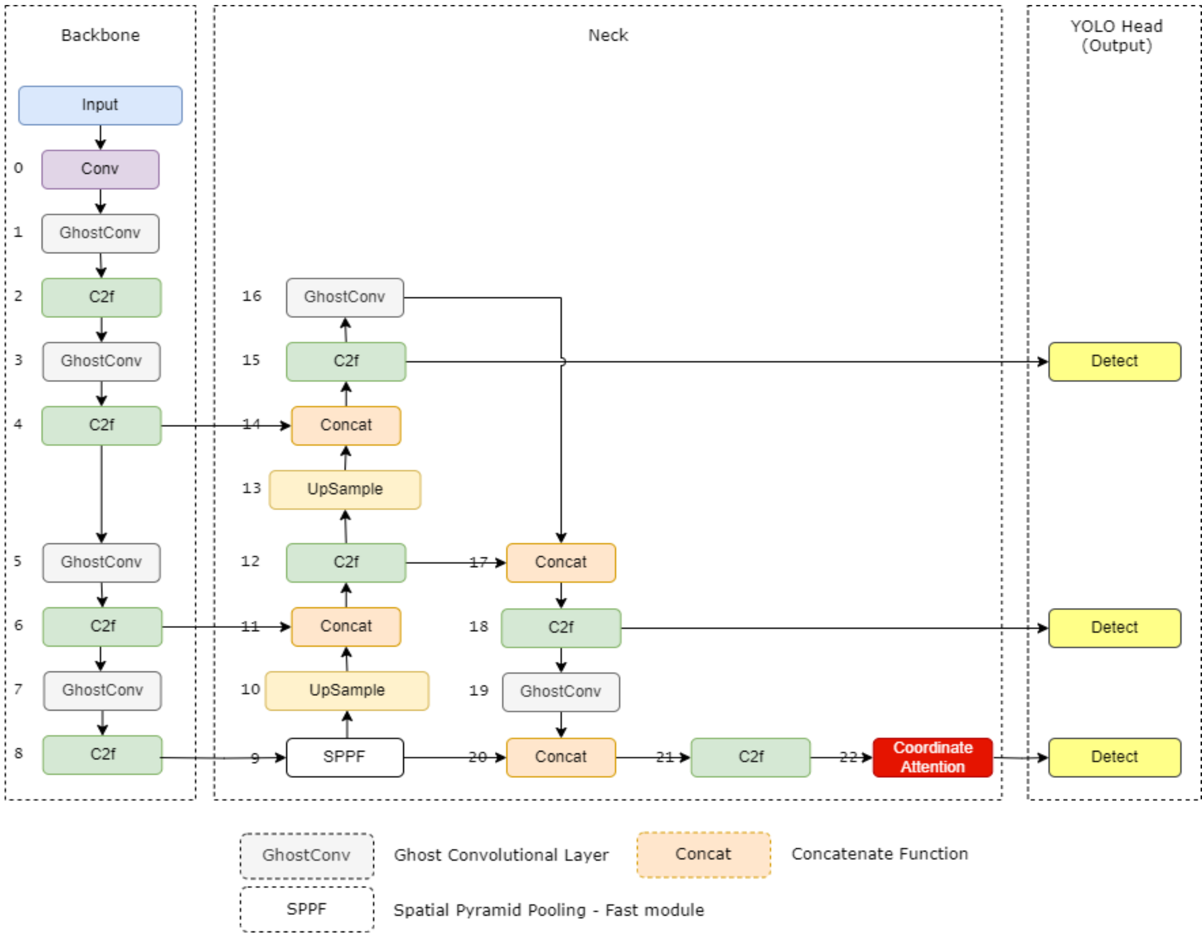

Figure 1. Improved YOLOv8 Network Design

B. GhostNet module

A GhostNet module is a type of CNN which is a plug-and-play module built using Ghost modules to replace general convolution, that aims to produce more features with fewer parameters using cheap operations. The main intuition behind it is the redundancy of feature maps generated by typical CNN. The core component of GhostNet is a stack of Ghost bottlenecks, with Ghost modules serving as the building blocks. The architecture of the GhostNet module is based on MobileNetV3 architecture by replacing the bottleneck block in MobileNetV3 with the Ghost bottlenecks. The module structure of Ghost bottlenecks is shown in Figure 2. The first Ghost module expands the quantity of channels available by acting as a layer. The expansion ratio, which is the ratio of input channels to output channels, defines a key aspect of the architecture. In the second Ghost module, channel reduction takes place to align with the shortest path. A shortcut connection is then established between the inputs and outputs of these two Ghost modules. Following this pattern, except for the second Ghost module inspired by MobileNetV2, batch normalization (BN) and ReLU nonlinearity are applied after each layer. It's important to note that the Ghost bottleneck, as described above, pertains to cases where the stride is equal to 1. In instances where the stride is set to 2, a downsampling layer is introduced, and a depthwise convolution (often referred to as DWConv) with a stride of 2 is inserted between the two Ghost modules. To enhance efficiency, the primary convolution within the Ghost module employs pointwise convolution. It is worth mentioning that, with the exception of the final bottleneck within each stage, all Ghost bottlenecks operate with a stride of 1, rather than 2.

