Enhanced YOLO-X Model for Tomato Disease Severity Detection using Field Dataset

Rajasree R [1] , C. Beulah Christalin Latha[2], Appu M [3]*

Abstract

In the past decade, the field of automatic plant disease identification has undergone significant complexity. Advancements in computer vision have enabled the rapid and precise detection of ailments, facilitated the development of effective treatments and ultimately led to higher crop yields. One of the most challenging scenarios in plant disease occurs when multiple diseases manifest on a single leaf, exacerbating the difficulty of diagnosis due to overlapping symptoms. This study addresses these challenges by employing an enhanced YOLO-X model for detection tomato leaf diseases. The technique presented here enhances the Spatial Pyramid Pooling layer in order to extract valuable features from training data of various sizes more efficiently. We were able to increase the model's ability to identify a broader spectrum of illness symptoms by concatenating variables from multiple layers and varying sizes. In addition, we incorporate a large number of connections to increase the generalizability of the design. The application of an IoU-based regression loss function increases the convergence of the network and the precision of the detection. For experimentation, we created a customized dataset consisting of 1220 tomato plant leaf images from various farms in Southern part of India, encompassing overlapping diseases and varying degrees of severity. The dataset includes images of healthy leaves as well as different severity levels of tomato leaf curl and tomato leaf mold stress on a single leaf. Our suggested improved SPP-based YOLO-X model beats the original YOLO-X model, according to experimental findings, which show an improvement in test dataset accuracy and a 73.42% mean Average Precision on field-collected dataset.

Keywords— Deep learning, convolutional neural network, one stage detection, plant disease, YOLO-X

I. INTRODUCTION

In India, tomatoes are a significant cash crop that is produced on 15% of the nation's total cultivated land. A significant portion of the global textile economy is contributed by the nation's tomato production and export, in addition to its local consumption. The crop is afflicted with several diseases during the course of its existence. A leaf might sometimes have many diseases, some of which have similar symptoms. Even an experienced pathologist may make mistakes when evaluating disease severity signs and the presence of numerous stressors. Precision farming practices have undergone a revolution with the development of artificial intelligence and computer vision technology. In plant disease detection systems, a number of machine learning and deep learning models have shown outstanding performance[1]. On field-collected or publicly accessible plant disease datasets, some researchers have combined deep learning-based feature extraction and classification tasks with transfer learning. In order to propose the use of pesticides or other preventative measures and achieve near-ideal performance in recognizing diseases signs automatically, several research have been conducted. Well-known deep learning architectures such as region-based convolutional networks [2], single shot detectors [3], and region proposal networks [4] have been employed in the area of plant leaf disease

[1]Department of Computer Science
Karpagam Academy of Higher Education, Coimbatore, Tamil Nadu, India

[2]Department of Digital Sciences
Karunya Institute of Technology and Sciences, Coimbatore, India

[3]Department of Computer Science
Sree Narayana Guru College, Coimbatore, India

*Corresponding Author

detection, with major alterations happening during the preceding few years [3]. Almost all previous studies either used the well-known Plant Village public dataset [3] or their own datasets collected in the field [5], [6].

But only a small number of studies have looked at 1) the stages of disease growth and 2) the chance that many living and nonliving things can attack a plant leaf at the same time. In these situations, it is hard for both human and automatic monitoring systems to figure out the type of infection and the exact area of sickness signs. Figure 1 shows, as an example, the different stages that tomato leaf curl disease goes through.

In this study, we describe a YOLO-X-s based detecting system that uses a modified Spatial Pyramid Block to combine fine spatial data with local features to find sickness phases and split diseases with symptoms that overlap. We made the spatial pyramid pooling block better by putting together feature mAPs at low-level scales. This helped us solve the problem more accurately. The original size feature vector was added to improve the quality of the features. The recognition performance got even better when the Alpha IoU regression loss function was used.

Here are the main things that this suggested plan brings to the table:

- A better YOLO-Xs model with a modified Pyramid pooling module (SPP) layer is given so that many diseases on a single leaf plant can be found. It collects location information at local, multi-scale levels to get the information it needs more quickly.
- To improve generalization and convergence, we used Alpha IoU (Intersection Over Union) loss as the bounding box regression for multiple disease localization when multiple diseases showed up on the same plant leaf.

With the help of enhancement, a group of unique shots from a tomato field are shown. The photos show how diseases spread and how many different diseases can be found on a single leaf.

II. REVIEW OF LITERATURE

Plant leaf diseases may be identified using computer vision-based methods for (1) detection and (2) identification. Both of the methodologies used in the area are prevalent in research literature released in the last ten years.

