DOI: 10.6977/IJoSI.202309_7(7).0005
A. R. Kolachi, S. R. Soomro, etc./Int. J. Systematic Innovation, 7(7),80-88(2023)

Cotton leaf disease classification using YOLO deep learning framework and indigenous dataset

Abdul Rahim Kolachi[1,2], Shoaib R. Soomro[1]*, Shadi Khan Baloch[2], Aamir Ali Patoli1, Sohail Anwar[1]
[1]Electronic Engineering Department, Mehran University of Engineering and Technology Jamshoro, Pakistan
[2]Mechatronic Engineering Department, Mehran University of Engineering and Technology Jamshoro, Pakistan
* Corresponding author E-mail: shoaib.soomro@faculty.muet.edu.pk
(Received 21 July 2023; Final version received 4 September 2023; Accepted 14 September 2023)

Abstract

Cotton is one of the economically significant agricultural products in the world and is among the key export resources in Pakistan. Despite the significant pest control techniques and mechanisms, the cotton crop is highly prone to bacterial and viral plant diseases that significantly reduce its yield. Early detection can enable the identification of infected field patches and plays an important role in controlling the spread of the disease. This paper presents the automated classification for bacterial blight and curl virus in cotton plants through the customized implementation of a state-of-the-art YOLO deep learning framework. The disease classification is performed on YOLOv5, and its performance is compared against YOLOv6 and YOLOv7. The transfer learning of the pre-trained model is facilitated through an indigenous image dataset collected from local agricultural fields in Sindh, Pakistan. Different augmentation techniques are employed to increase the size and diversity of the dataset. The employed model is evaluated for various performance metrics, such as accuracy, mean average precision, and confusion matrix. The results indicate 92% accuracy in disease classifications. The confusion matrix analysis indicates up to 100% true positive rates for curl virus, and an 88% true positive rate for detecting bacterial blight and healthy leaves. An inference time of 25 milliseconds indicates fast prediction suitable for on-field real-time applications and potential incorporation of the model in the point of care testing (PoCT) devices.

Keywords: Cotton disease classification, deep learning, real-time detection, YOLO.

1.Introduction

Agriculture is a driving force behind the economic growth of a country and its success is highly dependent on the quality and quantity of its agricultural harvests. Cotton cultivation stands out as a major contributor to both the economy and industries of a nation. Not only does it provide essential fiber, but it also generates oil and protein, making it a valuable commodity on a global scale (Bodhe et al., 2018). However, the cotton crop is vulnerable to a variety of plant diseases that have caused a significant decline in its yield and productivity in recent years. These diseases often manifest as visible symptoms on the leaves of the plant, and while farmers have traditionally relied on their observations for diagnosis, this method is prone to inaccuracies and can result in the overuse of pesticides. This, in turn, can lead to further reductions in crop yield and potential harm to the environment. It is, therefore, of utmost importance to find an accurate and efficient method for diagnosing cotton crop diseases. In recent years, various machine-learning techniques and tools have been widely used to perform the automated detection of plant diseases in cotton crops (Ahmed, 2021; Kumar et al., 2021; Prashar et al., 2017; Zekiwos & Bruck, 2021). The development of a reliable model that can diagnose cotton crop diseases with precision and speed is crucial to prevent the spread of diseases in the early stages and to ensure the right amount of pesticide is applied to affected plants.

This work presents the automated cotton plant disease classification method based on you only look once (YOLO) deep learning model. Compared to the existing works, the presented work uses the locally collected dataset combined with the dataset available from online resources and performs the detailed implementation of YOLOv5, and compares it with YOLOv6 and YOLOv7. The subsequent sections are distributed as follows: Section II presents the review of existing literature on plant leaf classification. Section III discusses the methodology of the presented implementation, and Section IV presents and discusses the results.

2.Literature Review

The detection of crop plant diseases has been explored using various techniques and methods. Following is a brief review of existing relevant literature work.