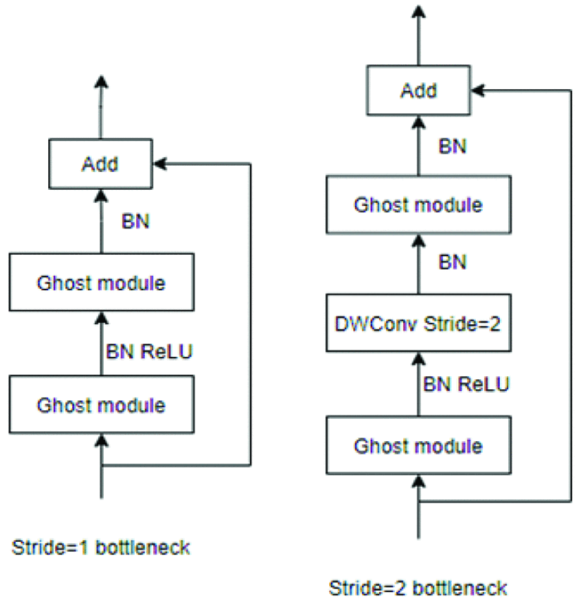

Figure 2. Ghost Bottleneck. Left: Ghost Bottleneck With Stride=1; Right: Ghost Bottleneck With Stride=2.

The first layer of the GhostNet module is a typical convolutional layer with 16 filters, while the subsequent layers consist of a sequence of Ghost bottlenecks with progressively more channels. GhostNet generates a certain percentage of the total output feature maps by standard convolution while the remaining is generated by a cheap linear operation, which is the DWConv. This cheap linear operation results in a significant decrease in parameters and FLOPs while preserving approximately the same performance as the original baseline model.

This study suggests utilising the GhostNet module to replace the convolutional layer in the Backbone network as it is a lightweight structure. The ghost module can lessen the weight and decrease the parameters of the network. This method employs an economical calculation approach to consolidate redundant feature map layer information, thereby reducing the computational load of the model and enhancing its operational speed. The network parameters can be compressed with the help of this module. The operational cost is decreased by the compressed network parameters. The model's endurance for spatial layout and object recognition is improved by the ability to extract spatial feature information of various sizes in place of conventional convolution operation [14]. GhostConv is used in place of Conv in the backbone and head layers of YOLOv8. By doing this, we are attempting to nearly completely avoid overfitting by reducing the parameters that must be learned in the network. The GhostConv module enables the reading of redundant material, improving comprehension of the input data. The model's depth and width can be reduced using GhostConv. This lowers the cost of computing.

C. Attention mechanism: Coordinate Attention (CA)

Convolutional networks have seen a lot of success in various computer vision applications. However, the convolution process in convolutional networks has a severe flaw in that it only operates in a local area, leaving out global information. Self-attention has emerged as a new improvement in capturing long-range interactions, but it has so far been used mostly for sequence modelling and generative modelling problems. Attention processes have been revealed to be beneficial in a range of computer vision tasks. However, most attention mechanisms are primarily focused on channel information, which limits expressivity by denying the spatial information present.

Therefore, the introduction of the CA mechanism, a network attention mechanism proposed by Hou et al. [15], aims to enhance the importance of key features by incorporating positional information into channel attention. In this research, CA is integrated into the YOLOv8 architecture's neck section, with the goal of enhancing the accuracy of plant disease detection. CA not only captures cross-channel information but also incorporates direction-aware and position-sensitive details through the integration of positional information into channel attention. The network can precisely determine the position of a specific object, boasting a more extensive receptive field compared to both the Balanced Attention Mechanism (BAM) and CBAM. The CA mechanism increases information flow efficiency in the neural network by assisting the neural network in paying attention to valid coordinates and suppressing invalid coordinates. CA mechanism has garnered increasing notice as it can deliver large performance advantages to downstream jobs, particularly those with dense predictions than other attention mechanisms with the lightweight property.

