IJIE

Penerbit

INTERNATIONAL JOURNAL OF INTEGRATED ENGINEERING
ISSN: 2229-838X e-ISSN: 2600-7916

Vol. 18 No. 4 (2026) 1-21
https://publisher.uthm.edu.my/ojs/index.php/ijie

Deep Performance Analysis of YOLOv5 to YOLOv11 for Detecting Corn Leaf Diseases

Urmila Pilania[1]*, Sachin Kumar Aggarwal[2], Jyoti Godara[3], Manoj Kumar[1], Neeru Singh[4], Roshi Saxena[5]

[1] *Computer Science & Technology, Manav Rachna University, Sector-43, 121001, INDIA*
[2] *Synechron Inc., UNITED STATES*
[3] *Department of Computer Science and Engineering, SGT University, Gurugram, INDIA*
[4] *Computer Science & Information Technology, GL Bajaj Institute of Technology & Management, Greater Noida, INDIA*
[5] *Xebia Technologies, Gurugram, INDIA*

*Corresponding Author: urmilapilania@gmail.com
DOI: https://doi.org/10.30880/ijie.2026.18.04.001

Article Info

Received: 21 November 2025
Accepted: 15 April 2026
Available online: 17 June 2026

Keywords

Corn leaf diseases, detection, YOLO models, deep learning, classification

Abstract

Corn leaf diseases affect agriculture worldwide which leads to low production and economic concerns. Timely detection of diseases in corn leaves could improve the growth of corn. Timely detection could also optimize resource uses, decreases overall cost and confirm high-quality of the crop. Deep learning techniques are playing important role in detection and classification of different leaf diseases. You Only Look Once (YOLO) models also follow the concept of deep learning for accurately detection and classification of the diseases in corn leaves. In the proposed work, YOLOv5, YOLO6, YOLOv8, YOLOv9, YOLOv10 and YOLOv11 models are trained on dataset from Kaggle for detection of diseases in corn leaves. Data augmentation techniques such as flip vertical, rotation between -15 degree to +15 degree, 90% clockwise rotation, shearing ±0 degree horizontal to ±15 degree vertical are applied on dataset at the time of training. The model is evaluated by applying the metrics: Accuracy, Recall, Precision, mAP@50, mAP@50-95, and time taken. To validate the experimental results authors utilized 6 versions of YOLO. YOLOv9 performed best among proposed version and it is justified by the experimental results as well.

1. Introduction

Corn is the most commonly consumed crops worldwide, and health of corn crop plays crucial role for food security. Though, corn crop is vulnerable to many diseases which could significantly decrease its harvesting. The timely identification, diagnosis and treatment of these diseases are crucial to ease their influence. Traditional diseases detection methods are very time consuming and leads to the errors as well. For avoiding such type of issues these days' researchers and agricultural scientists are trying to innovative computer vision methods for real-time disease identifications [1]. In this field You Only Look Once (YOLO) versions, are playing important roles. YOLO provides fast and accurate detection of these diseases.

YOLO is a deep learning model which detects objects in real-time. YOLO process the complete image at once which helps it making very fast. It could process input images at high rate in real-time. It works very efficiently with various sizes of objects such as nano, small, medium and large. It efficiently reduces the background errors and improves ability of model to differentiate objects in input image. It can also work on images with low resolution. Good training is required to detect corn leaf diseases [2]. A large dataset of corn leaf with proper labelling is required for training. At the time of training, the model learns to select the features of various diseases and locate them within input image. After training model is deployed on devices for detection of diseases in corn leafs in real-time. It could also classify the diseases detected. It provides high accuracy in detection of diseases in corn leaves. The various versions of YOLO have been differentiating in Table 1. Research contributions for the proposed work are as follow:

i. Authors did performance assessment of YOLO versions for detection and classification of corn leaf diseases under same simulation conditions.
ii. The assessment metrics comprising: Accuracy, Recall, Precision, mAP@50, mAP@50-95, and time taken are calculated to confirm standard comparison.
iii. Authors explored different YOLO architectures and its impact on classification of corn leaf diseases.
iv. Authors identified the best appropriate YOLO version for real-time agricultural deployment.
v. Authors have taken varying leaf conditions and achieved qualitative exploration for prediction of yields to observe misclassification patterns and model robustness.
vi. Established a benchmark framework that can serve as a reference for future research in deep learning–based crop disease detection.

Paper is organized in five sections. Section 1 is the introduction of corn diseases and how YOLO can help in early detection of these diseases. Section 2 is the literature review of the corn diseases and machine learning models use in detection of these diseases. Section 3 is the proposed models to detect diseases in corn leaf. Section 4 is about the result and discussion of YOLO models. Section 5 concludes the research work along with its future scope.

Table 1 *Comparison of YOLO versions [2]*

Features Release Date	YOLOv5 2020	YOLOv6 2022	YOLOv8 2023	YOLOv9 2023	YOLOv10 2024	YOLOv11 2024
Architecture	Custom architecture as inspired by YOLOv4	Improvement over YOLOv5	Extended Efficient Layer Aggregation Network	Multiscale feature integration	It has lightweight classification head along with spatial channel decoupled down sampling and partial self-attention for global features.	It has C3k2 blocks for multi-scale feature extraction along with CBS layers for enhanced refinement and advanced modular detection head.
Speed	Balance the speed and accuracy	Balance speed, accuracy and computational cost	Higher than YOLOv7	Most efficient over all the versions of YOLO	Performed low compared to other YOLO versions	Balanced accuracy, cost and computational cost
Accuracy	Higher than YOLOv4	High accuracy with high computational speed	Higher than YOLOv6	Most efficient over all the versions of YOLO	Performed low compared to other YOLO versions	Achieved remarkable accuracy
Multi-scale Prediction	Auto learning bounding box anchors	YOLOv5 foundation	Improved architecture, further optimizations, support for different tasks	Focus on variable size objects and result in improved security	It focuses on dual-label assignment for better prediction	It focuses on instance segmentation, pose estimation, oriented bounding boxes for angled objects
Augmentation	MixUp, Mosaic, Augmentations	Advance	Advance	It does advanced augmentation	Advanced	Advanced

Features	YOLOv5	YOLOv6	YOLOv8	YOLOv9	YOLOv10	YOLOv11
Release Date	2020	2022	2023	2023	2024	2024
Deployment	Very Flexible	PyTorch, ONNX, TensorRT	PyTorch, ONNX, TensorRT	PyTorch, TensorFlow, or ONNX Runtime	PyTorch, TensorFlow, or ONNX Runtime	PyTorch, TensorFlow, or ONNX Runtime

2. Literature Review

The detection of rice diseases in early stage leads to the reduced use of pesticides and ensure good quality rice. In the work [3] authors proposed Multi-scale YOLOv5 to predict the diseases in rice. The RLD dataset has been utilized by the authors to predict the diseases in the rice. The rice leaf images were pre-processed and then labeling was done in dataset. Backbone network was created using DenseNet-201 and depth-aware instance technique was used to do the segmentation. Features were retrieved from the segmented input images which help in detection different diseases in rice leaf. The proposed work was evaluated on various parameters such as precision, accuracy, execution time, recall, inference time, and F1 score.

The most commonly occurring diseases in rice leaf are blast disease and brown spot disease. For good production of rice it is important to monitor and maintain rice leaf from many diseases. In the proposed work [4], the rice disease was detected whether it was leaf blast or brown spot disease. The YOLO algorithm was applied for the detection of the rice diseases. The custom dataset of 200 input images has been utilized to train the proposed YOLO model. For the proposed work accuracy was calculated to be 90% for leaf blast and 70% for the brown spot diseases.

The production of rice is affected by many factors like plant diseases, changes in climate, floods, droughts, and use of pests. The climate conditions are unpredictable and out of control for us. But the plant diseases can be predicted on time to enhance the growth of rice in rice plants. The proposed work [5] used YOLO for timely detection of rice diseases. With the help of spotting and discoloration on leaf of rice plants we can identify the diseases in plants. Authors in this work applied YOLO and Convolutional Neural Network (CNN) model to identify the diseases in rice leafs. They used the big public Indonesian dataset for training of the proposed model. The proposed work was evaluated by calculating the accuracy of the model and it is compared with various existing models. The diseases identified were also classified in very low time which was recorded as 40 second.

In many countries rice is the main grain which is fulfilling the needs of living hood of human beings. The plant of rice is affected by many diseases due to the degradation on environmental conditions. Authors in [6] proposed YOLOv5 for the identification of rice leaf diseases. The proposed model was compared with YOLOv3 and v4 versions of it. The result of version 5 was calculated to be best among the 3 and 4. For training of the proposed model dataset of 400 images was chosen from kaggle. The experiment was performed on google colab. For evaluation of the work accuracy was calculated for 100 epochs.

In work [7] authors proposed YOLOv5 model for the identification of rice leaf diseases. The farmers are not much educated to identify the type of diseases in plants. If the diseases are identified on the early stage then accordingly the treatment could be suggested and provided to the plants. The early detection leads to the early treatment resulting into high production of the rice crop. Authors in this work used a big dataset of 1500 images to train the model. Authors calculated precision, which was found to be 90%, recall was 67%, mAP was 76%, and F1 score was 81%.

Rice crop plays an important role in Indonesians food chain. Due to the degradation of environment many type of diseases are occurring in the rice leaves. Two most affecting rice diseases are brown spots and leaf blight. In the proposed work [8] authors applied YOLOv5 for identification of diseases and CNN was used for classification of diseases. For localization of diseases bounding box was used which shows the type of diseases. For the evaluation of work mAP was calculated as 69%, which was found to be satisfactory.

In the proposed work [9] YOLOv7 model has been proposed by the authors. YOLOv7 utilized MobileNetV3 technique for retrieval of selected features, decreasing parameters, and proper coordination in different parameters for improved accuracy. In the proposed work authors utilized large dataset of 3773 images for training of the model. The proposed work attained good accuracy of 92.3% and mAP at the rate of 0.5 was 93.7%.

For improvement in the growth of rice the early detection of diseases is compulsory. These days' deep learning techniques are playing an important role in detection of different type of diseases in plants. Authors in [10] have used updated YOLOv8 in which original Box Loss function was replaced by combination of EIoU loss and $\alpha\alpha$-IoU loss to enhance performance of the rice leaf disease identification system. The proposed technique worked in two stages using artificial intelligence. At first stage rice leaf with diseases are collected and then these images were segregated into blast leaf, leaf folder, and brown spot sets, respectively. After training of segregated data using YOLOv8 model it was deployed on IoT devices for identification of diseases. The proposed work was compared with YOLOv5 for validation of the results. The dataset of 3175 images was utilized out of which 2608 images were used for training, 326 images for validation and 241 images for testing.

The existing crop diseases detection methods are having some limitations. The techniques does not work well in situations like small size of plant, variations in type of plants, and image quality etc. In paper [11], authors applied YOLOv8n for the detection of diseases in rice leaves. Authors first worked on different parameters to understand the complete process of diseases detection. The aim of authors was to retrieve features from different types of images and then C2f module of YOLOv8 model was applied to improve the performance of work. The simple structure of YOLOv8n was replaced by bidirectional feature pyramid network to attain good results and reduce complexity. The IOU loss function was replaced by wise IOU to deliver better results. For evaluation of work, mAP at rate of 0.5 was calculated which was found to be satisfactory.