Ashourloo et al., 2016 used machine learning algorithms such as partial least square regression (PLSR), ν support vector regression (ν-SVR), and Gaussian process regression (GPR) methods to detect wheat leaf rust disease (Ashourloo et al., 2016). They evaluated the impact of the training data on the results and explored the influence of disease symptoms on the prediction performance of the ML algorithms. The performance of the machine learning methods is compared with the spectral vegetation indices (SVIs). Rothe et al.,2012 introduced a pattern recognition-based method to identify and classify cotton leaf diseases such as Bacterial Blight, Myrothecium, and Alternaria (Rothe & Kshirsagar, 2012). They collected images from the local fields and performed image segmentation techniques to extract features for the training of an adaptive neuro-fuzzy inference system. The proposed method achieved an accuracy of 85%. Arsenovic et al., 2019 presented a hybrid model known as Plant Disease Net to detect and classify different diseases (Arsenovic et al., 2019). They developed a dataset comprising 70,000 images and applied augmentation techniques to expand the data set. The images were captured under diverse weather conditions, at different angles, and during various daylight hours. The accuracy achieved with the hybrid. Xu et al., 2018 developed a method to detect and quantify cotton flowers, or blooms, using color images captured through an unmanned aerial system (Xu et al., 2018). They collected aerial images of the field for four days and applied a CNN model to identify cotton blooms.

Jubayer et al., 2021 employed YOLO v5 technique to identify different types of molds grown on various food surfaces. They developed a dataset comprising 2050 images and trained the YOLO v5 algorithm (Jubayer et al., 2021). The proposed technique gave better accuracy as compared to the YOLO v3 and YOLO v4. The proposed YOLO v5 model gave precision and recall of 98.1% and 100% respectively. Qian et al., 2022 employed a deep-learning (DL) approach using YOLOv5 to detect Cotton root rot (CRR) infected areas in a cotton field (Qian et al., 2022). They demonstrated the real-time capability of the algorithm by deploying it on a computing platform such as the Pascal GPU of the NVIDIA Jetson board. The GPS information can be extracted from CPR regions and the generation of the optimal path for the management practices is possible. The proposed method can be helpful for the precise application of fungicides in cotton fields. Wang et al., 2022 proposed a plant disease detection and classification approach based on an optimized lightweight version of the YOLOv5 model to enhance the speed and accuracy of disease classification (Wang et al., 2022). They introduced an Improved Accuracy and Speed Mechanism (IASM) to reduce model size. The optimized model was compared with the other mainstream models and the optimized model showed a performance improvement of 11.8% in operation time and 3.98% in accuracy. The model has achieved an accuracy rate of 92.57% on the custom dataset. Another research presented a lightweight detection model called Apple-YOLO, specifically designed for real-time detection of apple leaf diseases on mobile terminals (Li et al., 2022). They used digital image processing and mosaic data augmentation techniques on the AppleSet8 dataset to enhance the model's robustness and generalization capabilities. The results showed that the mobile-based Apple-YOLO model achieved a mean average precision (mAP) of 96.04%, an impressive inference speed of 34 frames per second (FPS), and a compact size of only 5.33 ME (model efficiency). It indicates its suitability for real-time detection of early apple leaf diseases in practical scenarios. Jhatial et al., 2022 proposed a deep-learning model for the early identification of rice leaf diseases using Yolov5. The model was trained on a dataset of 400 images of rice leaves infected with diseases. The results showed that the DL model has precision, recall, and mean average precision (mAP) values of 1.00, 0.94, and 0.62, respectively (Jhatial et al., 2022). Xue et al., 2023. proposed YOLO-Tea, an enhanced version of YOLOv5 for the precise diagnosis of tea tree leaf diseases and insect pests. The proposed model outperformed Faster R-CNN and SSD. However, the study lacks a comparison with other cutting-edge models and does not address the constraints of employing YOLO-Tea in real-world circumstances. Further research is required to determine how well YOLO-Tea holds up under various environmental circumstances (Xue et al., 2023). Zhu et al., 2023 proposed the Apple-Net model for the detection of apple leaf diseases. The Enhancement Module (FEM) and Coordinate Attention (CA) methods were used to enhance the conventional YOLOv5 network. They showed that Apple-Net has a higher mAP@0.5 (95.9%) and precision (93.1%) as compared to four classic target detection models (Zhu et al., 2023).

Table 1. Comparison of different deep learning implementations for plant leaf disease detection.