Highlights of various modern and cutting-edge techniques for identifying and detecting plant leaf diseases literatures are included in Table 1. In this part, these methods will be thoroughly reviewed in relation to: (1) the kind of application targeted, (2) the methodology employed, (3) the contribution. The usage of deep learning in this field of study has increased during the previous several years. Transfer learning and data enrichment have made it easier to use deep learning models on a variety of devices, such as central processing units (CPUs) and graphics processing units (GPUs). In the study done by Abayomi et al. [10], the MobileNet v2 model was trained by adding more color space data in a few different ways. Transfer learning and data enrichment have made it easier to run deep learning models on CPUs, GPUs, and other types of computer systems. This is because these two methods have been put together. In the study done by Abayomi et al. [10], color space data addition methods were used to train the MobileNet v2 model. To compare the effectiveness of the classification, the scientists trained the model using images of cassava leaf disease of various quality levels. According to a study, low-quality images cause the classification accuracy to decrease. The sick region has also been identified using high-quality images and a color difference, according to authors [11]. Using advanced machine learning classifiers, such as the bagging tree ensemble, it is now possible to identify sick regions based on color and textual information with an overall accuracy of 99%. Plant disease monitoring systems that use computer vision are meant to automatically find and identify the part of a plant that is sick. Because of this, these systems use customizable deep-learning meta-architectures that have been used in the related study.

The earliest deployed deep learning algorithms were region-based convolutional neural networks (RCNNs), and their purpose was to recognize objects in general [3]. In line with Fast Convolutional Neural Network [12], Faster Region based CNNs, and R-FCNNs [3]. Segmentation and noise reduction operations were carried out using the OTSU algorithm and multilayer median filters, respectively. By using a two-stage detector, the technique achieves an inference time of 0.52 s. Rehman et al. [14] employed a comparable two-stage detector, Mask RCNN, to identify sick regions after contrast stretching. CNN performs feature extraction of improved areas, which was afterwards categorized. After applying Kapur's entropy to choose the best features, accuracy was improved. Another important work [15] identified various rice diseases by using a quicker R-CNN with a reinforced backbone to analyze the still pictures of the rice. In terms of recall and accuracy, the upgraded two-stage model performs better than earlier models such as YOLO-V3, while having longer detection times. The model presented in [16] significantly improved map by using a superior anchor box method that was based on a more efficient RCNN model for weed detection.

With single shot detection (SSD), you need to make a region proposal network in order to get the best total speed and a faster inference time. But you can get both without making a region proposal network. They use certain boxes to figure out how likely it is that a certain item will be in a picture. This improved model, which used the Inception module and Rainbow union, was used to get information about features and make it easier to find ill spots on apples [17]. Both the VGG and the origin module were added to the model so that it could diagnose diseases with a map of 78.8%. YOLO (You Only Look Once) [18] models, on the other hand, are single-stage detectors that can find and label objects in a picture with just one forward spread. The name of these models comes from the saying "you only look once." It initially divides a picture into a grid to begin the detection process, and for each bounding box, it then forecasts the likelihood of an item and its class. In order to identify tomato diseases in difficult background settings, Wang et al. [19] utilized a similar YOLO architecture and added the DenseNet block for feature extraction. Additional information is provided in Table 1 of the relevant work comparison. The YOLO-V5 model was effective in identifying bacterial leaf spots. Additionally, the outcomes were contrasted with those of YOLO-V3, YOLO-V4, Single Shot Detector algorithm, and other two-stage detectors techniques [20].

A new upgrade to the YOLO series, called YOLO-X [26], has made a considerable improvement in the field of object identification. To improve feature extraction, YOLO-X makes use of the YOLO-V3 with a Darknet architecture added with SPP Layer baseline structure. To diagnose kiwi leaf diseases, DeepLabV3 and UNet are used in conjunction with YOLO-X to separate the leaves from the complicated backdrop. The mix of Cross Stage Partial Network and sigmoid activation function in the approach of finding colon cancer has led to a better version of the YOLO v3 algorithm.

The model is strengthened for real-time polyp identification by using the CIOU loss function [28]. The YOLO-V5 model was used by Chen et al. [22] to accurately detect plant disease. Using the SE module and Involution Bottleneck, the accuracy and number of parameters were improved. The researchers must overcome difficulties with disease development and detecting many lesion areas in a single frame. Multiple diseases may be difficult to identify, both manually and when using artificial image processing algorithms [29], since their locations and symptoms tend to overlap. The Efficient Net model, which has a classification accuracy of 96%, was successfully used to address the plant disease detection problem on a single cucumber leaf [25]. The authors utilized a Ranger optimizer to find symptoms that seemed to be related.

Although SSD algorithm with respectable object identification and detection performance have been employed in the methodologies from the literature (included in Table 1).