The CA mechanism can be divided into two steps: Coordinate information embedding and coordinate attention generation. The specific principle is illustrated in Figure 3.

- Step 1: Coordinate information embedding

Global pooling (Equation (1)) is used for global encoding in the channel attention mechanism. However, it is difficult to retain location information. To address this issue, the CA module converts it into a pair of one-dimensional feature encoding operations.

$$Z_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} x_c(i,j) \tag{1}$$

To elaborate, when provided with input X, two pooling kernels, (H, 1) for encoding information along the horizontal coordinate and (W, 1) for the vertical dimension, are employed. The output for the c-th channel at height h can be expressed in Equation 2.

$$Z_c^h(h) = \frac{1}{w} \sum_{0 \le i < w} x_c(h,i) \tag{2}$$

Similarly, the output of the c-th channel along width w dimensions can be formulated as in Equation 3.

$$Z_c^w(w) = \frac{1}{H} \sum_{0 \le i < H} x_c(j,w) \tag{3}$$

Consequently, these two transformations play a crucial role in enabling the attention mechanism to grasp long-term dependencies in one direction while preserving precise location information in the other direction. This capability enhances the network's ability to pinpoint objects with greater accuracy.

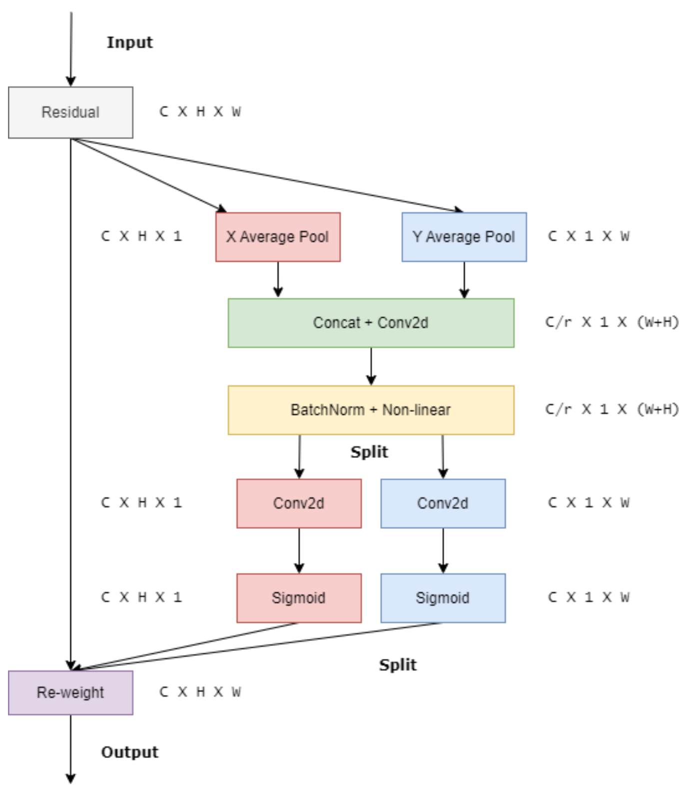

Figure 3. Coordinate Attention Module

- Step 2: Coordinate attention generation

To harness the embedded information more effectively, a swift and efficient method of coordinate attention generation was employed as the second transformation. The CA module utilizes a shared 1×1 convolutional transform function to combine and modify these pieces of information as in Equation (4).

$$f = Relu(F_1([z^h, z^w])) \tag{4}$$

where, [,] indicates concatenate operation along the spatial dimension, $f \in R^{C/r\times(H+W)}$ is the output feature map of the ReLU layer, and r is the reduction rate.

Subsequently, f is then decomposed into two separate tensors, $f^h \in R^{C/r\times H}$ and $f^w \in R^{C/r\times W}$. The feature maps, f^w and f^h, are used to restore to the same number of channels as input $x_c\ (i, j)$, with the following results (see Equations (4) and (5)).

$$g^h = \sigma(F_h)(f^h)) \tag{5}$$

$$g^w = \sigma(F_w)(f^w)) \tag{6}$$

where σ is the sigmoid activation function, and F_h and F_w are the convolution operations in the height and width direction for f^h and f^w separately.

Then, expanding g^h and g^w and using as attention weights, the CA module's final output is as in Equation (7).

$$y_c(i, j) = x_c(i, j) \times g_c^h(i) \times g_c^w(j) \tag{7}$$

where y_c is the c-th channel within the generated feature map y of the attention block and $y_c(i, j)$ is the result of the attention-weighted feature map in the width and height directions.