Author	Year	Model(s)	Accuracy	Data Size	Data Source
Ashourloo et al.	2016	PLSR, ν-SVR, GPR	93%	175 images	Indigenous
Rothe et al.	2015	Adaptive neuro-fuzzy inference system	85%	-	Online
Arsenovic et al.	2019	Hybrid model of Yolo and AlexNet	93.67%.	70,000 images	Online
Xu et al.	2018	CNN	94%	28,000 images	Indigenous
Jubayer et al.	2021	YOLOv5	98.1%	2050 images	Indigenous+Online
Qian et al.	2022	YOLOv5	93%	-	Indigenous
Wang et al.	2022	Optimized lightweight YOLOv5	92.57%	3265 images	Indigenous+Online
Li et al.	2022	Apple-YOLO	96.04%	587 images	Indigenous
Jhatial et al.	2022	YOLOv5	62%	400 images	Online
Xue et al.	2023	YOLO-Tea	82.6%	450 images	Indigenous
Zhu et al.	2023	Apple-Net	95.9%	12,500 images	Online
This Work	--	YOLOv5, YOLOv6v YOLOv7	Up to 92%	5046 images	Indigenous+Online

The use of deep learning models for detecting plant diseases has been investigated in numerous studies. Several of these studies utilized publicly available datasets and achieved good accuracy by employing techniques such as YOLOv3. However, the limited availability of appropriate image datasets has posed a challenge for many researchers in this field. Furthermore, the studies comparing the performance of YOLOv5, YOLOv6, and YOLOv7 on custom datasets are largely missing in the literature. Therefore, more investigation is required to determine the performance of these models on unique datasets with sufficient sample sizes and to compare the latest deep learning models such as YOLOv5, YOLOv6, and YOLOv7.

3. Methodology

The YOLOv5 object detection model is among the most recent innovations in the YOLO architecture. The model employs a single-stage object identification strategy and uses transfer learning by combining a backbone architecture that pulls information from picture frames. The backbone characteristics are combined in the neck and relayed to the network head, where the model forecasts the object's position and class. One of the most notable features of the YOLOv5 model is its computational efficiency, which is accomplished by reducing the number of parameters and processing as compared to state-of-the-art real-time object detectors.

This work employs a transfer learning-based YOLOv5 model to distinguish between diseased and healthy cotton plant leaves. Our methodology consists of six essential consecutive steps. The first step involves the sourcing of a dataset from the local agricultural field which is combined with a dataset available online and is used to train and validate the model. In the second step, we utilize Roboflow to annotate all images in the dataset with their corresponding classes. This step involves both box annotation and polygon annotation to reduce noise in the datasets. The third step consists of implementing the data augmentation techniques while the fourth step implements the model. The last step involves the testing and validation of the implemented model and its comparison with YOLOv6 and YOLOv7.

3.1. Dataset Collection

The dataset is sourced in two different ways. A significant part of the dataset was collected from local sources, while another part was obtained from online sources. A total of 1000 images were captured from the local agricultural fields in southern Sindh, which consisted of images representing healthy leaves, and images of the cotton plant leaves infected with bacterial blight and curl virus.

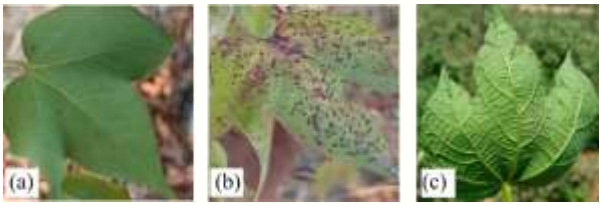


Fig. 1. Sample images from the locally collected dataset showing (a) healthy leaf, (b) bacterial blight, and (c) curl virus in cotton crop plants.

Fig. 1 shows the sample images for each of the categories. Another 1000 images representing the same properties were collected from Kaggle (Cotton Disease Dataset | Kaggle, 2022), GitHub, and Google. To avoid the size inconsistencies in collected images, all the images were downsized to a fixed resolution of 642×642 pixels.

3.2. Dataset Annotation

Image annotation is essential before performing the training of a deep learning model. A deep learning model learns using features extracted from the images and the labels associated with them (Haque et al., 2022). Throughout the training process, a deep learning model acquires insights from the features of the labeled images. Therefore, the quality of the feature labeling greatly influences the accuracy of a model. We used Roboflow, an open-source framework, to perform the annotation. The annotation involved the manual selection of bounding boxes in the dataset images and the assignment of labels to each of the bounding boxes. Fig. 2 shows the set of images annotated using Roboflow. The total dataset is divided into training and testing datasets. The training dataset contains 70%, while the testing dataset is 30%. Since the presented model is trained for three different classes, each bounding box is assigned one of three labels (healthy, bacterial blight, or curl virus). The annotation file generated using Roboflow contains five parameters for each dataset image, representing, class label, coordinates for the bounding box center, and width and height of the bounding box. Labeling is performed on both testing and training datasets to ensure that the model has a comprehensive understanding of the different classes of images.