TABLE I. SELECTED STUDIES FROM THE LITERATURE ON PLANT DISEASE DETECTION

Reference	Application Targeted	Methodology Employed	Contribution
[7]	Automatic detection of plant diseases	Classic machine learning	Utilization of various classic machine learning approaches
[13]	Rice disease identification	Faster region based model + K-means clustering	Identification of rice diseases using a two-stage detector with clustering
[17]	Identification of apple sick patches	YOLO-V5 model with Inception module and Rainbow concatenation	Enhanced feature extraction and improved identification of apple sick patches
[19]	Tomato disease identification	YOLO architecture with DenseNet block	Utilization of YOLO architecture for identifying tomato diseases
[22]	Plant disease detection	YOLO-V5 model with SE module and Involution Bottleneck	Accurate detection of plant diseases with improved accuracy and parameters
[23]	Segmentation of sick lesions on paddy leaves	CNN and YOLO	Segmentation of sick lesions with refined model parameters
[24]	Classification of severity of strawberry leaf disease	Faster RCNN and Siamese network	Identification of leaf position and estimation of severity

The majority of these investigations have focused on classifying lesions or locating locations. The difficulties associated with accurately distinguishing distinct phases of disease severity and a multitude of stresses on a single leaf are presently the ones that have received the least amount of attention. In order to identify overlapping and overlapping plant leaf diseases, we suggest in this study an updated YOLO-X model tailored specifically for the application.

III. MATERIALS AND METHODS

The suggested technique is fully explained in this part and will be utilized to handle the issues of (1) leaf disease progression symptoms and (2) the presence of several diseases on a single tomato leaf. The supplied data is initially pre-processed to remove extraneous background information and guarantee class balance. The suggested deep learning model's specifics are then discussed, with a focus on the two crucial instances indicated above.

A. Customized Dataset Creation

Tomatoes are grown in different parts of Southern India from March to May. For the purpose of this study, images were taken from various tomato fields in Tamilnadu and Kerala. No extra pesticides or fertilizers were used to maintain the conditions that allowed the infections to flourish. The pictures were taken at various angles using a smartphone (Apple IPhone X) that was held 15 to 25 cm away from the tomato leaf. Original image size was 1125 x 2436 pixels with 19.5:9 ratio. Images were taken in the morning, with varied lighting conditions. The smartphone's focus was changed to portrait mode and zoom mode to capture one leaf. Seven different categories of tomato curl severity and disease coexistence images were created from the data.

Figure 1 displays a few examples of images extracted from the dataset. In India, the ailment known as "tomato curl virus" is widespread. Stage 1 of tomato leaf curl's early signs is described as leaf chlorosis. Within two to three weeks, the symptoms become worse as the leaf's veins start to thicken and darken. From the underside of the leaf, the thickening of the veins is plainly visible. The second stage of the tomato leaf disease is distinguished by the subsequent curling of the leaf margins. Three weeks later, the leaves begin to develop dark black specks that prevent the virus from spreading.

The pathologists determine that it is tomato leaf mold brought on by the aphid infestation. The curl virus and leaf mold infect almost all of the nearby plants.

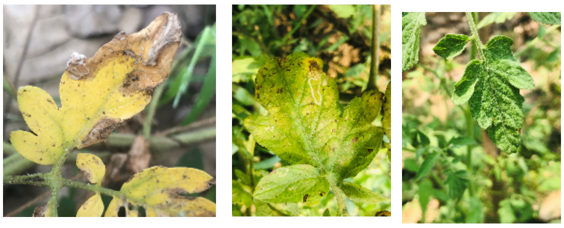


Fig. 1. Sample dataset images collected from the tomato field.

B. Pre-processing the Dataset Images

Since leaf photographs were taken in actual field settings, they included background details like dirt and tree branches, among other things. The work of the suggested detection system is anticipated to be made simpler by extracting just the leaf region containing one or more diseases and deleting the background information [30]. There are many methods in place to eliminate extraneous background details from real-world situations [9]. Grab Cut technique [31] is a quick and effective machine learning-based approach that can eliminate unwanted background data with little human adjustments. On the basis of graph cuts, the backdrop is removed. According to a user-provided window, anything beyond the window is taken into account to be the clear backdrop, however within the window, both the foreground and the background may exist. The procedure is repeated until convergence in order to fine-tune the background removal job. On the hardware utilized for testing, background removal was reported to take an average of 4.12 seconds per picture. The algorithm's ability to eliminate background data outside of the infected leaf was determined to be effective. After the backgrounds have been removed, the images are annotated before being subjected to image augmentation methods including flipping, rotating, and brightness boosting. As thoroughly covered in Section IV-A, this aids in enhancing the model's training performance and preventing overfitting.

C. Annotation and Labelling

The Roboflow annotation tool was used to manually design the ground truth boxes and labeling the healthy and diseased tomato leaf images. According to Figure 2 shows the experiments done for the annotation and labeling, sample taken from the roboflow tool. Each box's enclosing box coordinates, height, breadth, and class name are listed in Figure 3. When an image is included into the model for learning, testing, or assessment, an accompanying XML file is included [17].