This procedure can strengthen the features that are directly related to leaf diseases and lessen the influence of irrelevant information.

D. Experimental Equipment Setup

The overall implementation is implemented using Google Colab. PyTorch 1.13.1 is the framework which is used to implement the proposed YOLOv8 algorithm and the experiment was conducted on Python 3.10.11. For hardware configuration, NVIDIA Tesla T4 GPU, 12GB RAM, and CUDA version 11.8 was used.

E. Dataset

In this paper, the PlantDoc dataset is used and obtained from Roboflow [16]. A recent study achieved a mean average precision of 71% on this dataset by leveraging YOLOv7 [17]. There are a total of 2,569 coloured leaf images in the dataset across 13 plant species and 30 classes which consist of healthy and diseased leaves in various natural backgrounds for image classification and object detection. This large-scale, non-lab dataset was developed by Singh et al. [18] to accelerate deep learning solutions for the detection of plant diseases because no other dataset accessible gives images captured in real-world settings. The dataset is split into training, validation and testing sets proportionally in every class with no overlap between the three sets to balance the distribution of classes evenly across the split groups and maintain the original class proportions with an 80:10:10 ratio. This ensures that each class is represented proportionally in the training, validation and testing sets to its occurrence in the entire dataset. This dataset was selected because it contains photographs taken in real settings, as opposed to images obtained in lab conditions in other datasets and its low and imbalanced sample size as well as the fact that a limited number of plant disease detection research were carried out using this dataset.

F. Data Preprocessing

A series of data augmentation and enhancement techniques are carried out to reduce network overfitting and improve the generalisation ability. Data augmentation enhances the learning of the deep model. As the dataset is small and imbalanced, some data preprocessing techniques are used to diversify the images and increase the training performance of the model. After image transformation and image augmentation (Figure 4), the size of the dataset is increased to 21,289 images. Some of the preprocessing techniques are:

- Auto-orientation: This method aims to extract an image from its Exchangeable Image File Format (EXIF) data by identifying the image's orientation, ensuring it is displayed in the same manner as when it was initially saved. EXIF files contain essential information pertaining to photographs, and they are generated by virtually all digital cameras when a photograph is captured. Within an EXIF file, one can find comprehensive metadata about the image, encompassing details such as exposure settings, location, orientation, and camera configurations.
- Object isolation: Crop and isolate bounding boxes to create separate independent and unique images.
- Static crop: An image is cropped to a specific horizontal or vertical section. In our research, the images are cropped into a 10% to 90% range of horizontal and vertical sections.
- Cutout: An augmentation technique which covers part of an image with a square at random that aids in training model to identify partially or completely obscured objects.
- Filter null images: Require at least 90% of images to contain annotations. By removing unnecessary samples, filtering out null images streamlines future data processing activities. This decreases the possibility of inaccurate conclusions or biased results from incorporating null photos in the study.
-

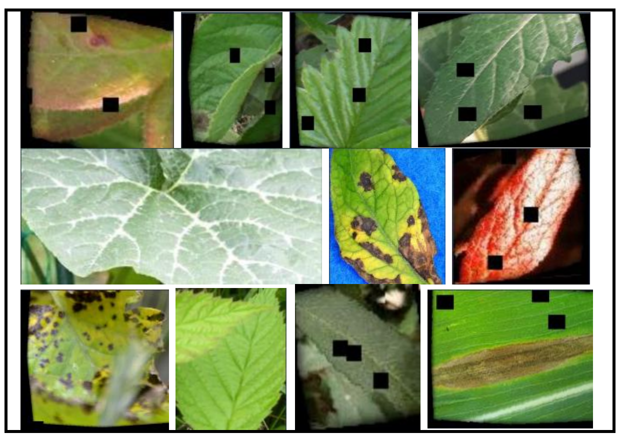

Figure 4. Sample Images in the Dataset after Data Preprocessing

G. Performance Indicators

To assess the effectiveness of the proposed algorithm, we evaluate its performance by comparing various metrics, including precision, recall and mAP, with those achieved by the original YOLOv8 algorithm as well as other object detection models.