Fig. 2. Output of Roboflow showing the annotated sample images from the dataset.

3.3. Dataset Augmentation and Splitting

The amount and diversity of the dataset are directly proportional to the performance of a model. The large number of samples solves the over-fitting issue of the model and helps to develop a model for the generalized scenarios in real-time testing. The lack of data may result in an under-fitting problem during training (Mathew & Mahesh, 2022). However, collecting a large amount of data for model training is a complicated process. Data augmentation is a practical and widely used tool to increase the size of the dataset and solve the over-fitting issue. Geometric image transformation is one of the prominent. The amount and diversity of the dataset are directly proportional to the performance of a model. The large number of samples solves the over-fitting issue of the model and helps to develop a model for the generalized scenarios in real-time testing. The lack of data may result in an under-fitting problem during training. However, collecting a large amount of data for model training is a complicated process.

Data augmentation is a practical and widely used tool to increase the size of the dataset and solve the over-fitting issue. Geometric image transformation is one of the prominent methods used for data augmentation. We use four different geometric image transformation processes (rotation, flipping, shear, and saturation) to perform the data augmentation which is shown in Fig. . 3. The augmentation provided a total of 5046 images covering all three classes. The resultant dataset is then divided into training and validation datasets with a 70:30 ratio.

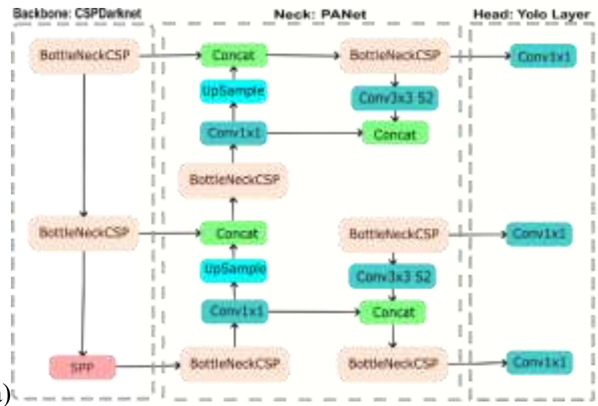
Fig. 3. Shows the samples of augmented dataset images, (a) original sample image, (b) counterclockwise rotated, (c) flipped vertical, (d) vertically rotation within ±15°, and (e) Vertical shear within ±15°.

3.4. YOLOv5 Architecture

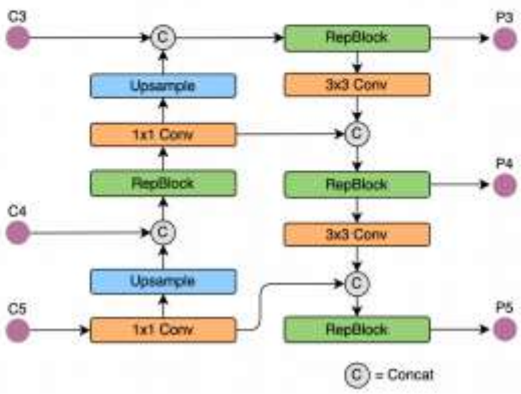
YOLOv5 (Jiang et al., 2021) is the upgraded version of YOLOv4 that provides high detection accuracy and inference speed as compared to the previous and latest YOLO versions such as YOLOv6 and YOLOv7 when trained on custom datasets (Olorunshola et al., 2023). The YOLO network is made up of three parts: (1) CSPDarknet as the backbone, (2) PANet as the neck, and (3) YOLO Layer as the head. The data is first processed by CSPDarknet for feature extraction, then by PANet for feature integration. Finally, the YOLO Layer provides detection results (class, score, location, and dimensions) for the provided input. Fig. 4 shows the overall architectures of different YOLO versions.