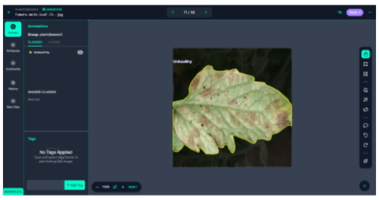
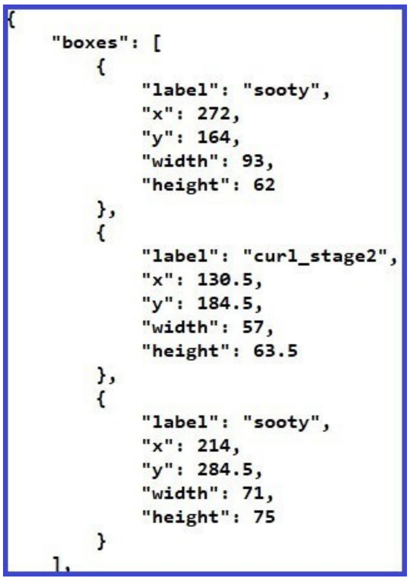


Fig. 2. Tomato leaf image annotation with bounding box created using roboflow tool. Images captured at different angles and sizes are included in the data set used for the tests. In order to maintain the bounding boxes on the item of interest regardless of any augmentation step applied to it subsequently, we executed an auto-orient procedure. As often requested by YOLO, the photographs were also scaled using a normalization set at a resolution of 416X416. When working with an unbalanced data set, we prefer to execute the augmentation techniques to minimize over fitting. Proposed study used augmentation techniques includes flip horizontal, rotation, and brightness (25%). The addition of these augmentations will increase the data set's size as well as the variety of the photographs taken in various lighting situations. The data set is augmented at random, increasing the size to 1, 112. Roboflow data services were used for each of these pre-processing phases.



```
{
    "boxes": [
        {
            "label": "sooty",
            "x": 272,
            "y": 164,
            "width": 93,
            "height": 62
        },
        {
            "label": "curl_stage2",
            "x": 130.5,
            "y": 184.5,
            "width": 57,
            "height": 63.5
        },
        {
            "label": "sooty",
            "x": 214,
            "y": 284.5,
            "width": 71,
            "height": 75
        }
    ],
```

Fig. 3. Annotation details of the xml file taken from the roboflow tool.

D. Hardware and software configuration

Every experiment was run on Windows 7 system using an Intel i3 processor. The necessary repositories and libraries were installed before configuring the experimental environment. Pre-trained weights for the YOLO-X model were downloaded when the dataset was uploaded to the drive.

Experiment was performed using google collab notebook environment. With the help of the hyper-parameters using 100 epochs, our suggested model was trained. The best weights produced after training were kept and then utilized to assess the effectiveness of our suggested model on test images with a batch size of 32. In order to conduct the experiment, 640 × 640 resolution images from the dataset were used. The dataset has been split automatically with the code on the collab on the basis of 80% of the images were utilized for training and testing, while the other 20% were used for validation process.

E. Enhanced YOLO-X proposed model for Tomato Leaf Disease Detection

YOLO-X deep learning model is is a single stage object trackertechnique that differs from YOLO-V3 in multiple respects.It derives from DarkNet53.YOLOs are frequently substituted by a severed skull.Beginning with a 1X1 convolutional layer, the feature channel for each level of FPN features is reduced to 256.Then, for the classification and regression tasks, respectively, we add two parallel branches with two 3X3 conv layers. The anchor-free detector YOLO-X [26] has shown exceptional speed and accuracy performance. To improve convergence while the system was being trained, the head of YOLO-X was cut off from the original detector [32]. Because to the implementation's anchor-free design, the overall number of trainable parameters has been significantly decreased. SimOTA [26], which reduces the amount of time required for training and aids in the solution of the Optimal Transport (OT) issue [21], is also used to enhance the label assignment approach.

For the application that is covered in this article, we build a more sophisticated YOLO-X model. The CSP darknet serves as the foundation for feature extraction in this model. In order to accomplish better feature fusion throughout the classification and regression tasks, the feature layers are first up-sampled for feature fusion and then down-sampled for classification [27]. One of the crucial elements that must exist for good feature extraction is the focus module. The input photographs are split into four pieces, and then concatenated in order to preserve information about the features of the objects. This makes it possible to observe the characteristics more clearly. The Bottleneck CSP layer, which comes after the top layer, is where the deep features are recovered with more accuracy. Convolution, related batch normalization, and activation processes are performed on the feature maps. To learn about overlapping and mild symptoms, advanced data augmentation methods like mosaic and mix-up are utilized during training [33]. The non-maximal suppression (NMS) strategy prevents the chance of multiple detections happening at the same time.

A definition for the modeling error that arises between the anticipated class and the ground truth is Binary Cross Entropy (BCE) loss [34] with logics. A sigmoid activation function is utilized to eliminate all accurate predictions [26]. The bounding box's coordinates (x, y, w, and h) are predicted in the regression branch's output. In order to forecast bounding box outputs, YOLO-X employs the IoU metric and compares its predictions to the actual data.

F. YOLO X Model Working environment with Bounding Box

The localization of objects and their categorization are two processes that are critically important for applications based on computer vision. How precisely a machine learning model can pinpoint an object's placement inside a scene or image is determined by the loss function [35]. This is why conventional single-stage and two-stage detectors are developed using the bounding box regression approach.