Modern agricultural such as hydroponic agriculture is also affected by diseases. For detection of diseases an automatic detection technique is required. The work in [12] involves the use of YOLOv8 and YOLOv9 models for detection of diseases. Dataset was taken from kaggle along with some images from hydroponic agriculture. The models were trained and test using accuracy, speed and robustness. The YOLOv9 performed better compared to YOLOv8 in terms of assessment metrics. YOLOv9 has taken less time compared to YOLOv8.

Authors in work applied improved YOLOv9 model for Northern Maize Leaf Blight (NLB). Authors in [13] updated loss function, neck network, and backbone network to get better features selection which in turn improves detection accuracy. The use of backbone's GhostConv made convolution operations easy while C3TR enhanced feature selection. Robustness was improved by revised loss function due to wise IOU. The neck's multi-scale fusion enhanced feature incorporation. For assessment of work accuracy and mAP was calculated. Accuracy was calculated as 86.1%, mAP at the rate 0.5 was also improved by 3% on NLB dataset.

Due to environmental degradation and pest uses there is noticeable reduction in corn crop production. Authors in [14] proposed YOLv9 model for detection of pest on corn crop. For better feature retrieval authors used Programmable Gradient Information (PGI) and the Generalized Efficient Layer Aggregation Network (GELAN). They also presented Multi-scale Spatial Pyramid Attention mechanism (MSPA) module. MSPA is able to retrieve features by catching classified information of multi-scale spatial information at a more detailed level using hierarchical residual connections. For the work they selected IP102 dataset which is widely used dataset. Dataset contains 3,480 images and used 11 distinct categories for training. For assessment of work, mAP is calculated which was found to be 96.3%. The proposed work was found to be cost effective and taken considerable time in pest detection. After literature survey some of the research gaps are listed as follow [15][16][17]:

i. A majority of the existing literature focuses only on individual variants of the YOLO series instead of undertaking comparative research for various generations of the state-of-the-art YOLO version.
ii. There is limited study of the development of architectural innovations in terms of YOLO models for agricultural disease detection.
iii. In addition, most related works only emphasize the precision, without taking appropriate considerations of the inference, computational complexity, and deployability.
iv. There is no standardized benchmarking framework available for the fair comparison of object detection models on corn leaf disease datasets.
v. Lack of in-depth analysis on the robustness of the model under changes in lighting conditions, occlusion, background noises, and varying orientations of the leaves.
vi. Lack of thorough investigation on failures and classification errors in YOLO variants for plant disease detection.
vii. Limited studies available to analyze the trade-off between light models and the precision achieved while deploying them in edge devices for precision agriculture.

3. Proposed Models

For training of the proposed models large dataset from Kaggle is chosen. Dataset named Corn or Maize leaf diseases has been selected. It is having 10046 images of corn leaves. Scientist or researchers are utilizing this dataset to monitor health of corn crop. 8790 images are used for training of the model, 419 images are used to validate the results, and 837 are used to test the models. Mentioned dataset contain images which as collected from Plant Village and PlantDoc dataset. The low quality images are removed from dataset so that model can be trained carefully. It has four classes in total. The class 0 is known as Common Rust, class 1 as Gray Leaf Spot, class 2 is Blight and class 3 is known as Healthy. Data augmentation techniques such as flip vertical, rotation between -15 degree to +5 degree, 90% clockwise rotation, shearing ±0 degree horizontal to ±15 degree vertical are applied on dataset as represented in Figure 1. In pre-processing auto-orientation and resize techniques are applicable on the dataset. In dataset 29988 annotations are given to the images as shown in Figure 2.

The dataset is labeled by authors using a complete manual process for drawing bounding boxes. This helped make sure the labels are of high quality. The open-source software LabelImg is used, and it creates labels that work with the YOLO model. People manually drew boxes around the parts of maize leaves that showed signs of disease, like blight, rust, and gray leaf spot. Each image is labeled with the correct disease type based on what the symptoms looked like and standard descriptions from agriculture. To keep the labels accurate and reduce differences between labelers, a specialist in plant diseases checked the labels. Also, a random group of images, about 10-15%, is checked again to make sure the boxes are drawn consistently and accurately. No automatic or partially automatic methods are used, so there is no risk of bias from models that had been trained before.

Implementation of the work has been done in Python on google colab. GPU of moderate RAM is used so batch size is chosen 32 which are found to be suitable for all the versions of YOLO model. Epoch of size 100 has been utilized for achieving better results. Confusion matrix evaluate the research work whether it is performing well or not. Confusion matrix works on the basis of parameters such as True Positive, True Negative, False Positive and False Negative. With the help of these parameters' accuracy, precision, recall, F1-score, mAP and time taken can be calculated.

3.1 YOLOv5 Architecture

YOLOv5 is enhanced over YOLOv4 in terms of architecture, training, and deployment. It gained importance due to the ease of use and its efficiency in object detection. CSPNet acts as backbone in YOLOv5 and it divide the feature map into two parts. The splitted parts are then merged through a cross-stage hierarchy. It utilizes the focus layer at the start of the network for dividing image into smaller patches. The focus layer decreases spatial dimensions while increasing the depth. Bottleneck CSP works on bottleneck layers with CSP networks to increase gradient flow and feature reuse. The neck network in YOLOv5 is accountable for combining features from various stages of the backbone to generate best feature demonstrations for prediction of objects. PANet in YOLOv5 act as neck, which improves feature pyramids by joining low and high-level feature maps. It permits the model to attain spatially information for best localization of objects. FPN layers aggregate features from several scales and allow the detection of objects of many sizes [19]. Figure 3 shows the outcome of model as confusion matrix. Confusion matrix is utilized to calculate the performance metrics such as accuracy, precision, recall, and F1-score etc.

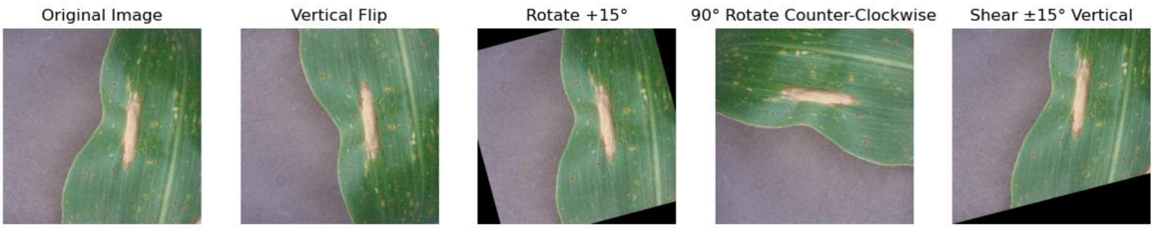

Fig. 1 *Augmentation of dataset*

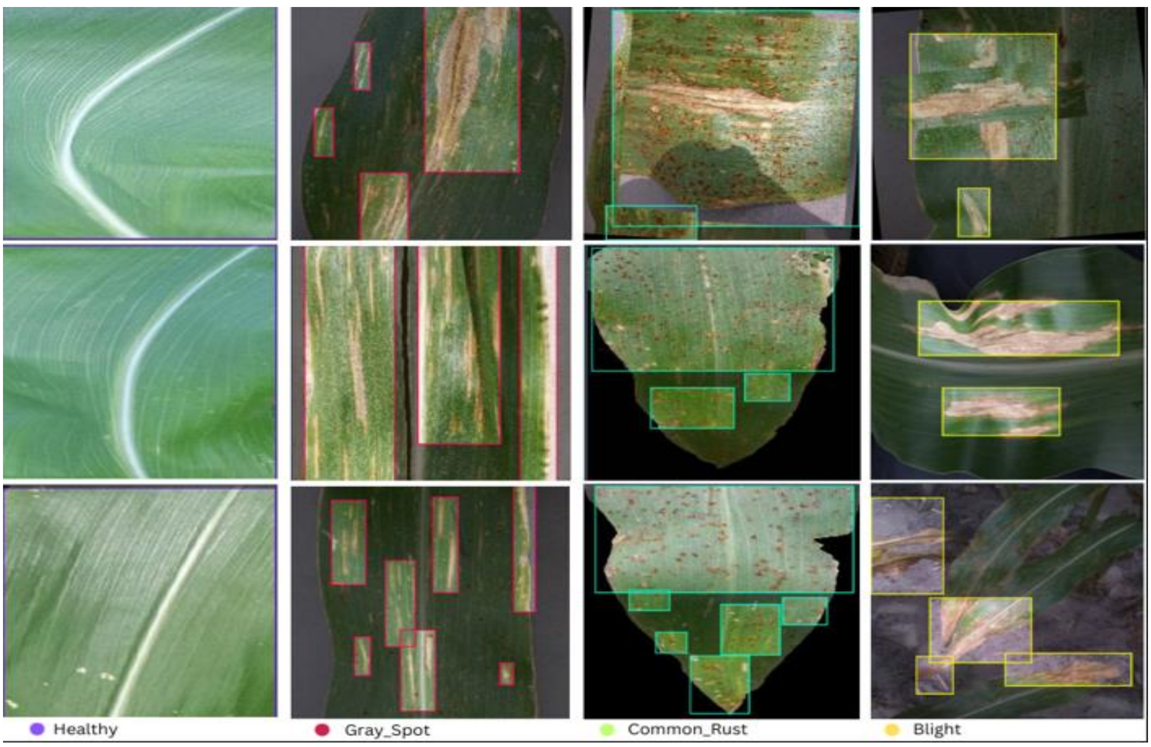

Fig. 2 *Annotations on dataset*

$$A = O_{object} + IoU_{prediction+truth} \quad (1)$$

Equation 1 represents the score of prediction of diseases using the rectangular. The O_{object} shows probability of existence of the object in rectangular box. $IoU_{prediction+truth}$ represents union over intersection between the predicted and ground truth boxes. Figure 4 represents the diseases detected in corn crops using YOLOv5 model. The presence of the diseases is indicated by the rectangular box labelled gray spot. This model utilizes the combination of FPN and PANet to generate a healthy multi-scale feature map. Its head is accountable for creation of the final predictions, containing bounding box coordinates, objectness scores, and class probabilities. It uses many activation functions for introducing non-linearity into the network. Along with leaky ReLU it utilizes sigmoid activation in last layer for identification of object and class probability. Mosaic, clipping, translation, scaling and many more data augmentation techniques are applied to improve the generalization of model [20]. It decreases problem of overfitting by softening the labels at the time of training. It is developed with lightweight and effective inference engine, allowing actual object prediction to have less powerful hardware including mobile devices. It could be installed in ONNX and TensorRT formats, further enhancing it for utilization in various environments. YOLOv5 has many versions such as small, medium, large and extra-large. It uses post processing techniques such as Non-Maximum Suppression (NMS) and IoU thresholding for accurate detection of object [21].