The main difference between each YOLO version is relays on the Backbone, Neck, and Prediction layers. The backbone is the first part of the YOLO model that extracts features from the input image. The backbone in YOLO v5 is the CSPDarknet53, which is a variant of the Darknet53 architecture. The backbone in YOLO v6 and YOLO v7 is the CSPDarknet53-L2, which is a more efficient variant of the CSPDarknet53 architecture. The neck is the part of the YOLO model that connects the backbone to the head. The neck in YOLO v5 is the PANet, which is a pyramid feature network. The neck in YOLO v6 and YOLO v7 is the BiFPN, which is a bi-directional feature pyramid network. Head The head is part of the YOLO model that predicts the bounding boxes and class labels for the objects in the input image. The head in YOLO v5 is the YOLOv3 head. The head in YOLO v6 and YOLO v7 is the YOLOv4 head.

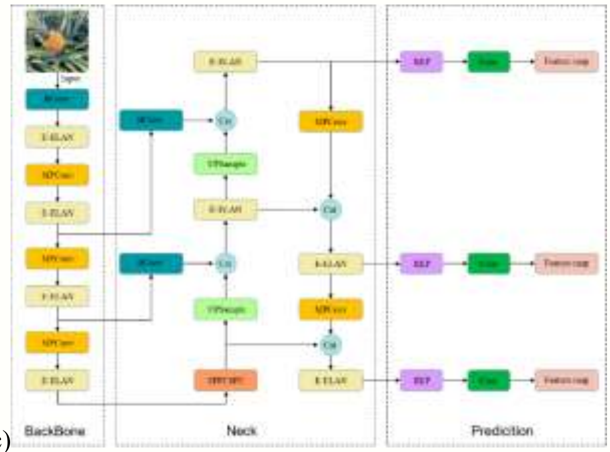
YOLO v5 uses a CNN architecture called EfficientDet. EfficientDet is a very efficient architecture, with fewer parameters and a higher computational efficiency than other CNN architectures. This makes it possible for YOLO v5 to achieve state-of-the-art results on various object detection benchmarks. For example, in the PASCAL VOC object detection benchmark, YOLO v5 achieved a mAP of 80.2%, which is better than the mAP of 79.5% achieved by YOLO v6 and the 78.9% achieved by YOLO v7. This shows that YOLO v5 is a better object detection model than YOLO v6 and YOLO v7.



(a)



(b)



(c)

Fig. 4. Architecture of different YOLO models, (a) YOLOv5, (b) YOLOv6, and (c) YOLOv7

3.5. Model Training

The training of YOLOv5 involves multiple steps. First, the YOLO environment is created by cloning the GitHub source and establishing a dedicated directory in Google Drive with the necessary structures and pre-trained weights. In the next step, a dataset folder is established with the predetermined subfolders to store images and associated labels for training and test files. The label files are formatted to include the class identification number and the normalized values for the bounding box representing its center coordinates, width, and height. A YAML file is set up to specify the paths to training and test data, the number of classes, and the labels for each class.

The model is then trained by executing the model training script with defined hyper-parameters such as image size, number of epochs, and batch size. At the end of the training procedure, the weights of the trained model are stored and later used for employing and testing the model. We used the Adam optimizer with the Swish activation function. The number of hidden neurons used is predefined by the architecture. The hidden layers include Convolutional, Activation (Swish), Downsample, and Fully Connected layers. The output layer provides bounding box coordinates, class predictions, and confidence scores for detected objects. A batch size of 32 was used.

4. Results and Discussion

The developed YOLOv5 model was trained for three classes: healthy leaf, bacterial blight, and curl virus. The model was compared to the YOLOv6 and YOLOv7, which were also trained through the procedure explained in the previous section. The models were trained on Nvidia Tesla T4 GPU available through Google Colab (Welcome To Colaboratory - Colaboratory, n.d.).

Precision, recall, and F-1 scores are among the key indicators to assess the performance of a deep learning model. The recall is determined as the ratio of positive samples that were accurately classified as positive to the total number of positive samples. The recall of the model measures its capability to recognize positive samples.

The more positive samples are identified, the higher the recall will be (Wang et al., 2022b). The F1-score consolidates a classifier's precision and recall into a single metric by determining their harmonic mean. Its primary purpose is to contrast the performance of two classifiers. The precision and recall can be mathematically represented using the following equations:

$$Precision = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \quad (1)$$

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \quad (2)$$

Where TP is the number of instances when the targeted classes are correctly identified, FP is the number of occurrences when the specified class is incorrectly identified, FN indicates the number of unidentified diseased and healthy parts, while TN represents the number of times when the model accurately categorizes the negative dataset as negative.