Vanishing gradients provide a challenge because they cause IoU losses, which prohibit the model becoming convergent. The basis of the problem is that the predicted boxes do not precisely overlap the ground truth boxes, leading to inaccurate findings. To increase the precision with which the objects were tracked, it was chosen to give a number of enhanced bounding box losses depending on a number of parameters. The Generalized Intersection over Union, the Distance Intersection over Union the Complete Intersection over Union, and the Efficient Intersection over Union (GIoU, DIoU, CIoU, EIoU) are acronyms for these intersections. The amount of overlap between the target and the anchor boxes is represented by these loss functions via the use of several metrics.

IV. RESULTS AND DISCUSSIONS

Training effectiveness was assessed based on increased convergence speed and detection performance of overlapping illness symptoms and severity classes using datasets collected from the field. For detection models, the Mean Average Accuracy (mAP) statistic is often used. This statistic indicates the accuracy for all classes when the Intersection over Union (IoU) criteria is set to 50%. The IoU threshold was held constant at 0.5 while calculating the mAP score. Similar levels of training accuracy were originally shown by the default YOLO-X model, but it was unable to converge in the last 30 epochs of training. As illustrated in Figure 4, this resulted in the default YOLO-X model's accuracy being lower (69.90%) than the suggested enhanced YOLO-X model's accuracy (73.42%).

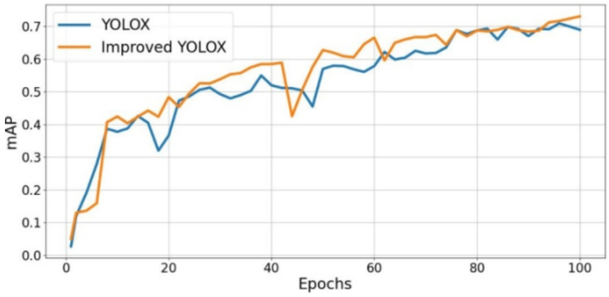


Fig 4. Graph depicts the YOLOX vs Improved YOLOX mAP vs epochs.

A SE block was added by the authors to the SPP module of the standard YOLO-X in an effort to boost detection performance. [39] To achieve this, many tests were conducted. The YOLO-X-SE model's training performance was plainly overfit in the last 20 training epochs, which decreased inference speed. Several unique classes were misclassified, despite the fact that our model's inference time is a little faster than the default YOLO-X. The model's validation and test results were assessed over a range of IOU losses in order to attain the best degree of convergence and mAP performance. The model performed noticeably better in terms of the results of the enhanced SPP block compared to vanilla-IoU and other regression techniques. For the localisation of overlapping medical symptoms, this was crucial.

The best weights are chosen when the training process is finished, and the model's performance on the test dataset is assessed using these weights. The regression will come to an end once the item included by the bounding box has been located. The BCE loss may be used to assess a bounding box's capacity to hold an item. The disease's ability to exist inside the anticipated bounding box is assessed using the confidence score. We conducted a number of tests to determine the ideal threshold value in order to increase the detection mAP score. This figure shouldn't be too high or low in order to avoid false positives and genuine predictions, respectively. The degree of confidence will stay at 0.25 after an analysis of the test data. Each projected bounding box in the test photos has a corresponding confidence level, which may be used to represent the detection performance of the test dataset as determined by Improved YOLO-X in Figure 8. The result offers proof that strengthens confidence in the localization and classification procedures.

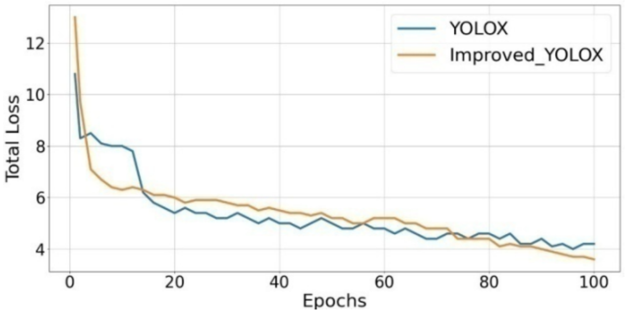


Fig. 5. Graph depicts the YOLOX vs Improved YOLOX loss vs epochs.

he YOLO-X model uses the same basic idea as its predecessor but with an SPP anchor-free system.

The decoupled head lowers convergence, as shown by comparing its performance to that of the YOLO-V4 [40] and YOLO-V5 [20] models, both of which were trained and validated using our Tomato severity dataset. The findings are shown in Table 2 depending on the accuracy and processing time needed for inference. The accuracy of detection has reduced even though YOLO-V4 and YOLO-V5 have inference rates that are noticeably quicker than our model. Both the default YOLO-X model and our Improved YOLO-X model demonstrate higher convergence performance compared to other models trained on our dataset because the anchor-free method decreases the possibility of complexity and obstacles emerging during training. The mAP values acquired during training using both the standard YOLO-X model and the suggested enhanced YOLO-X model are shown in Figure 4.