Precision measures the relevance of the results whereas recall evaluates how many truly relevant results are returned (Equations (8) and (9)). The mAP stands for Mean Average Precision, which is the mean value of AP for different categories (Equation (10)).

$$Precision = \frac{TP}{TP + FP} \tag{8}$$

$$Recall = \frac{TP}{TP + FN} \tag{9}$$

$$mAP = \frac{\Sigma AP}{n} \tag{10}$$

In Equations (8)-(10), TP (True Positive) denotes the count of accurately identified diseased leaves, FP (False Positive) represents the number of healthy leaves incorrectly classified as having a disease, FN (False Negative) signifies the instances of leaves with diseases being incorrectly identified as healthy. Additionally, AP denotes the area under the precision-recall curve, and 'n' stands for the total number of classes. The mAP includes both instances of false positives (FP) and false negatives (FN) while displaying the balance between precision and recall. This property makes mAP a desirable metric for a wide range of detection tasks.

IV. RESULTS AND DISCUSSIONS

The $mAP_{0.5}$ is chosen as the main metric to evaluate the results as it takes into accounts both FP and FN and reflects the trade-off between precision and recall. Table 2 summarises the performance of different variations of the experiment.

Table 2. Performance Scores of Models Without Data Preprocessing on the PlantDoc Dataset

Model	Precision	Recall	$mAP_{0.5}$(%)	$mAP_{0.5:0.95}$(%)
YOLOv8s	58.48	53.67	58.02	40.42
YOLOv8n	56.17	53.80	57.97	39.82

When evaluating the models without any data preprocessing, one of the important observations from Table 2 is that the models' performance was comparatively poor compared to the versions that underwent preprocessing from Table 3. Data preprocessing techniques are critical in enhancing the quality and applicability of input data for models. We saw a significant improvement in the overall performance of both the YOLOv8n and YOLOv8s models after employing these techniques.

Furthermore, we investigated the effect of transfer learning on model performance. Transfer learning is a technique that allows models to learn from existing knowledge and adapt it to new tasks by leveraging pre-trained models on huge datasets. When compared to non-transfer learning equivalents, the addition of transfer learning had a surprising effect on both models, resulting in significant performance gains. This discovery emphasises the efficacy of using pre-trained models as a starting point and fine-tuning them for specific tasks, ultimately improving the detection capabilities of YOLOv8n and YOLOv8s.

When comparing YOLOv8n to YOLOv8s, it is worth noting that the latter demonstrated greater performance. This is due to the YOLOv8s architecture's bigger size, increased depth, and increased number of parameters. Due to its increased capacity, YOLOv8s can capture more subtle information and representations, allowing it to achieve higher detection accuracy. The model's increased complexity provides for a more comprehensive grasp of the input data, resulting in enhanced performance in object detection tasks. Overall, our findings highlight the significance of data preparation and transfer learning in maximising the potential of YOLOv8n and YOLOv8s models. Furthermore, the findings emphasise the benefits of the YOLOv8s model in terms of higher performance, emphasising the importance of architectural advances in object detection models.

Besides, we present findings that demonstrate the usefulness of including the CA mechanism in the models, outperforming the performance of the conventional models shown. Specifically, the inclusion of the CA mechanism in YOLOv8n, coupled with transfer learning, led to the highest $mAP_{0.5}$ of 68.8%, which is 4.2% higher than the conventional model, as compared to other variations explored in our experiments for the baseline model. This demonstrates the significant impact of the CA mechanism in improving the model's ability to accurately detect objects, especially when combined with transfer learning while only slightly increasing the computational complexity of the model. Likewise, when we incorporated the CA mechanism in YOLOv8s alongside transfer learning, the model achieved the highest $mAP_{0.5}$ of 72.2%, among all the experimental variations including those involving the YOLOv8n model. This result emphasises the effectiveness of the CA mechanism in enhancing the detection performance of YOLOv8s, further solidifying its superiority over other models. These results underline the potential of incorporating attention mechanisms like CA to improve object detection capabilities and provide insights for further advancements in the field. As depicted in Figure 5, the incorrect predictions are mostly due to blurred images, cropped leaves or overlapping leaves in the images. Better preprocessing and augmentation techniques such as denoising, deblurring and sharpening as well as bounding box adjustment on the initial dataset may improve the performance of the current model. A popular approach, ensemble models may be infused into the model for enhanced performance but it is essential to note that ensemble methods can be computationally expensive which contradicts the objective of this study.

Figure 5. Sample Results from YOLOv8s with CA Mechanism. Left: Validation Batch Labels; Right: Validation Batch Prediction

The integration of the GhostNet module into the YOLOv8 model primarily aims to decrease the model's computational complexity and enhance its overall lightweight nature, but it can be seen from Table 3 that this integration also makes the model perform better than its predecessor by improving the model's capability to identify and categorise plants and their diseases in the dataset.