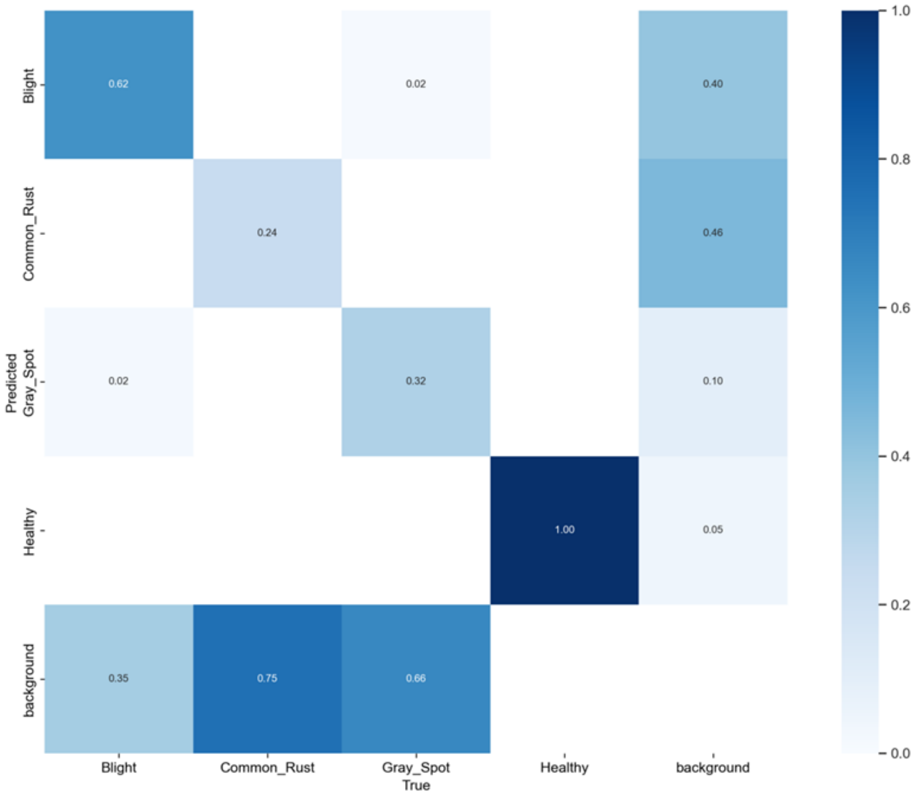

Fig. 3 *Confusion matrix normalized*

3.2 YOLOv6 Architecture

It is another addition in YOLO which leads to accurate object detection in real-time. It has improved accuracy and efficiency over its older versions. It has included several advancements in architecture and optimized the outcome of the model in terms of time and cost. YOLOv6 added the concept of new backbone known as efficientRep which help in retrieval of selected features maintaining speed and accuracy both. It has RepVGG in its backbone which provides a trade-off between training efficiency and speed by parameterization techniques. RepVGG blocks are made up of multiple branches including identity mapping and convolutions layers [22].

These branches are combined in a single convolution layer at the time of inference and speed up the process of detection. It also contains CSPNet for division of feature maps and combines it in a cross-stage style [23]. Figure 5 represents different labels on the given dataset. The dataset carries different categories of data such as blight, common rust, Grey_spot, and health. Equation 2 introduces the use of loss function in YOLOv6. Loss function is the sum of three types of losses as shown in equation also. The loss function optimizes bounding box regression, object confidence, and prediction of class [24].

$$Loss_{Total} = Loss_{box} + Loss_{object} + Loss_{class} \quad (2)$$

Figure 6 represents outcome of models in terms of detection of diseases in corn crops. Mainly two types of diseases are identified, which are gray spot and blight. Due to this combination computational cost is reduced and improves the learning capacity of the model. The neck bone of YOLOv6 efficiently aggregate selected features from different layers for robust variable size object detection in real-time. PANet improve the localization ability by aggregating low-level spatial data with high level semantic data. The FPN layer help in generating a feature pyramid which in turn provides efficiency for detection of variable size objects. Its design is lightweight, which confirms efficiency of computation and speed of detection.

It does proper classification of objects with bounding box regression. It works on non-linearity of the network due to the involvement of SiLU and Sigmoid activation function. Mosaic technique does data augmentation for generalization of the model. It uses 16-bit and 32-bit floating-point operations to improve training speed and decrease memory utilization without cooperating model accuracy. YOLOv6 could be transferred to setups like TensorRT and ONNX, which are improved for deployment on various hardware platforms [25].

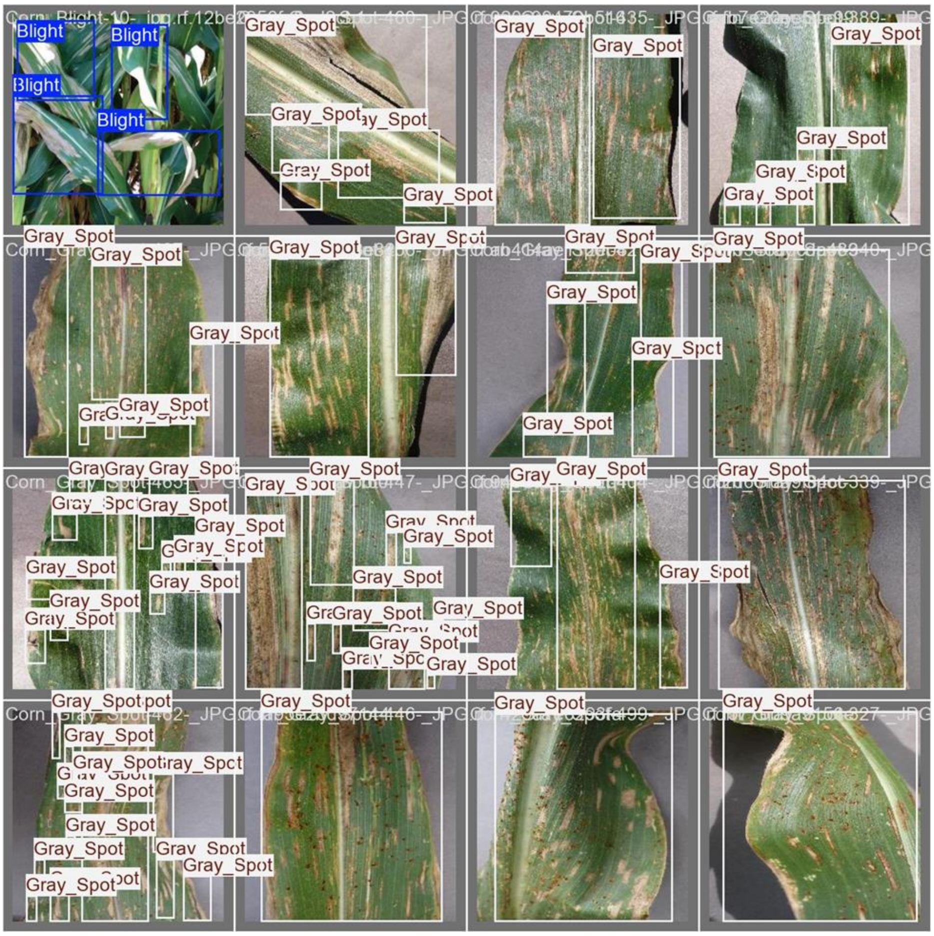

Fig. 4 *Detection of corn diseases*

3.3 YOLOv8 Architecture

It is the latest version of YOLO, developed for high speed and accuracy. It is designed on the strong points of previous versions with the addition in its architecture as shown in Figure 7. YOLOv8 has strong backbone and responsible for retrieving selected components of the input image. YOLOv8 utilizes updated version of CSPNet and responsible for splitting the feature map into various parts. Then cross stage merging is done to decrease the computational cost. It improves the architecture by optimizing how features are divided and concatenated to each other. Its backbone contains a deep stem block with many convolutional layers with small kernels of 3x3, that helps in taking fine details in the input image [17].