TABLE II. RESULT OF MODEL COMPARISON BASED ON TIME ATTRIBUTE

Experime-nted model	mAP in Healthy	mAP in Leaf Curl Disease - Stage 1	mAP Leaf Curl Disease - Stage 2	mAP in leaf mold
YOLO-v4	27.87%	56.61%	40.12%	54.01%
YOLO-v5	65.91%	29.21%	47.72%	53%
YOLO-X	64.21%	49.58%	61.23%	62.31%
YOLO-X-SE	56.01%	63%	64.02%	59.82%
Enhanced YOLO-X	62.37%	62.32%	65.72%	75.02%

evaluate how well our suggested YOLO-X model performs in classifying overlapping symptoms and severity phases. Because it was incorrectly categorized with the healthy and curl stage-2 class, curl stage-1 was recognized with substantially lower average precision. The overlapping signs of leaf mold and curl stage-2 were also more accurately picked up by our upgraded YOLO-X model. As the class does not visually resemble any other classes and there are not many problematic images in our test dataset, the models successfully identify leaf stress.

A. Comparison with Other Existing Models

Here, we contrast the effectiveness of our upgraded YOLO-X model with that of current state-of-the-art models. The performance of our proposed model is contrasted with that of cutting-edge models in TABLEAU 3. The best accurate models for our dataset are training. We trained the YOLO-V4, YOLO-V5, YOLO-V7, Efficientdet, and YOLO-X models on our tomato severity dataset for 100 iterations using the default code settings. The mAP scores are noticeably lower than those anticipated by the YOLO-X model, as seen in Table 3. We created the YOLO-X-ti-lite model, a YOLO-X version that works well for edge computing. An SPP block that has been tailored for embedded device operation is included in the model. However, it did not provide the expected results for a specific dataset. The advantages of applying Spatial Pyramid Pooling findings into mAP analysis are shown in Table 4. It indicates that when SPP with 3,5,7,9 connections is taken into account, our Enhanced YOLO-X model performs better.

TABLE III. RESULT ANALYSIS OF THE PROPOSED MODEL IN COMPARISON WITH OTHER STATE OF HEART ALGORITHMS

Experime-nted model	**mAP in Healthy**	**mAP in Leaf Curl Disease - Stage 1**	**mAP Leaf Curl Disease - Stage 2**	**mAP in leaf mold**
YOLO-v4	27.87%	56.61%	40.12%	54.01%
YOLO-v5	65.91%	29.21%	47.72%	53%
YOLO-X	64.21%	49.58%	61.23%	62.31%
YOLO-X-SE	56.01%	63%	64.02%	59.82%
Enhanced YOLO-X	62.37%	62.32%	65.72%	75.02%

TABLE IV. AN ANALYSIS OF EVALUATION METRICS USING MEAN AVERAGE PRECISION WITH SPATIAL PYRAMID POOLING

Spatial Pyramid Pooling (5,9,14)	**Spatial Pyramid Pooling (3,5,7,9)**	**Skip Connecti ons**	**mAP**
Yes	No	No	69.90 %
No	Yes	No	67.42%
No	Yes	Yes	71.31%
No	Yes	Yes	73.42%

B. Future Work and research opportunities

Multiple stresses on the host and multiple disease stages on the leaf are typical outdoor conditions. For these two scenarios, this research suggests a deep learning-based solution that has been put to the test on a large dataset. However, the following are some possible improvements:

In order for our model to be applicable in real-world scenarios, its reliability is of utmost importance. To achieve this, the dataset will be continuously enhanced by incorporating new photographs depicting various ailments and harvests.

- It is strongly advised to subject the recommended model to additional training and testing, specifically using individual leaves placed against a clean background.
- To enhance its practicality and efficiency, we intend to allocate a greater financial investment towards obtaining more field samples.
- Furthermore, to expedite the training and testing processes on state-of-the-art technology, we will prioritize designing the model to be as lightweight as possible.

V. CONCLUSIONS

The suggested study offers a framework for classifying symptoms of a particular disease on tomato plants according to increasing severity. The detection of many illnesses that are present on a single leaf may also be done using this approach. We have suggested a YOLO-X-based model with an enhanced Spatial Pyramid Pooling block to achieve this. Various pooling rates were used to aggregate multi-scale characteristics. Remaining links were added to better preserve spatial information. With the use of this model, we were able to identify diseases symptoms that were similar and overlapped more accurately. The suggested model outperformed the default YOLO-X by 3.27 percent according to experimental findings, achieving mAP scores of 73.42% and 72.31% using training dataset and testing dataset respectively. Additionally, Curl stage-2 and Leaf mold yielded the greatest results, with average precisions of 65.76% and 74.02%, respectively, for overlapping and co-existing classes.

REFERENCES

[1] J. G. A. Barbedo, “Factors influencing the use of deep learning for plant disease recognition,” Biosyst. Eng., vol. 172, pp. 84–91, Aug. 2018.