Table 3. Performance Scores of Models on the PlantDoc Dataset. The bold font indicates the best results.

Model	Ghost module	CA mechanism	Transfer learning	Precision	Recall	$mAP_{0.5}$(%)	$mAP_{0.5:0.95}$(%)
YOLOv8s (baseline)			✓	66.6	68.8	70.7	70.3
				64.3	60.7	65.4	65
		✓	✓	**71.6**	66.5	**72.2**	**71.5**
		✓		66.2	57.8	66.2	65.8
	✓		✓	59.9	68.8	69.2	68.7
	✓			59	69	68.5	68
	✓	✓	✓	60.1	**72.5**	69.1	68.6
	✓	✓		65.2	64.3	69.3	68.9
YOLOv8n (baseline)			✓	62.8	65.1	67.6	67.2
				57.8	63.4	64.6	64.3
		✓	✓	64.1	67.6	68.8	68.4
		✓		59.3	65.8	66.4	65.9
	✓		✓	63.1	63.9	68.1	67.7
	✓			62.7	60.5	66.5	66
	✓	✓	✓	62.8	62.7	66.6	66.3
	✓	✓		66.9	58.6	68	67.7

The YOLOv8n/s achieved the highest mAP when both the GhostNet module and the CA mechanism are incorporated in the architecture of the conventional model while lowering the computational complexity of the model, accomplishing two goals at once, when compared to other variations of experiments without taking transfer learning into account. For some unknown reasons, the models with the GhostNet module and CA mechanism with the presence of transfer learning did not achieve better performance than the model without transfer learning as opposed to other experimental variations. This may need investigation in future work.

Table 4 summarises the speed performance of the models using Giga Floating-Point Operations Per Second (GFLOPs), parameter count and inference speed. YOLOv8n has a lower amount of GFLOPs and several parameters than YOLOv8s due to its smaller size. Thus, the inference speed of YOLOv8n is faster than YOLOv8s. As depicted in the table, YOLOv8n/s have a lower number of GFLOPs and parameters with the presence of the GhostNet module. This proved that the initial purpose of introducing the GhostNet module into the architecture to achieve a more lightweight design has been achieved. Models with the presence of a CA mechanism were initially expected to increase the computational complexity of the model by a little. However, the final outcomes show that they have the same number of GFLOPs and inference speed as the conventional models with just a slight difference in the number of model parameters in which models with CA mechanism have 0.01M higher number of parameters. This indicates that the CA mechanism is a suitable attention mechanism to improve the accuracy of the model without too much increase in computational complexity. The conventional YOLOv8s has an inference speed comparable to YOLOv8n due to the input image size of 416×416 for the model which is different from the other models as their input image size is 640×640. The difference in image sizes used is because this research utilised the image size with the best training result out of these two sizes.

Table 4. Performance Metrics of the Models: GFLOPS, Parameter Count, and Inference Speed

Model	Ghost	CA mechanism	Transfer Learning	GFLOPs	Parameter quantity	Inference speed (ms)
YOLOv8s			✓	28.5	11.13 M	1.7
						1.8
		✓	✓	28.5	11.16 M	1.7
		✓				3.2
	✓		✓	26.3	10 M	3.6
	✓					3.2
	✓	✓	✓	26.3	10.02 M	3.4
	✓	✓				3.4
YOLOv8n			✓	8.1	3.01 M	1.9
						2.1
		✓	✓	8.1	3.02 M	1.9
		✓				1.9
	✓			7.6	2.73 M	2.0
	✓		✓			1.9
	✓	✓	✓	7.6	2.74 M	1.9
	✓	✓				1.9

The comparison of various models is presented in Table 5. Our research selected the trained model with the highest performance overall for comparison purposes with existing studies, which is, YOLOv8s with the integration of the CA mechanism. From the table, it can be observed that our approach has outperformed the other methods by achieving the highest $mAP_{0.5}$ of 72.2%. This shows that the proposed model has the potential for the implementation of plant disease detection. Moreover, it can be observed that the YOLOv7 by Vaidya et al. [17] performs just slightly worse than our proposed model. However, the YOLOv7 model proposed is computationally heavier than YOLOv8s with 75.1 MB in size and with an average time to detect an irregularity in an image (inference time) of 6.8 ms, which shows that it is not very suitable for lightweight implementation even though it has high performance. On the other hand, even though YOLOv8s is computationally more complex than YOLOv8n, its size of 22.3 MB can still be considered lightweight while maintaining high accuracy.