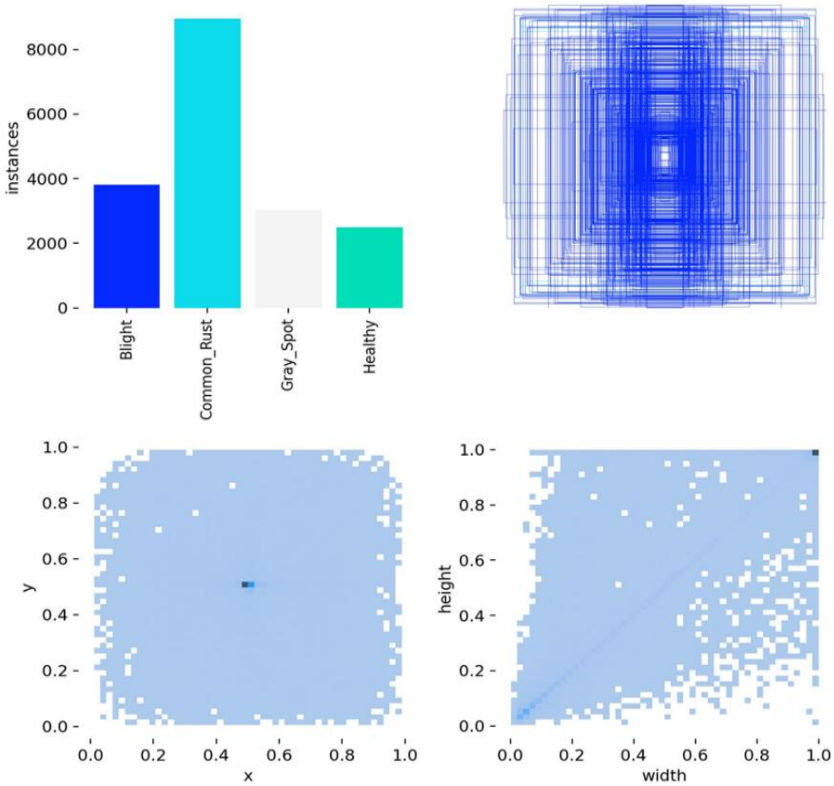

Fig. 5 *Labels in dataset*

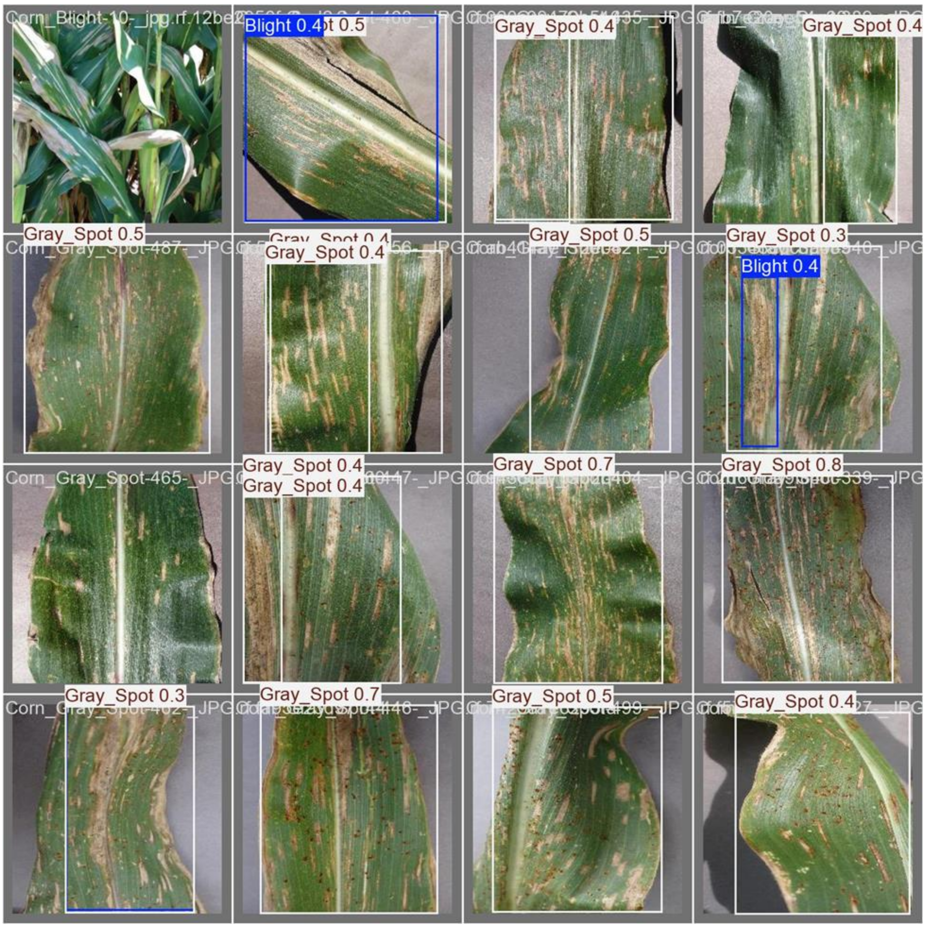

Fig. 6 *Diseases detection*

Int. Journal of Integrated Engineering Vol. 18 No. 4 (2026) p. 1-21

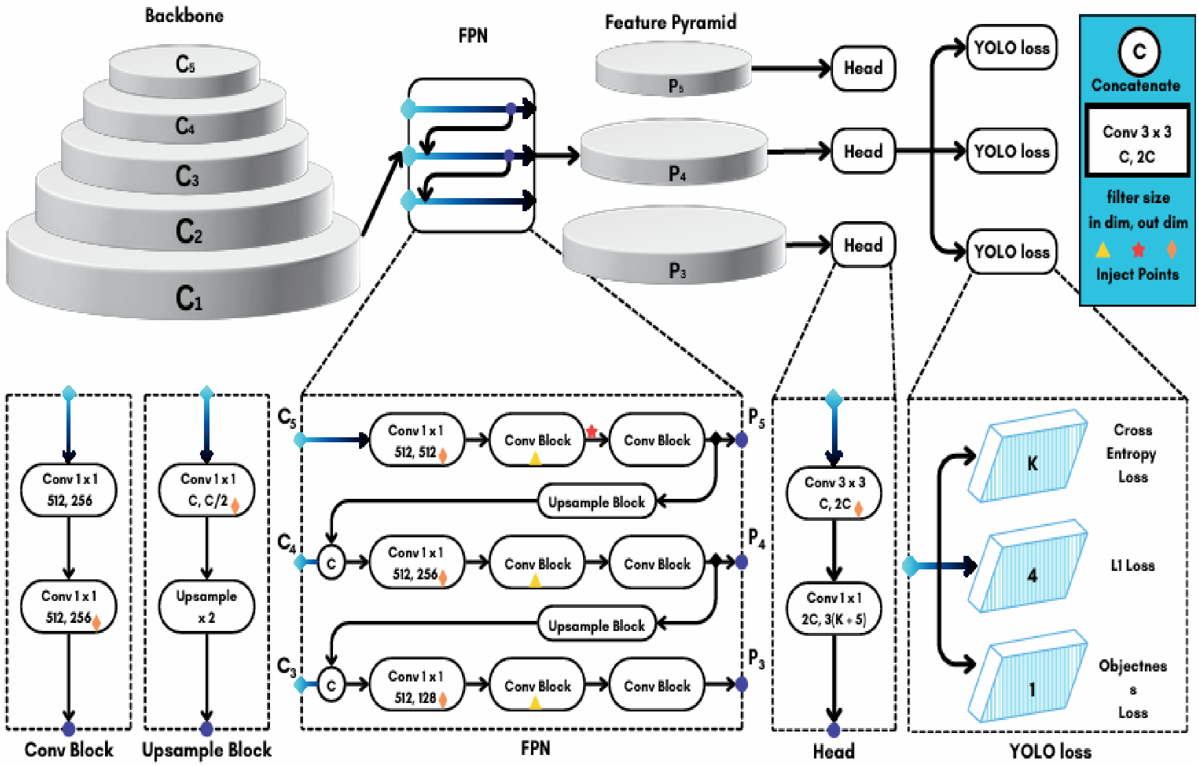

Fig. 7 *Architecture of YOLOv8 [18]*

The neck bone utilizes enhanced PANet which in turn optimizes the process of path aggregation. It allows for fusion of selected features of input image at various levels in network. In this way YOLv8 provides rich information for detection of the real-time objects. FPN combine the selected features at various stages, these features are variable in size as well. The head backbone provides final detection of the objects using bounding box and object score. It utilizes CIoU(Complete Intersection over Union) loss for calculating bounding box regression as shown in Equation 3. Where D is the distance between centers of predicted and ground truth boxes, c is diagonal length of smallest enclosing box, v is aspect ratio, and α is balancing parameter for aspect ratio [26].

$$L_{box} = 1 - IoU + \frac{D^2\left(b_{predicted}, b_{truth}\right)}{c^2} + \alpha v \tag{3}$$

It also uses decoupled heads for separate branches for better performance. It also carry dynamic head mechanism which is used at the time of training for concentrating on the most challenging parts of job like identifying small real-time objects and handling complex backgrounds conditions. It uses auxiliary blocks which give extra supervision at the time of training the dataset. During training it differentiate between the different classes and help in providing extra gradients. It leads to better localization of objects with bounding box. It uses activation function for adding more learning and generalization to the model. Figure 8 represents Labels Correlogram for YOLOv8, it analyses the distribution and co-occurrence of object classes in dataset. Correlogram can easily detect the patterns in class labeling which may affect model performance [27].

SiLU has efficiency for providing smoother inclines and enhances training dynamics. Leaky ReLU is utilized for specific parts of the network to avoid the dying ReLU problem, specifically in case of deeper layers. It do dynamic data augmentation using techniques like Mosaic, MixUp, and CutMix, for enhancing the variety in training data, leading to improved simplification and robustness. SAT enhance the model's robustness against attacks by presenting small disquiets in input images during training, making the model more robust in real-world scenarios. YOLOv8 models could be transferred to ONNX and TensorRT formats by permitting optimized deployment across various hardware platforms, containing edge devices and mobile processors. It can detect variable size object detection including nano, small, medium, and large [28].

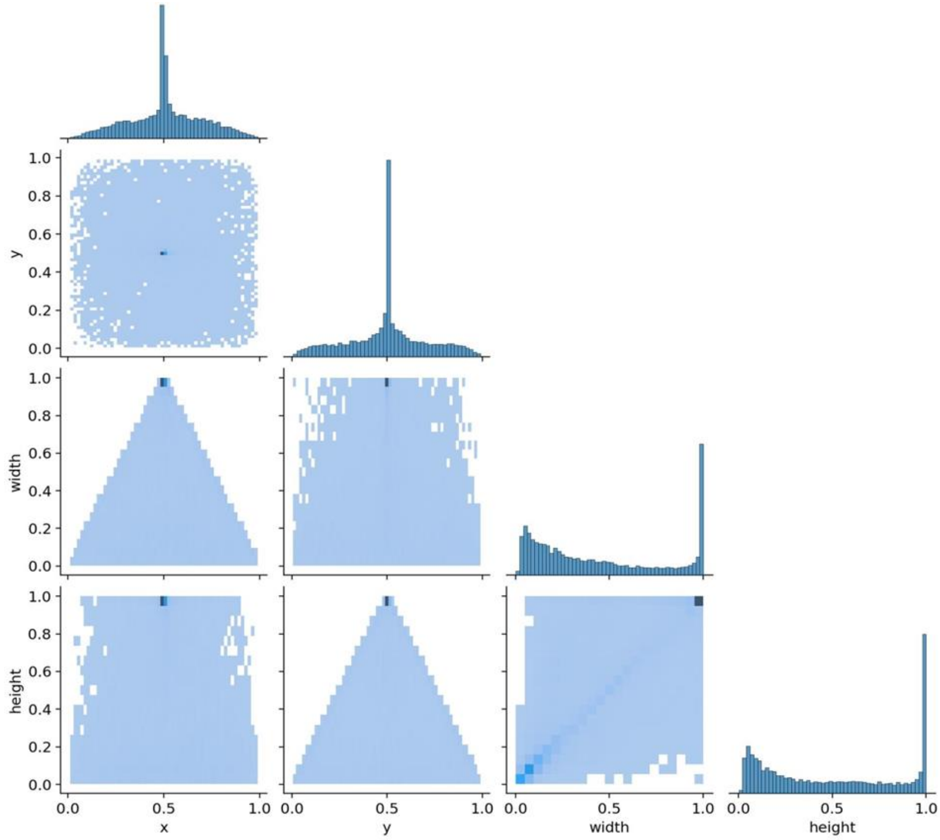

Fig. 8 *Labels correlogram*

3.4 YOLOv9 Architecture

YOLOv9 is the advanced version of YOLOv8 which incorporate backbones such as CSP-Darknet or CSPNet-based architectures. The neck in YOLO architecture is normally responsible for combining and integrating features from various phases for multi-scale detection. FPN and PANet are general architectures in YOLO versions but YOLOv9 introduces revised versions of these architectures to enhance feature synthesis efficiency. The head network makes bounding box and do class prediction using anchor free technique for simplification and enhanced speed. The loss function in YOLO9 is stronger to manage imbalanced data and edge cases [29].

$$s_j = s_j \times \exp\left(-\frac{(IoU(b_i, b_j)^2}{\sigma}\right) \tag{4}$$

The transformer and attention mechanisms in it focus on specific parts of an image to find more accurate results. It also does optimization in quantization, cropping, and mixed-precision training. For improvements in post preprocessing it apply Soft-NMS, Weighted-NMS, or clustering-based methods as shown in Equation 4. The b_i and b_j are bounding box coordinates; s_i represents sorted predictions, and σ control rate of reduction in score. In this way YOLOv9 perform better than the other existing versions of YOLO [12]. The detailed architecture of YOLOv9 is shown in Figure 9. Figure 10 depicts detection of healthy parts of corn crops with the help of rectangular boxes.

3.5 YOLOv10 Architecture

YOLOv10 model has hybrid backbone which is combination of Convolution Neural Network (CNN) and transformers. Vision transformers has important role in image processing which integrate CNN to get the enhanced results. In complex images it finds both local and global dependencies, improving accuracy of the model. Swim transformers used in backbone provide high accuracy. It uses FPN as represented by Equation 5 and PANet as represented by Equation 6 for aggregation of features on large scale. It also uses AFP which dynamically do feature integration based on image content. Self-attention in YOLOv10 is incorporated to find the region of interest from the input image.