[2] Z. Lin, S. Mu, F. Huang, K. A. Mateen, M. Wang, W. Gao, and J. Jia, “A unified matrix-based convolutional neural network for fine- grained image classification of wheat leaf diseases,” IEEE Access, vol. 7, pp. 11570–11590, 2019.

[3] M. H. Saleem, S. Khanchi, J. Potgieter, and K. M. Arif, “Image-based plant disease identification by deep learning meta-architectures,” Plants, vol. 9, no. 11, p. 1451, Oct. 2020.

[4] R. Wang, L. Jiao, C. Xie, P. Chen, J. Du, and R. Li, “S-

RPN: Sampling- balanced region proposal network for small crop pest detection," Comput. Electron. Agricult., vol. 187, Aug. 2021, Art. no. 106290.

[5] D. Jiang, G. Li, C. Tan, L. Huang, Y. Sun, and J. Kong, "Semantic segmentation for multiscale target based on object recognition using the improved Faster-RCNN model," Future Gener. Comput. Syst., vol. 123, pp. 94–104, Oct. 2021.

[6] A. Fuentes, S. Yoon, S. C. Kim, and D. S. Park, "A robust deep-learning- based detector for real-time tomato plant diseases and pests recognition," Sensors, vol. 17, no. 9, p. 2022, 2017.

[7] N. Kundu, G. Rani, V. S. Dhaka, K. Gupta, S. C. Nayak, S. Verma, M. F. Ijaz, and M. Woźniak, "IoT and interpretable machine learning based framework for disease prediction in pearl millet," Sensors, vol. 21, no. 16, p. 5386, Aug. 2021.

[8] S. K. Upadhyay and A. Kumar, "A novel approach for Rice plant diseases classification with deep convolutional neural network," Int. J. Inf. Technol., vol. 14, no. 1, pp. 185–199, Feb. 2022.

[9] B. M. Patil and V. Burkpalli, "Segmentation of tomato leaf images using a modified chan vese method," Multimedia Tools Appl., vol. 81, no. 11, pp. 15419–15437, May 2022.

[10] O. O. Abayomi-Alli, R. Damaševičius, S. Misra, and R. Maskeliu¯nas, "Cassava disease recognition from low-quality images using enhanced data augmentation model and deep learning," Exp. Syst., vol. 38, no. 7, 2021, Art. no. e12746.

[11] A. Almadhor, H. T. Rauf, M. I. U. Lali, R. Damaševičius, B. Alouffi, and A. Alharbi, "AI-driven framework for recognition of guava plant diseases through machine learning from DSLR camera sensor based high resolution imagery," Sensors, vol. 21, no. 11, p. 3830, 2021.

[12] A. Pramanik, S. K. Pal, J. Maiti, and P. Mitra, "Granulated RCNN and multi-class deep SORT for multi-object detection and tracking," IEEE Trans. Emerg. Topics Comput. Intell., vol. 6, no. 1, pp. 171–181, Feb. 2022.

[13] G. Zhou, W. Zhang, A. Chen, M. He, and X. Ma, "Rapid detection of Rice disease based on FCM-KM and faster R-CNN fusion," IEEE Access, vol. 7, pp. 143190–143206, 2019.

[14] Z. U. Rehman, M. A. Khan, F. Ahmed, R. Damaševičius, S. R. Naqvi, W. Nisar, and K. Javed, "Recognizing apple leaf diseases using a novel parallel real-time processing framework based on mask RCNN and transfer learning: An application for smart agriculture," IET Image Processing, vol. 15, no. 10, pp. 2157–2168, 2021.

[15] D. Li, R. Wang, C. Xie, L. Liu, J. Zhang, R. Li, F. Wang, M. Zhou, and W. Liu, "A recognition method for Rice plant diseases and pests video detection based on deep convolutional neural network," Sensors, vol. 20, no. 3, p. 578, Jan. 2020.

[16] M. H. Saleem, J. Potgieter, and K. M. Arif, "Weed detection by faster RCNN model: An enhanced anchor box approach," Agronomy, vol. 12, no. 7, p. 1580, Jun. 2022.

[17] P. Jiang, Y. Chen, B. Liu, D. He, and C. Liang, "Real-time detection of apple leaf diseases using deep learning approach based on improved convolutional neural networks," IEEE Access, vol. 7, pp. 59069–59080, 2019.

[18] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR),

Jun. 2016, pp. 779–788.

[19] X. Wang and J. Liu, “Tomato anomalies detection in greenhouse scenarios based on YOLO-dense,” Frontiers Plant Sci., vol. 12, p. 533, Apr. 2021.

[20] M. P. Mathew and T. Y. Mahesh, “Leaf-based disease detection in bell pepper plant using YOLO v5,” Signal, Image Video Process., vol. 16, no. 3, pp. 841–847, Apr. 2022.

[21] J. Yao, Y. Wang, Y. Xiang, J. Yang, Y. Zhu, X. Li, S. Li, J. Zhang, andG. Gong, “Two-stage detection algorithm for kiwifruit leaf diseases based on deep learning,” Plants, vol. 11, no. 6, p. 768, Mar. 2022.