Table 5. Comparison between the Proposed Work with Other Researchers' Work on the Same PlantDoc Dataset. The bold font indicates the best results.

Study	Approach	$mAP_{0.5}$(%)
Karantoumanis et al., 2022 [19]	Faster R-CNN	42.8
Shill & Rahman, 2021 [5]	YOLOv3	53.08
Shill & Rahman, 2021 [5]	YOLOv4	55.45
Jocher et al., 2020 [20]	YOLOv5s	53.5
RangiLyu, 2020/2023 [21]	nanodet-plus	55.3
Li et al., 2022 [9]	Improved YOLOv5	58.2
Liu et al., 2022 [22]	YOLOX-ASSANano	58.85
Vaidya et al., 2023 [17]	YOLOv7	71
Our Approach	YOLOv8s with Coordinate Attention	**72.2**

V. CONCLUSION

Plant disease detection is a critical domain to be investigated as they contribute to sustainable agriculture, higher crop yields, lower costs, and better resource management [23]. Through this research, we have addressed the critical challenge of higher accuracy with lower computational cost in a small imbalanced dataset with complicated backgrounds. The result shows that the best performance was obtained using YOLOv8s in the PlantDoc dataset with CA mechanism and transfer learning with 72.2% mAP. The best result achieved without transfer learning was obtained using YOLOv8s with Ghost module and CA mechanism with 69.3% mAP. Our approach has yielded a good result that surpasses and outperforms the studies in the same field using the same dataset. The efficient implementation of our technique provides practitioners and researchers with significant insights into plant disease detection and classification, allowing them to make more accurate forecasts and informed decisions.

Additional research avenues could explore alternative data preprocessing techniques to potentially enhance the model's performance. Additionally, there might be other optimal parameter configurations that can contribute to the model's effectiveness. Further investigations can be conducted to experiment with the integration of the GhostNet module at different locations, aiming to reduce the number of trainable parameters and GFLOPs in the conventional YOLOv8 model. Additionally, it is worth exploring different locations for the CA mechanism within the YOLOv8 architecture, as this may lead to higher performance compared to the current placement of the CA mechanism in the YOLOv8 model utilised in this work. Due to the dataset's small scale and limited variations, the model's performance is relatively modest. It would be beneficial to generate and combine more datasets for plant disease detection captured in various environments to create a larger and more diverse dataset, enabling the model to learn more effectively.

ACKNOWLEDGEMENT

A version of this paper was presented at the third International Conference on Computer, Information Technology and Intelligent Computing, CITIC 2023, held in Malaysia on 26th-28th July 2023.

The research for this paper was not funded.

REFERENCES

[1] J. D. Lovrenciear, “Malaysia’s looming food security crisis needs critical appraisal”, 6th February 2022. https://www.malaysiakini.com/news/609642.

[2] W. Mbinda, A. Kavoo, F. Maina, M. Odeph, C. Mweu, N. Nzilani and M. Ngugi, “Farmers’ knowledge and perception of finger millet blast disease and its control practices in western Kenya”, CABI Agriculture and Bioscience, vol. *2,* no. 1, pp. 13, 2021. https://doi.org/10.1186/s43170-021-00033-y.

[3] T. Diwan, G. Anirudh and J. V. Tembhurne, “Object detection using YOLO: Challenges, architectural successors, datasets and applications”, Multimedia Tools and Applications, 2022. https://doi.org/10.1007/s11042-022-13644-y.

[4] T. -H. Wu, T. -W. Wang and Y. -Q. Liu, “Real-Time Vehicle and Distance Detection Based on Improved Yolo v5 Network”, The 3rd World Symposium on Artificial Intelligence (WSAI), 2021, pp. 24–28. https://doi.org/10.1109/WSAI51899.2021.9486316.

[5] M. M. Ozguven and K. Adem, “Automatic detection and classification of leaf spot disease in sugar beet using deep learning algorithms”, Physica A: Statistical Mechanics and Its Applications, vol. 535, 122537, 2019. https://doi.org/10.1016/j.physa.2019.122537.

[6] C. Song, C. Wang and Y. Yang, “Automatic Detection and Image Recognition of Precision Agriculture for Citrus Diseases”, The IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE), 2020, pp. 187–190. https://doi.org/10.1109/ECICE50847.2020.9301932.