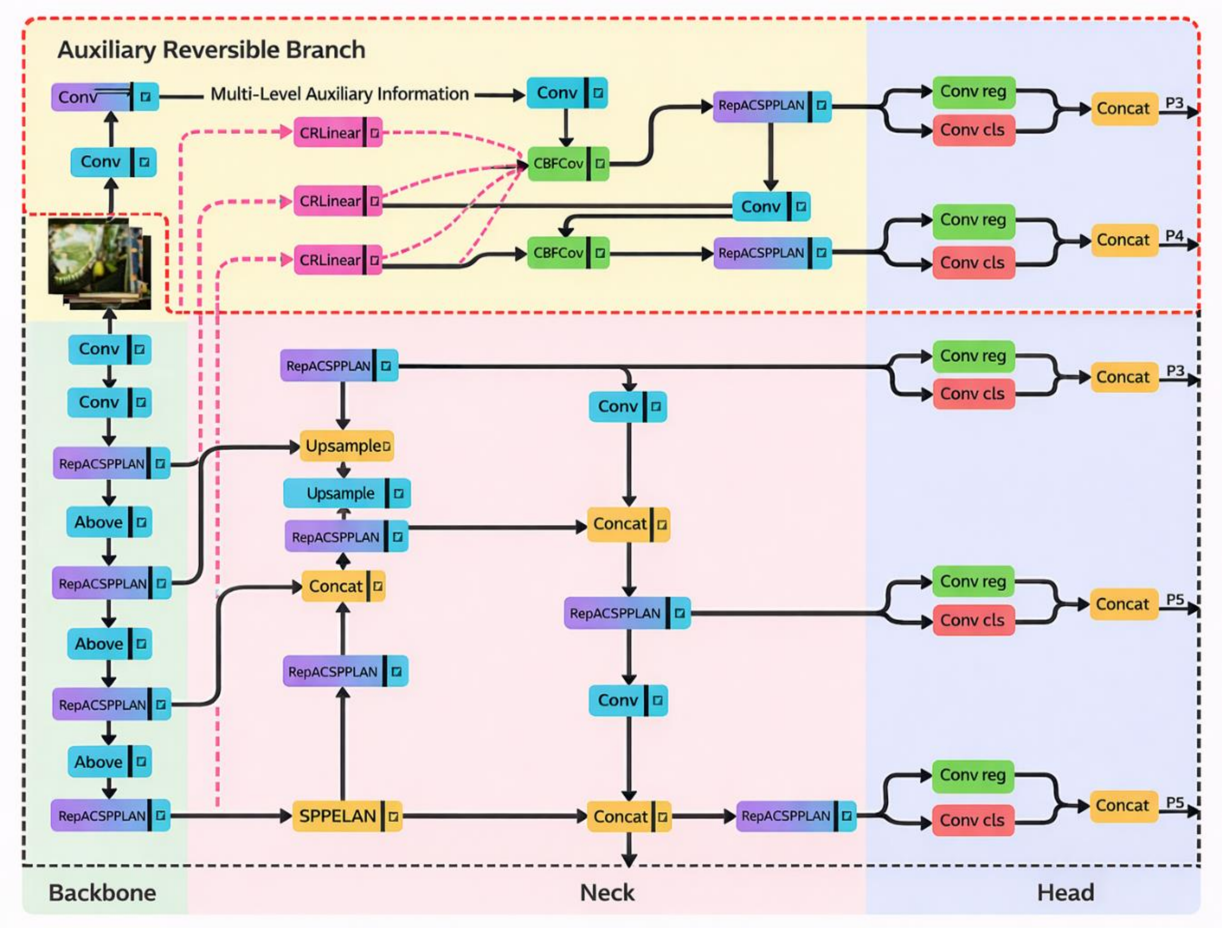

Fig. 9 *YOLOv9 architecture*

$$P_l = W_1 \times X_l + Upsample(P_{l+1}) \quad (5)$$

Where P_l is the feature at level l, $W_1 \times X_l$ shows convolution used at feature X_l, $Upsample(P_{l+1})$ scale up the higher-level feature map P_{l+1}. W_2 is used for learnable weight and $F_{bottom\text{-}up}$ represents features from bottom-up pathways.

$$F_{out} = W_2 \times \left(Concat\left(P_l, F_{bottom-up}\right)\right) \quad (6)$$

The anchor-free method in YOLOv10 dynamically adjusts the features of image. It enhances the training speed and decrease the computational cost. It utilizes loss function to improve performance of model by working objects, classes and occlusions. It utilizes self-learning on labeled small dataset. It also does post processing to decrease over-suppression on overlapping bounding boxes [19]. Figure 11 represents architecture of YOLOv10, it carry various layers. Figure 12 shows confusion matrix for the mentioned version. Confusion matrix is further utilized to calculate various performance metrics such as accuracy, precision, recall, F1-score etc.

Fig. 10 *Detection using YOLOv9*

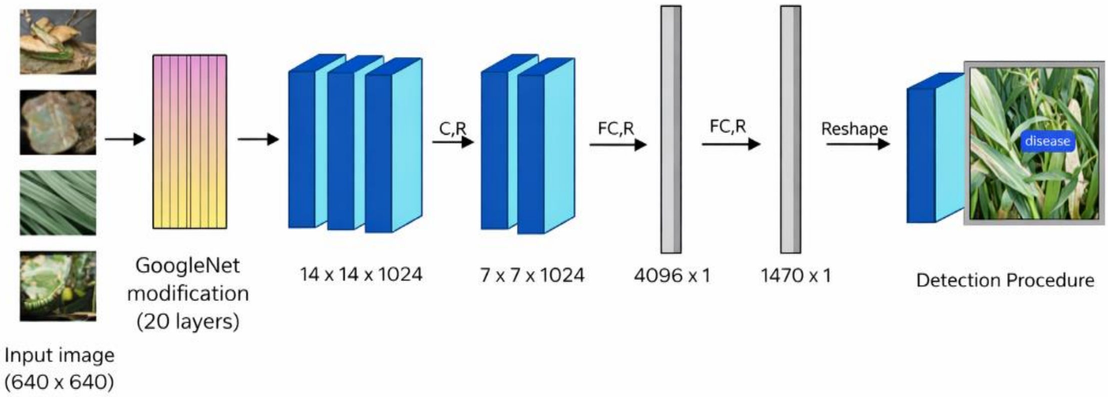

Fig. 11 *YOLOv10 architecture*

3.6 YOLOv11 Architecture

YOLOv11 overcome the demerits of YOLOv10 and improve the outcome. It has unified transformer-CNN backbone which could dynamically change between convolutional and transformer processing. The hybrid structure of YOLOv11 offer capabilities of CNNs for local feature retrieval along with the transformers' strength in capturing long-range dependencies. It also integrates ConvNextV3 as new backbone designed explicitly for object prediction. It uses Dynamic Multi-Level Feature Fusion (MLFF) as represented by Equation 7, which adds advancement over traditional methods such as FPN or PANet. It utilizes MLFF for better feature selection based on the complexity and content of the input image. Cross-attention in neck node enables the model to access contextual information. Cross-attention allows handling complex scenes with unique features [2]. Figure 13 represent the labeled diseases in corn crop. Prediction in Figure 13 are done using YOLOv11, mainly two diseases are found which common rust and gray spot.