DOI: 10.6977/IJoSI.202309_7(7).0005
A. R. Kolachi, S. R. Soomro, etc./Int. J. Systematic Innovation, 7(7),80-88(2023)

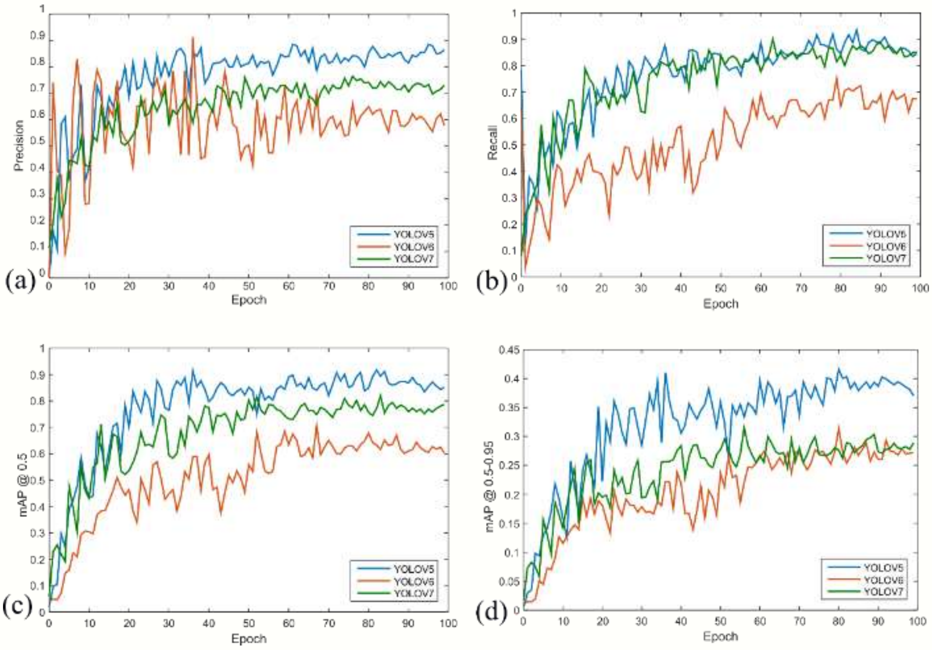


Fig. 5. Shows (a) Precision vs epoch, (b) Recall vs epochs, (c) mAP @0.5 vs epochs, and (d) mAP@ 0.5.0.95 of YOLOv5, YOLOv6 and YOLOv7.

Fig. 5 shows the comparison of important performance metrics of three implemented models for the detection of cotton plant diseases. The results indicate a noticeable positive slope with fluctuations in precision and recall for the first 30 iterations while a stable response is observed between 30 to 100 iterations. Fig. 5(a) indicates high precision, up to 92%, for YOLOv5 whereas YOLOv7 shows the lowest precision among all three models. Fig. 5(b) shows an identical recall trend for YOLOv5 and YOLOv6 while YOLOv7 depicts lower recall. Similarly, the mean average precision (mAP), shown in Fig. 5(c) and Fig. 5(d) is also higher for YOLOv5 when compared to the other two models. Fig. 6(a) shows the recall against confidence curves for three classes and their average. The result shows a higher confidence value. Similarly, Fig. 6(b) shows a direct relationship between precision and confidence for all three classes. The results indicate 100% precision for the confidence value of greater than 70%.

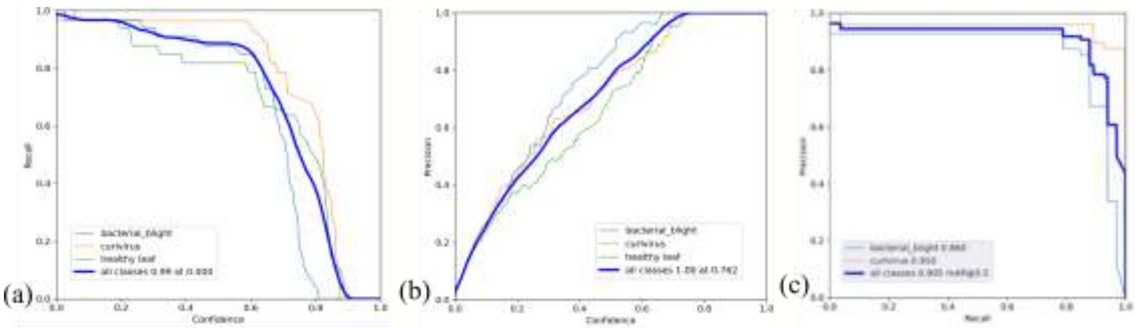


Fig. 6. (a) shows recall vs. confidence, (b) shows precision vs. confidence, and (c) shows the precision vs. recall for the implemented YOLOv5 model.