[7] A. Shill and Md. A. Rahman, “Plant Disease Detection Based on YOLOv3 and YOLOv4”, 2021 International Conference on Automation, Control and Mechatronics for Industry 4.0 (ACMI), 2021, pp. 1–6. https://doi.org/10.1109/ACMI53878.2021.9528179.

[8] M. P. Mathew and T. Y. Mahesh, “Leaf-based disease detection in bell pepper plant using YOLO v5”, Signal, Image and Video Processing, vol. 16, no. 3, pp. 841–847, 2022. https://doi.org/10.1007/s11760-021-02024-y.

[9] Z. Zhang, Y. Qiao, Y. Guo and D. He, “Deep Learning Based Automatic Grape Downy Mildew Detection”, Frontiers in Plant Science, vol. 13, 872107, 2022. https://doi.org/10.3389/fpls.2022.872107.

[10] J. Li, Y. Qiao, S. Liu, J. Zhang, Z. Yang and M. Wang, “An improved YOLOv5-based vegetable disease detection method”, Computers and Electronics in Agriculture, vol. 202, 107345, 2022. https://doi.org/10.1016/j.compag.2022.107345.

[11] H. Wang, S. Shang, D. Wang, X. He, K. Feng and H. Zhu, “Plant Disease Detection and Classification Method Based on the Optimized Lightweight YOLOv5 Model”, Agriculture, vol. 12, no. 7, Article 7, 2022. https://doi.org/10.3390/agriculture12070931.

[12] P. Bedi and P. Gole, “PlantGhostNet: An Efficient Novel Convolutional Neural Network Model to Identify Plant Diseases Automatically”, The 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), 2021, pp. 1–6. https://doi.org/10.1109/ICRITO51393.2021.9596543.

[13] A. Bhujel, N. -E. Kim, E. Arulmozhi, J. K. Basak and H. -T. Kim, “A Lightweight Attention-Based Convolutional Neural Networks for Tomato Leaf Disease Classification”, Agriculture, vol. 12, no. 2, Article 2, 2022. https://doi.org/10.3390/agriculture12020228.

[14] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu and C. Xu, “GhostNet: More Features from Cheap Operations”, (arXiv:1911.11907; Version 1). arXiv. 2019. http://arxiv.org/abs/1911.11907.

[15] Q. Hou, D. Zhou and J. Feng, “Coordinate Attention for Efficient Mobile Network Design”, (arXiv:2103.02907; Version 1). arXiv. 2021. http://arxiv.org/abs/2103.02907.

[16] Roboflow, Dataset, 2023. https://public.roboflow.com/object-detection/plantdoc.

[17] S. Vaidya, S. Kavthekar and A. Joshi, “Leveraging YOLOv7 for Plant Disease Detection”, The 4th International Conference on Innovative Trends in Information Technology (ICITIIT), 2023, pp. 1–6. https://doi.org/10.1109/ICITIIT57246.2023.10068590.

[18] D. Singh, N. Jain, P. Jain, P. Kayal, S. Kumawat and N. Batra, “PlantDoc: A Dataset for Visual Plant Disease Detection”, Proceedings of the 7th ACM IKDD CoDS and 25th COMAD, 2020, pp. 249–253. https://doi.org/10.1145/3371158.3371196.

[19] E. Karantoumanis, V. Balafas, M. Louta and N. Ploskas, “Computational comparison of image preprocessing techniques for plant diseases detection”, The 7th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM), 2022, pp. 1–5. https://doi.org/10.1109/SEEDA-CECNSM57760.2022.9932972.

[20] G. Jocher, K. Nishimura, T. Mineeva and R. Vilariño, “YOLOv5 2020”, 2020. GitHub Repository: Https://Github. Com/Ultralytics/Yolov5.

[21] RangiLyu, NanoDet-Plus [Python], 2023. https://github.com/RangiLyu/nanodet. (Original work published 2020)

[22] S. Liu, Y. Qiao, J. Li, H. Zhang, M. Zhang and M. Wang, “An Improved Lightweight Network for Real-Time Detection of Apple Leaf Diseases in Natural Scenes”, Agronomy, vol. 12, no. 10, Article 10, 2022. https://doi.org/10.3390/agronomy12102363.

[23] M. Xin, L. W. Ang and S. Palaniappan, “A Data Augmented Method for Plant Disease Leaf Image Recognition based on Enhanced GAN Model Network”, Journal of Informatics and Web Engineering (JIWE), vol. 2, no. 1, pp. 1-12, 2023. https://doi.org/10.33093/jiwe.2023.2.1.1.