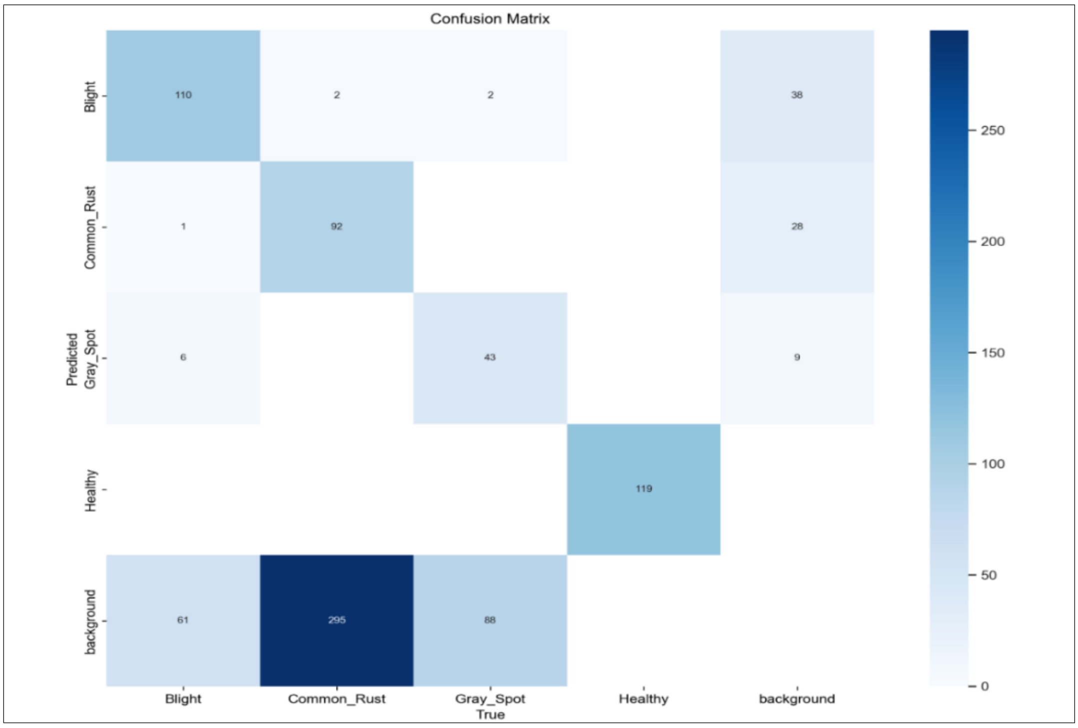

Fig. 12 *Confusion matrix by YOLOv10*

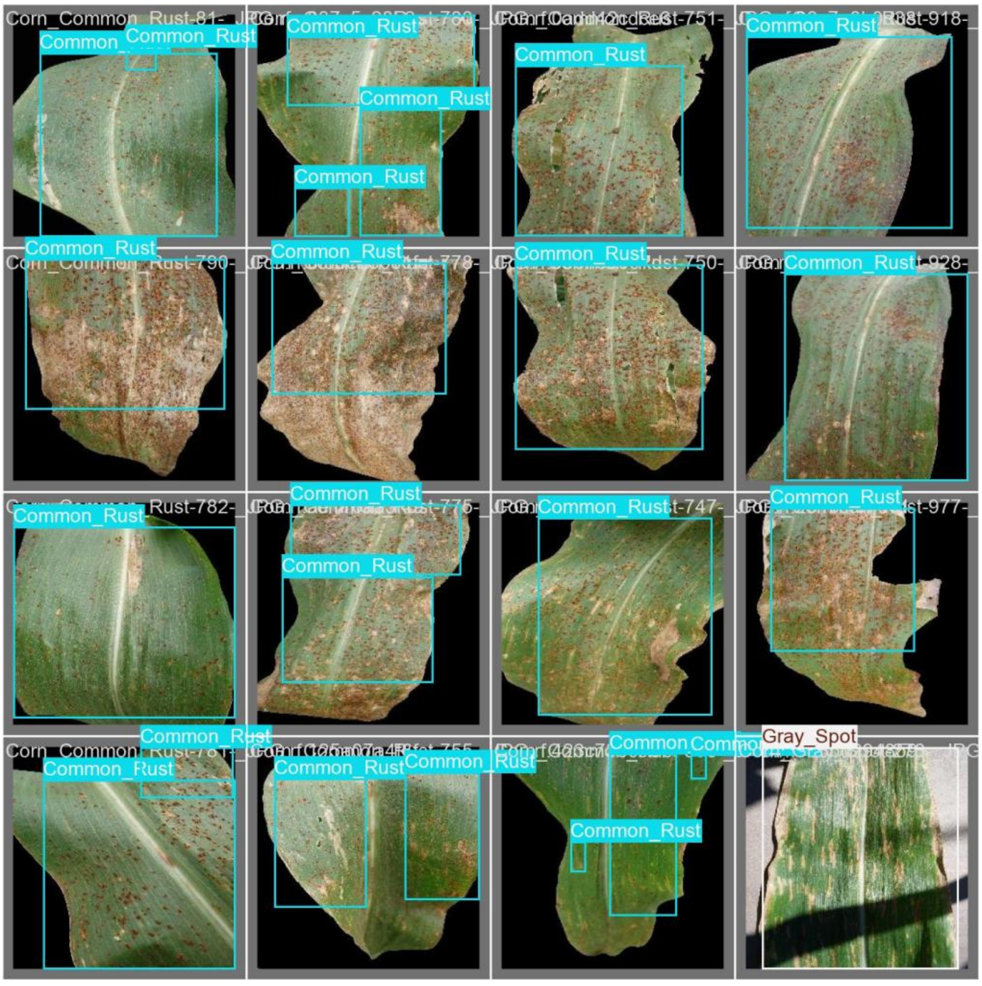

Fig. 13 *Classification of corn diseases*

$$D_{Multiscale} = \sum_{j=1}^{m} W_j \times F_j \tag{7}$$

Where $D_{Multiscale}$ is multiscale detection of leaf diseases using FPN or PANet, F_j is used for feature maps on various layers, W_j is used for learnable weights. It sometimes utilizes dynamic heads for various scales depending upon the size of object. It uses set-based detection for simplification of bounding box and decreases post-processing requirements. It uses unified loss function for dynamical balancing of classification, localization, and IoU-based loss. Contrastive loss gives objects distinction for better feature selection resulting into robust feature prediction in challenging conditions. SCAF feature in YOLOv11 integrate spatial and channel attention mechanism for finding where and what to pay attention to in an image. An LNMS technique, in it utilizes learned approach for training of prediction resulting into decreased load and more accurate.

4. Results and Discussion

For training of the proposed models large dataset from Kaggle is chosen. Dataset named Corn or Maize leaf diseases has been selected. It is having 10046 images of corn leaves with four classes. Scientist or researchers are utilizing this dataset to monitor health of corn crop. 8790 images are used for training of the model, 419 images are used to validate the results, and 837 are used to test the models. Data augmentation techniques such as flip vertical, rotation between -15 degree to +15-degree, 90% clockwise rotation, shearing ±0 degree horizontal to ±15-degree vertical are applied on dataset.

The YOLO 6 versions are implemented in python language to predict the corn leaf diseases. For the implementation of work GPU has been utilized. Authors have taken 100 epochs to train the models and batch size is fixed at 32. Table 2 shows the performance metrics achieved for all the proposed versions of YOLO. YOLOv9 outperformed among the proposed versions. For the assessment of the work, authors have utilized mAP at different learning rates, precision, recall metrics, and time taken.

The training conditions are kept consistent across all versions of YOLO, ranging from YOLOv5 to YOLOv11. In more detail, the batch sizes across all the models are set to 32 during the training process, while the epoch count is set to 100. Additionally, each version of the model is trained in an identical GPU environment. The starting learning rate was identical across all versions. By standardizing these settings, the differences in the performance of YOLO versions can be linked mainly to architectural enhancements, not training parameter variations. In addition, it is important to note that in cases where any of the models would have necessitated version-specific default changes, these were maintained with default values as recommended while ensuring that training parameters are consistent.

In the work standardized pre-processing approach is applied to guarantee the quality of data for all YOLO versions. The corn leaf images, depicting various classes of diseases, as well as healthy images, are gathered from field and public image sources. Irrelevant images, blurred images, duplicate images, and images with improper labels are discarded. Infected areas of diseased leaves are marked by bounding boxes and saved as YOLO format, which includes a class ID and the normalized coordinates for x-centre, y-centre, width, and height of the boxes. The images are normalized to a fixed dimension, typically 640 x 640 pixels, by resizing images while maintaining the aspect ratio, and pixel values normalized to [0,1].

mAP@50 and mAP@50–95 are two different measures for evaluating the performance of the model on the test dataset, which indicate the level of strictness and the degree of comprehensiveness of these two metrics. The procedure for calculating mAP@50 is to determine the average precision using a single threshold value, which is 0.5. This means that if the intersection over union between the bounding boxes of the detected image and the ground truth image is greater than 0.5, i.e., if there is at least 50% overlap, the bounding boxes are said to correctly detect the object. On the other hand, the procedure for calculating mAP@50–95 is to determine the average precision at various intersection over union threshold values, which range from 0.5 to 0.95 with steps of 0.05. Therefore, it appears that mAP@50–95 is a more comprehensive measure than mAP@50. YOLOv9 at the rate of 50 performed best among the proposed versions and YOLOv8 performed lowest which is also represented in Figure 14.

Figure 15 shows the mAP@50–95 comparison of YOLO versions for corn leaf disease detection across 100 training epochs. All models exhibit sharp fluctuations during the initial epochs, followed by steady improvement and stabilization after approximately 30–40 epochs. YOLOv9 achieves the highest and most consistent mAP@50–95 values (around 0.38–0.39), closely followed by YOLOv11 and YOLOv8. YOLOv6 and YOLOv5 show comparatively lower but stable performance, while YOLOv10 demonstrates slightly higher variability in early training. Overall, YOLOv9 provides the best overall localization and detection performance among the compared models.

Table 2 *Performance metrics achieved*

Performance Metric	YOLOv5	YOLOv6	YOLOv8	YOLOv9	YOLOv10	YOLOv11
Accuracy	73	69	70	74	73	72
mAP@50	56.39	54.71	55.91	57.45	56.01	56.63
mAP@50-95	38.32	38.28	38.22	39.07	38.45	38.69
Precision	68.96	68.14	69.58	71.22	67.83	68.72
Recall	56.94	56.84	56.73	57.42	57.09	57.35
Time taken (Minutes)	100.58	67.01	75.91	158.20	98.76	69.70

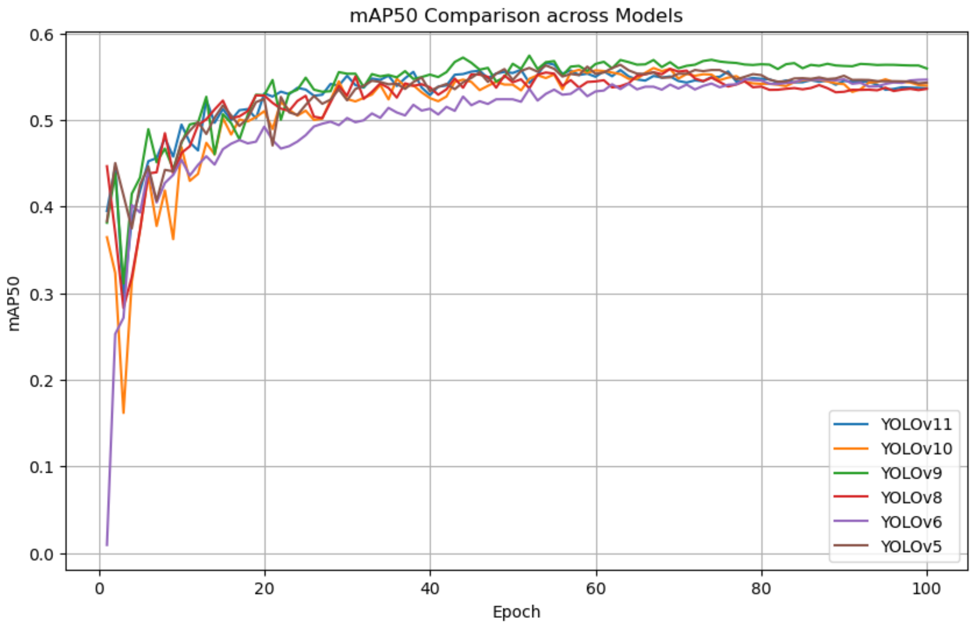

Fig. 14 *mAP50 achieved*

The mAP measures how well the YOLO model could predict and localize the object in the input image. It is also known as precision-recall curve which could be generated by changing the threshold. Average precision for a class is known as AUC. It can be calculated by Equation 8 as in paper [20] [21].

$$\text{mAP} = \frac{\sum AP_{class_i}}{Number\ of\ classes} \tag{8}$$

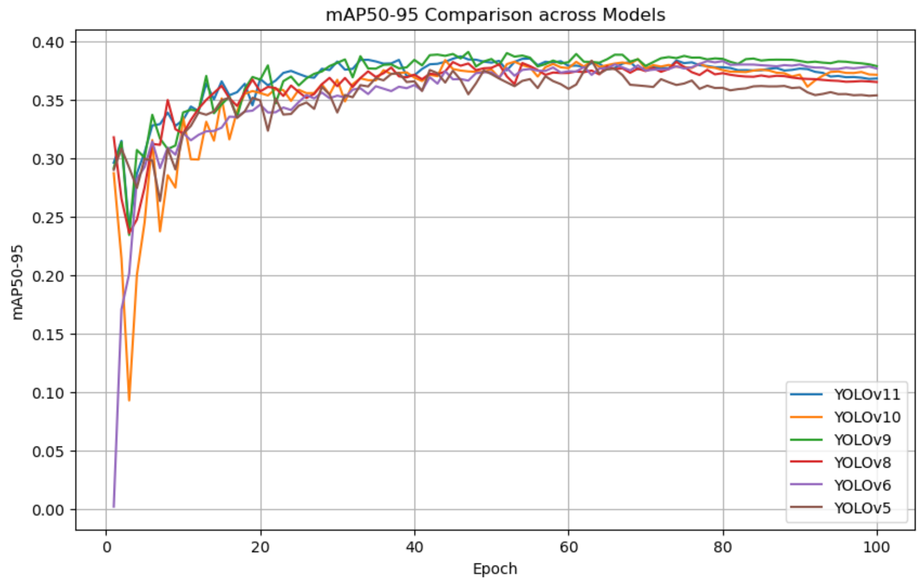

Fig. 15 *mAP50-95 achieved*

It represents the ratio of accurately predicted positive outcomes divided by the whole of accurately predicted positive and false positive outcomes. The mathematical formula for precision is given in Equation 9 as in [22].

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \tag{9}$$

As shown in Figure 16 YOLOv9 achieved highest precision value of 71.22% and YOLOv10 achieved lowest precision value which is 57.42%. Figure 1 presents the precision performance of YOLOv5, YOLOv6, YOLOv8, YOLOv9, YOLOv10, and YOLOv11 models for corn leaf disease detection with the training of 100 epochs. In Figure 16, we can see that the precision performance of six models increases dramatically during the beginning. Later on, the precision performance of these models converges. It can be observed that the precision performance of these models begins to settle down approximately after 30-40 epochs. Among these models, YOLOv9 produces the highest precision value.