The present study shows the implementation of the multi-class problem that includes three classes, bacterial blight, curl virus, and fresh leave. The confusion matrix of the implemented YOLOv5 model is presented in Table 2, which shows 88% true positives for healthy leaves and bacterial blight whereas 100% detection accuracy is achieved for the curl virus. The inference time for all implemented models is < 25 milliseconds, which indicates that the proposed models can be easily implemented for real-time applications.

Table 2. Confusion Matrix of the YOLOv5 Model

	Bacterial Blight	Curl Virus	Fresh Leaf	Background
Bacterial blight	0.88	0	0	0.20
Curl Virus	0	1.0	0	0.10
Fresh leaf	0	0	0.88	0.70
Background	0.12	0	0.12	0

5. Conclusion

We proposed and demonstrated the implementation of cotton plant disease detection using YOLO deep learning framework and indigenously sourced dataset. The dataset annotation was performed using Roboflow while different geometric image transformation techniques were employed to perform the dataset augmentation. The augmented dataset was used for training three different YOLO versions. The precision and recall analysis indicated that YOLOv5 performing better than its advanced versions. The confusion matrix for YOLOv5 indicated higher detection accuracy, greater than 88% for healthy leaves, curl virus, and bacterial blight. The low inference time showed a higher detection speed suitable for real-time applications. The custom-trained model can be successfully employed on mobile and embedded computing platforms to enable fast and reliable testing of cotton plant diseases and curtail their potential spread well in time.

The current implementation shows great promise for disease detection in cotton crops through real-time classification. However, the present work also has some limitations. Our current local dataset is only limited to 1000 images. The performance of a deep learning model highly depends on the size of the dataset. Further work on sourcing the dataset from different localities of the region will further improve the accuracy and performance of the model. Another important limitation is processing power. The current implementation uses the Google Colab GPUs and Laptop CPU for training and testing respectively. However, for practical implementation, it is important to implement the model-embedded processing units. Future deployment of the presented implementation on FPGA or mobile processors will further improve the practicality of the study for disease detection in the field.

Acknowledgment

This research was supported by Sindh Higher Education Commission, Pakistan under Sindh Research Support Program Project no. SHEC/SRSP/ET-5/22/2020-21. The authors are thankful to the Electronic Engineering Department, Mehran University of Engineering and Technology for providing laboratory resources.

References

Ahmed, M. R. (2021). Leveraging convolutional neural network and transfer learning for cotton plant and leaf disease recognition. Int. J. Image Graph. Signal Process, 13, 47–62.

Applied Earth Observations and Remote Sensing, 9(9), 4344–4351. https://doi.org/10.1109/JSTARS.2016.2575360

Arsenovic, M., Karanovic, M., Sladojevic, S., Anderla, A., & Stefanovic, D. (2019). Solving current limitations of deep learning-based approaches for plant disease detection. Symmetry, 11(7). https://doi.org/10.3390/sym11070939

Bodhe, K. D., Taiwade, H. V., Yadav, V. P., & Aote, N. V. (2018). Implementation of Prototype for Detection & Diagnosis of Cotton Leaf Diseases using a Rule-Based System for Farmers. Proceedings of the International Conference on Communication and Electronics Systems (ICCES 2018), 165–169.