[22] Z. Chen, R. Wu, Y. Lin, C. Li, S. Chen, Z. Yuan, S. Chen, and X. Zou, “Plant disease recognition model based on improved YOLO-V5,” Agron- omy, vol. 12, no. 2, p. 365, Jan. 2022.

[23] G. Ganesan and J. Chinnappan, “Hybridization of ResNet with YOLO classifier for automated paddy leaf disease recognition: An optimized model,” J. Field Robot., vol. 39, no. 7, pp. 1085–1109, Oct. 2022.

[24] J. Pan, L. Xia, Q. Wu, Y. Guo, Y. Chen, and X. Tian, “Automatic strawberry leaf scorch severity estimation via Faster R-CNN and few-shot learning,” Ecol. Informat., vol. 70, Sep. 2022, Art. no. 101706.

[25] P. Zhang, L. Yang, and D. Li, “EfficientNet-B4-Ranger: A novel method for greenhouse cucumber disease recognition under natural com- plex environment,” Comput. Electron. Agricult., vol. 176, Sep. 2020, Art. no. 105652.

[26] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, “YOLO-X: Exceeding YOLO series in 2021,” 2021, arXiv:2107.08430.

[27] I. Pacal, A. Karaman, D. Karaboga, B. Akay, A. Basturk, U. Nalbantoglu, and S. Coskun, “An efficient real-time colonic polyp detection with YOLO algorithms trained by using negative samples and large datasets,” Comput. Biol. Med., vol. 141, Feb. 2022, Art. no. 105031.

[28] H. Zhai, J. Cheng, and M. Wang, “Rethink the IoU-based loss functions for bounding box regression,” in Proc. IEEE 9th Joint Int. Inf. Technol. Artif. Intell. Conf. (ITAIC), Dec. 2020, pp. 1522–1528.

[29] S. K. Noon, M. Amjad, M. A. Qureshi, and A. Mannan, “Use of deep learning techniques for identification of plant leaf stresses: A review,” Sustain. Comput., Informat. Syst., vol. 28, Dec. 2020, Art. no. 100443.

[30] S. M. Jaisakthi, P. Mirunalini, D. Thenmozhi, and Vatsala, “Grape leaf disease identification using machine learning techniques,” in Proc. Int. Conf. Comput. Intell. Data Sci. (ICCIDS), Feb. 2019, pp. 1–6.

[31] S. Sun, M. Jiang, D. He, Y. Long, and H. Song, “Recognition of green apples in an orchard environment by combining the GrabCut model and Ncut algorithm,” Biosyst. Eng., vol. 187, pp. 201–213, Nov. 2019.

[32] Y. Li, Z. Guo, F. Shuang, M. Zhang, and X. Li, “Key technologies of machine vision for weeding robots: A review and benchmark,” Comput. Electron. Agricult., vol. 196, May 2022, Art. no. 106880.

[33] G. Wang, H. Zheng, and X. Zhang, “A robust checkerboard corner detec- tion method for camera calibration based on improved YOLO-X,” Frontiers Phys., vol. 9, p. 828, Feb. 2022.

[34] U. Ruby and V. Yendapalli, “Binary cross entropy with deep learning technique for image classification,” Int. J. Adv. Trends Comput. Sci. Eng., vol. 9, no. 4, pp. 5393–5397, Aug.

2020.

[35] Z. Gevorgyan, "SIoU loss: More powerful learning for bounding box regression," 2022, arXiv:2205.12740.

[36] J. He, S. Erfani, X. Ma, J. Bailey, Y. Chi, and X.-S. Hua, "α-IoU:A family of power intersection over union losses for bounding box regression," in Proc. Adv. Neural Inf. Process. Syst., vol. 34, 2021, pp. 20230–20242.

[37] C.-L. Wang, M.-W. Li, Y.-K. Chan, S.-S. Yu, J. H. Ou, C.-Y. Chen, M.-H. Lee, and C.-H. Lin, "Multi-scale features fusion convolutional neu- ral networks for Rice leaf disease identification," J. Imag. Sci. Technol., vol. 66, no. 5, pp. 1–12, 2022.

[38] T. Wu, Q. Huang, Z. Liu, Y. Wang, and D. Lin, "Distribution-balanced loss for multi-label classification in long-tailed datasets," in Proc. Eur. Conf. Comput. Vis. Cham, Switzerland: Springer, 2020, pp. 162–178.

[39] G. Li, X. Huang, J. Ai, Z. Yi, and W. Xie, "Lemon-YOLO: An efficient object detection method for lemons in the natural environment," IET Image Process., vol. 15, no. 9, pp. 1998–2009, Jul. 2021.

[40] A. Bochkovskiy, C.-Y. Wang, and H.-Y. Mark Liao, "YOLO-V4: Optimal speed and accuracy of object detection," 2020, arXiv:2004.10934.

[41] C.-Y. Wang, A. Bochkovskiy, and H.-Y. Mark Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," 2022, arXiv:2207.02696.

[42] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 10781–10790.