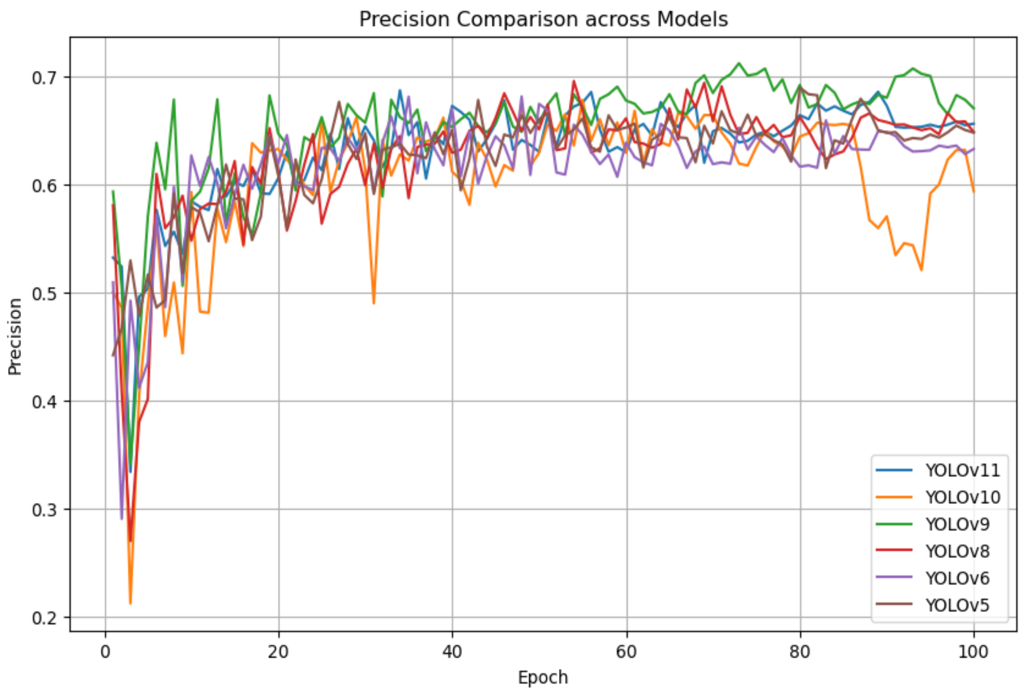

Fig. 16 *Precision attained*

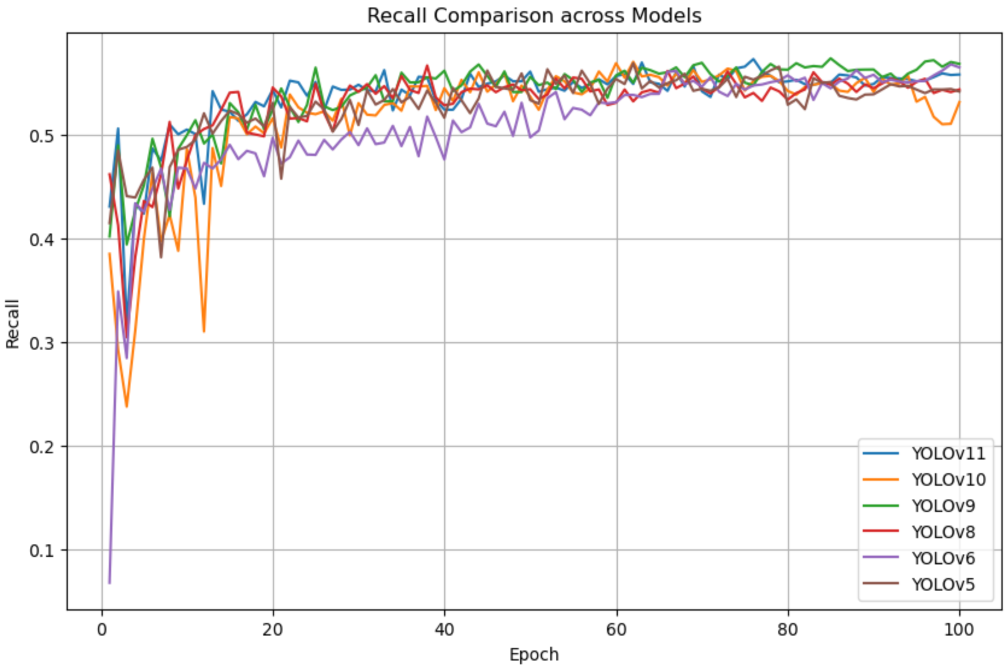

Fig. 17 *Recall attained*

Recall, which assesses a classification model to capture all relevant instances of a given class, is also known as sensitivity or true positive rate. It is measured as the ratio of real positives to the whole of false negatives and true positives. The mathematical formula for recall is represented by Equation 10 as in paper [30].

$$\text{Recall} = \frac{\text{True Positive}}{\text{False Positive} + \text{False Negative}} \tag{10}$$

Figure 17 describes the recall comparison between YOLO versions for detecting corn leaf disease, where all these models are trained for 100 epochs. It is apparent that there are significant fluctuations during the initial epochs for all these models, but after 30-40 epochs, these models show steady improvement. Among these models, YOLOv11 and YOLOv9 show the best recall values, i.e., 0.56/0.57, for detecting true disease. YOLOv8 and YOLOv5 show consistent performance, while YOLOv6 shows steady improvement, and YOLOv10 shows fluctuations during initial epochs but converges at late epochs. It is evident that YOLOv9 shows high recall compared to other models.

Accuracy as represented in Equation 11 is the ratio of correctly predicted samples to the total number of samples assessed. It is counted as best metric for assessment of classification problems.

$$\text{Accuracy} = \frac{\text{Count of correct predictions}}{\text{Total count of Predictions}} \tag{11}$$

Figure 18 describes the accuracy comparison of YOLO versions during the prediction of corn leaf disease for 100 epoch training data. According to the figure, we can see that all the models have shown fluctuations until the 20th epoch and then have started to become stable. From the given accuracy comparison, we can identify that the YOLOv9 and YOLOv10 models have shown high accuracy values, mostly even above 70, and have been closely associated with 73 and 74. The accuracy of YOLOv11 and YOLOv5 models is stable and high, even at 70, while the accuracy of YOLOv6 is quite low for all epochs.

Figure 19 represents the time taken by YOLO versions to train 100 epochs. YOLOv9 has taken maximum time which is 158.20 minutes and YOLOv6 has taken minimum time of 67.01 minutes. It has been observed that with the higher attainment of the performance metrics the time also increased. However, YOLOv6 model and the YOLOv11 model require lower training times than the other models. The YOLOv8 model shows moderate training time.

Although YOLOv9 presented the maximum computational cost for training, it is found to be the best model due to high performance in terms of performance metrics. These metrics p are considered essential performance metrics in disease detection. In terms of an agricultural scenario related to disease detection in crops such as corn, the accuracy of detection compared to extra training time is more important. Training is done in an offline scenario, whereas in a real-time scenario, inference using these models would be done. Therefore, apart from YOLOv9, all other models show comparable performance. Although YOLOv9 shows improvement in terms of localization and classification using the highest achieved mAP, it also resulted in even higher precision and recall, ensuring less false positives and false negatives, which could otherwise lead to misdiagnosis and further damage to the crop. Along with lots of achievements by the YOLO versions there are still some limitations which are given as follow:

i. Moderate accuracy ceiling (~73-74%), suggesting that there is little room for further improvement in model performance for the newer versions of YOLO.
ii. Low recall (~57%), resulting in false negatives or undetected diseases.
iii. Class imbalance issue, which affects the performance of minority class disease detection.
iv. High inter-class similarity among corn leaf diseases, resulting in misclassifications.
v. Variability in performance across epochs, suggesting instabilities in the training process.
vi. Increased computational requirements and training time for newer versions of YOLO.
vii. Lack of external validation to assess generalization performance.
viii. Sensitivity to real-world conditions.
ix. Lack of statistical significance testing.
x. Lack of explainability or attention map analysis to understand model behavior.
xi. Possible noise in annotations, which can affect detection accuracy.
xii. Limited augmentation strategy, which can affect robustness.

5. Conclusion

The study focused on versions of YOLO including YOLOv5, YOLO6, YOLOv8, YOLOv9, YOLOv10 and YOLOv11 for corn leaf disease detection. The experimental results verified and validated on the large Kaggle dataset. YOLOv9 has achieved accuracy of 74%, 39.07% mAP@50-95, mAP@50 is 57.45%, precision of 71.22%, recall value of 57.42%, and time taken to train 100 epochs is 158.20 minutes. YOLOv9 performed best as evident from experimental results and YOLOv10 performed lowest for detection of corn leaf diseases. The designs of these models help farmers in identification of corn leaf diseases and improve the production of corn. The early detection of corn leaf diseases reduces the cost; provide proper use of resources, and minimum use of plasticizer to avoid degradation of environment. Consuming healthy corn by the people leads to their good health as well. In future the models can be trained on multiple datasets for real-time detection of corn leaf diseases.