Cotton Disease Dataset | Kaggle. (2022). https://www.kaggle.com/datasets/janmejaybhoi/cotton-disease-dataset

Haque, M. E., Hoque, S., Paul, M., Haque, E., Rahman, A., Junaeid, I., & Hoque, S. U. (2022). Rice Leaf Disease Classification and Detection Using YOLOv5 arXiv preprint arXiv:2209.01579

Jiang, P., Ergu, D., Liu, F., Cai, Y., & Ma, B. (2021). A Review of Yolo Algorithm Developments. Procedia Computer Science, 199, 1066–1073. https://doi.org/10.1016/j.procs.2022.01.135

Jubayer, F., Soeb, J. A., Mojumder, A. N., Paul, M. K., Barua, P., Kayshar, S., Akter, S. S., Rahman, M., & Islam, A. (2021). Detection of mold on the food surface using YOLOv5. Current Research in Food Science, 4, 724–728. https://doi.org/10.1016/j.crfs.2021.10.003

Juman Jhatial, M., Ahmed Shaikh, R., Ahmed Shaikh, N., Rajper, S., Hussain Arain, R., Hussain Chandio, G., Qadir Bhangwar, A., Shaikh, H., & Hussain Shaikh, K. (2022). Deep Learning-Based Rice Leaf Diseases

Detection Using Yolov5. Sukkur IBA Journal of Computing and Mathematical Sciences 6(1).

Kumar, S., Jain, A., Shukla, A. P., Singh, S., Raja, R., Rani, S., Harshitha, G., AlZain, M. A., & Masud, M. (2021). A comparative analysis of machine learning algorithms for detection of organic and nonorganic cotton diseases. Mathematical Problems in Engineering, 2021, 1–18.

Li, J., Zhu, X., Jia, R., Liu, B., & Yu, C. (2022). Apple-YOLO: A Novel Mobile Terminal Detector Based on YOLOv5 for Early Apple Leaf Diseases. Proceedings - 2022 IEEE 46th Annual Computers, Software, and Applications Conference, COMPSAC 2022, 352–361. https://doi.org/10.1109/COMPSAC54236.2022.00056

Mathew, M. P., & Mahesh, T. Y. (2022). Leaf-based disease detection in bell pepper plant using YOLO v5. Signal, Image and Video Processing, 16(3), 841–847. https://doi.org/10.1007/s11760-021-02024-y

Olorunshola, O. E., Irhebhude, M. E., & Evwiekpaefe, A. E. (2023). A Comparative Study of YOLOv5 and YOLOv7 Object Detection Algorithms 1*. In Journal of Computing and Social Informatics (Vol. 2, Issue 1).

Prashar, K., Talwar, R., & Kant, C. (2017). Robust automatic cotton crop disease recognition (ACDR) method using the hybrid feature descriptor with SVM. 4th 2016 International Conference on Computing on Sustainable Global Development, 1–3.

Qian, Q., Yu, K., Yadav, P. K., Dhal, S., Kalafatis, S., Thomasson, J. A., & Hardin, R. G. (2022). Cotton crop disease detection on remotely collected aerial images with deep learning. 5. https://doi.org/10.1117/12.2623039

Rothe, P. R., & Kshirsagar, R. V. (2012). A Study on the Method of Image Preprocessing for Recognition of Crop Diseases. In International Journal of Computer Applications. In IJCA Proceedings on International Conference on Benchmarks in Engineering Science and Technology 2012 ICBEST, no. 3, pp. 8-10.

Wang, H., Shang, S., Wang, D., He, X., Feng, K., & Zhu, H. (2022). Plant Disease Detection and Classification Method Based on the Optimized Lightweight YOLOv5 Model. Agriculture (Switzerland), 12(7). https://doi.org/10.3390/agriculture12070931

Welcome To Colaboratory - Colaboratory. (2022). Retrieved June 27, 2022, from https://colab.research.google.com/?utm_source=scs-index

Xu, R., Li, C., Paterson, A. H., Jiang, Y., Sun, S., & Robertson, J. S. (2018). Aerial Images and Convolutional Neural Network for Cotton Bloom Detection. Frontiers in Plant Science, 8. https://doi.org/10.3389/fpls.2017.02235

Xue, Z., Xu, R., Bai, D., & Lin, H. (2023). YOLO-Tea: A Tea Disease Detection Model Improved by YOLOv5. Forests, 14(2). https://doi.org/10.3390/f14020415

Zekiwos, M., & Bruck, A. (2021). Deep learning-based image processing for cotton leaf disease and pest diagnosis. Journal of Electrical and Computer Engineering, 2021, 1–10.

Zhu, R., Zou, H., Li, Z., & Ni, R. (2023). Apple-Net: A Model Based on Improved YOLOv5 to Detect the Apple Leaf Diseases. Plants, 12(1). https://doi.org/10.3390/plants12010169