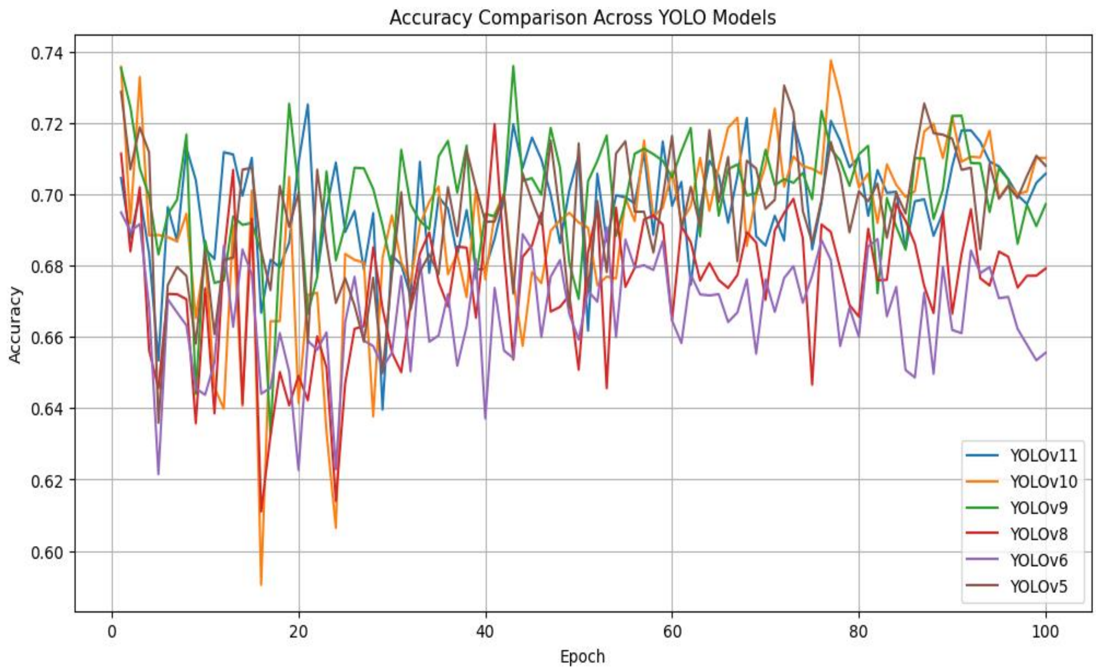

Fig. 18 *Accuracy comparison*

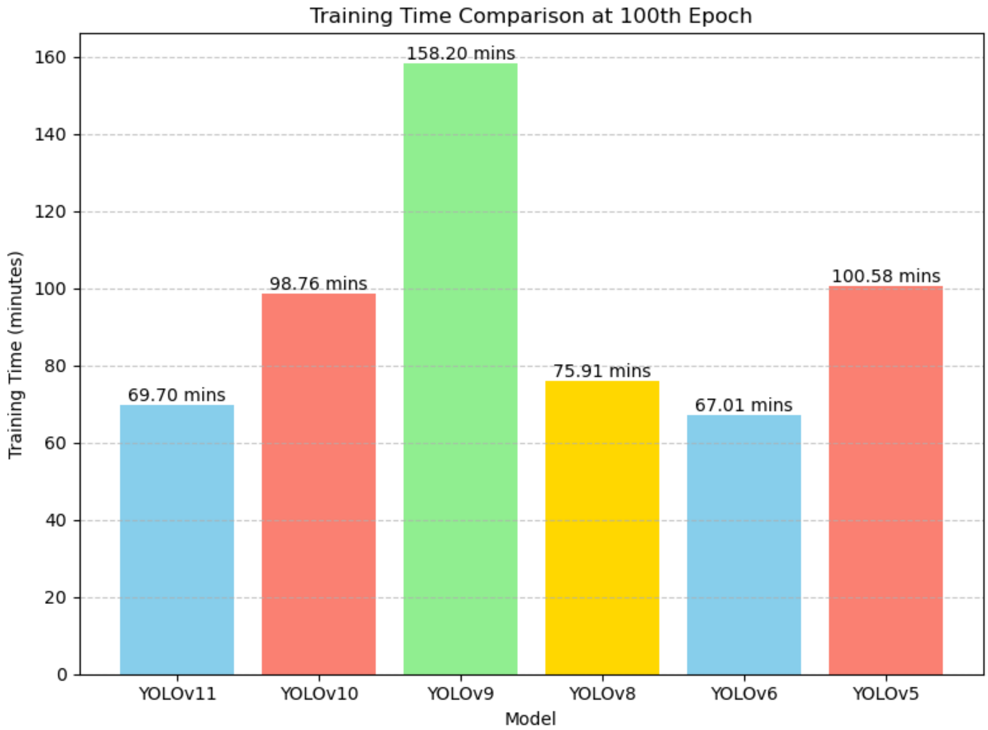

Fig. 19 *Training time*

Acknowledgement

The authors conducted the research independently, without receiving any support from the university or any other agency.

Conflict of Interest

Authors declare that there is no conflict of interests regarding the publication of the paper.

Author Contribution

The authors are responsible for the study conception, research design, data collection, data analysis, result interpretation and manuscript drafting.

References

[1] R. Li et al., "Lightweight Network for Corn Leaf Disease Identification Based on Improved YOLO v8s," Agriculture (Switzerland), vol. 14, no. 2, Feb. 2024, doi: 10.3390/agriculture14020220.

[2] R. Sapkota, Z. Meng, M. Churuvija, X. Du, Z. Ma, and M. Karkee, "Comprehensive Performance Evaluation of YOLO11, YOLOv10, YOLOv9 and YOLOv8 on Detecting and Counting Fruitlet in Complex Orchard Environments," Jul. 2024, [Online]. Available: http://arxiv.org/abs/2407.12040

[3] V. S. Kumar, M. Jaganathan, A. Viswanathan, M. Umamaheswari, and J. Vignesh, "Rice leaf disease detection based on bidirectional feature attention pyramid network with YOLO v5 model," Environ Res Commun, vol. 5, no. 6, Jun. 2023, doi: 10.1088/2515-7620/acdece.

[4] Y. S. J. V. MK Agbulos, "Identification of leaf blast and brown spot diseases on rice leaf with YOLO algorithm," in 2021 IEEE 7th International Conference on Control Science and Systems Engineering (ICCSSE), 2021.

[5] Faruq Aziz; Ferda Ernawan; Mohammad Fakhreldin; Prajanto Wahyu Adi, "YOLO Network-Based for Detection of Rice Leaf Disease," in 2023 International Conference on Information Technology Research and Innovation (ICITRI), 2023.

[6] M. Juman Jhatial et al., "Deep Learning-Based Rice Leaf Diseases Detection Using Yolov5," vol. 6, no. 1.

[7] M. E. Haque, A. Rahman, I. Junaeid, S. U. Hoque, and M. Paul, "Rice Leaf Disease Classification and Detection Using YOLOv5," Sep. 2022, [Online]. Available: http://arxiv.org/abs/2209.01579

[8] N. Chitraningrum et al., “Comparison Study of Corn Leaf Disease Detection based on Deep Learning YOLO-v5 and YOLO-v8,” Journal of Engineering and Technological Sciences, vol. 56, no. 1, pp. 61–70, Feb. 2024, doi: 10.5614/j.eng.technol.sci.2024.56.1.5.

[9] L. Jia et al., “MobileNet-CA-YOLO: An Improved YOLOv7 Based on the MobileNetV3 and Attention Mechanism for Rice Pests and Diseases Detection,” Agriculture (Switzerland), vol. 13, no. 7, Jul. 2023, doi: 10.3390/agriculture13071285.

[10] D. C. Trinh, A. T. Mac, K. G. Dang, H. T. Nguyen, H. T. Nguyen, and T. D. Bui, “Alpha-EIOU-YOLOv8: An Improved Algorithm for Rice Leaf Disease Detection,” AgriEngineering, vol. 6, no. 1, pp. 302–317, Mar. 2024, doi: 10.3390/agriengineering6010018.

[11] J. Y. X. Z. B. Z. & Z. S. Yu Lu, “YOLOv8-Rice: a rice leaf disease detection model based on YOLOv8,” in Paddy Water Environ 22, 695–710 (2024). https://doi.org/10.1007/s10333-024-00990-w, 2023.

[12] A. Tripathi, V. Gohokar, and R. Kute, “Comparative Analysis of YOLOv8 and YOLOv9 Models for Real-Time Plant Disease Detection in Hydroponics,” Engineering, Technology and Applied Science Research, vol. 14, no. 5, pp. 17269–17275, Oct. 2024, doi: 10.48084/etasr.8301.

[13] Kaustubh Gharat; Harshita Jogi; Kaustubh Gode; Kiran Talele; Sujata Kulkarni; Maheshkumar H. Kolekar, “Enhanced Detection of Maize Leaf Blight in Dynamic Field Conditions Using Modified YOLOv9Kaustubh Gharat; Harshita Jogi; Kaustubh Gode; Kiran Talele; Sujata Kulkarni; Maheshkumar H. Kolekar,” in 2024 IEEE Space, Aerospace and Defence Conference (SPACE), 2024.

[14] Yuheng Li; Meng Wang; Chunhui Wang; Ming Zhong, “A method for maize pest detection based on improved YOLO-v9 model,” in 2024 7th International Conference on Computer Information Science and Application Technology (CISAT), 2024.

[15] M. Kumar, U. Pilania, S. Thakur, and T. Bhayana, “YOLOv5x-based Brain Tumor Detection for Healthcare Applications,” Procedia Comput Sci, vol. 233, pp. 950–959, 2024, doi: 10.1016/j.procs.2024.03.284.

[16] E. Iren, “Comparison of YOLOv5 and YOLOv6 Models for Plant Leaf Disease Detection,” Engineering, Technology and Applied Science Research, vol. 14, no. 2, pp. 13714–13719, Apr. 2024, doi: 10.48084/etasr.7033.

[17] S. Yang, J. Yao, and G. Teng, “Corn Leaf Spot Disease Recognition Based on Improved YOLOv8,” Agriculture (Switzerland), vol. 14, no. 5, May 2024, doi: 10.3390/agriculture14050666.

[18] R. N. Ariwa et al., “Comparative Performance Analysis of YOLOv8Model and Traditional Machine Learning Models for Maize Leaf Disease Detection,” Kasu Journal of Computer Science, vol. 1, no. 2, pp. 251–264, Jun. 2024, doi: 10.47514/kjcs/2024.1.2.008.

[19] M. A. R. Alif and M. Hussain, “YOLOv1 to YOLOv10: A comprehensive review of YOLO variants and their application in the agricultural domain,” Jun. 2024, [Online]. Available: http://arxiv.org/abs/2406.10139

[20] U. P. T. B. S. T. Manoj Kumar, “Utilizing YOLOv5x for the Detection and Classification of Brain Tumors,” in 2024 2nd International Conference on Disruptive Technologies (ICDT), 2024, pp. 1343–1348.

[21] M. Kumar and U. Pilania, “A framework for detection of drone using yolov5x for security surveillance system,” International Journal of Systematic Innovation, vol. 8, no. 3, pp. 99–114, 2024, doi: 10.6977/IJoSI.202409_8(3).0008.

[22] Hao, S., Gao, E., Ji, Z., & Ganchev, I. (2025). BCS_YOLO: Research on Corn Leaf Disease and Pest Detection Based on YOLOv11n. Applied Sciences, 15(15), 8231.

[23] Urmila Pilania; Manoj Kumar, “Automated Monitoring of Hydroponic System using IoT and Cloud based Technology for Sustainable Agriculture,” in 2024 1st International Conference on Advanced Computing and Emerging Technologies (ACET), 2024.

[24] Guo, D., Yuan, G., Liu, B., Liu, Z., Fen, L., Zhang, D., ... & Guo, J. (2025). SMA-YOLO: An enhanced architecture for the detection of corn diseases based on YOLOv12. Smart Agricultural Technology, 101502.

[25] Peng, G., Wang, K., Ma, J., Cui, B., & Wang, D. (2025). AGRI-YOLO: a lightweight model for corn weed detection with enhanced YOLO v11n. Agriculture, 15(18), 1971.

[26] Song, D., Peng, Y., Gu, X., & U, K. (2025). A Lightweight YOLOv8-Based Network for Efficient Corn Disease Detection. Mathematics, 13(24), 4002.

[27] Duan, Z., Jin, X., Yu, X., Ruan, X., Zhu, Y., Jiang, T., ... & Li, X. ABE-YOLO: A Lightweight and Efficient Model for Maize Ear Detection. Available at SSRN 6148968.

[28] Ali, B., Sarhan, A. M., & Hassanien, A. E. (2026). Leveraging Deep Learning for Apple and Maize Plant Disease Detection for Sustainable Smart Agriculture Solution. In Innovative AI Technologies Driving Sustainable Farming: Strategies for Improving Food Security (pp. 189-213). Cham: Springer Nature Switzerland.

[29] Ali, B., Sarhan, A. M., & Hassanien, A. E. (2026). Leveraging Deep Learning for Apple and Maize Plant Disease Detection for Sustainable Smart Agriculture Solution. In Innovative AI Technologies Driving Sustainable Farming: Strategies for Improving Food Security (pp. 189-213). Cham: Springer Nature Switzerland.

[30] Liu, M., & Gao, J. (2026). SLD-YOLO11: A Topology-Reconstructed Lightweight Detector for Fine-Grained Maize–Weed Discrimination in Complex Field Environments. Agronomy, 16(3), 328.