International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 05 | May - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

Automated Plant Disease Detection Using Deep Learning

Dr. Dwiti Krishna Bebarta[1], Gajjarapu Bhavya[2], Tumu Swapna[3], Chadalavada Sri Malvina[4], Gangiredla Asha Deepthi Naidu[5]

[1]Associate Professor, Department of Computer Science and Engineering, Gayatri Vidya Parishad college of Engineering for Women, Visakhapatnam, Andhra Pradesh, India.

[2,3,4,5]Research Scholar, department of Computer Science and Engineering, Gayatri Vidya Parishad College of Engineering for Women, Visakhapatnam, Andhra Pradesh, India.

dkbebarta@gvpcew.ac.in, bhavyagajjarapu143@gmail.com, swapna.tumu2000@gmail.com, srimalvinachadalavada@gmail.com, ashagangiredla@gmail.com

***Abstract*— Traditional methods of diagnosing plant diseases are mainly based on expert diagnosis which easily causes delay in crop disease control and crop management. Due to the problems of many target areas and similar target types in the process of plant disease detection, the identification accuracy and speed are required to be high. Therefore, it is necessary to optimize and improve the existing methods (CNN, RCNN, Fast RCNN, Faster RCNN, and SSD) to meet the detection needs. So, we came up with a deep learning-based approach to identify the plant leaf diseases and classify the diseases using an object detection model called YOLO. There are different versions of YOLO, proposed in recent times, in that YOLOv5 model is considered one of the best models and the other is the recent version, YOLOv8 which was proposed in 2022. So, in this paper we compared the two models of YOLO, YOLOv5 and YOLOv8 on the same dataset, and found that for the dataset used the YOLOv5 model was found to be the better model with a mAP of 0.641, while the YOLOv8 model has mAP of 0.516. This study proposes that the YOLOv5 model is suitable for plant disease identification tasks by comparing it with the latest version of the YOLOv8 model which was proposed in 2022. To make the YOLO model to be better it can be optimized and the transfer learning ability of the model can be used to expand the application scope in the future.**

Keywords*— Deep Learning, CNN, RCNN, Fast RCNN, Faster RCNN, SSD, YOLO, mAP, Transfer Learning, Disease Classification, Detection.

I. INTRODUCTION

Detection of plant diseases is essential to agriculture because it affects crop quality, yield, and overall food security. Pathogen-induced diseases, which include those caused by fungi, bacteria, viruses, and pests, can lead to significant financial losses for farmers and threaten the world's food supplies. To implement timely intervention measures, such as targeted pesticide use, crop rotation, and genetic resistance breeding, early identification and precise diagnosis of plant diseases are essential. Conventional techniques for diagnosing diseases, which rely on visual inspection by human experts, are often labour- intensive, time-consuming, and prone to errors. To enable preventive management methods and reduce production losses, there is an urgent need for automated and effective disease detection systems that can quickly identify and classify illnesses across extensive agricultural landscapes.

A subset of machine learning methods called "deep learning," which draws inspiration from the composition and operations of the human brain, has become a potent tool for automated disease diagnosis in various fields, including agriculture. When it comes to picture classification, object detection, and segmentation tasks, YOLO, a class of deep learning models designed for analyzing visual data, has shown remarkable performance. Deep learning algorithms can identify complex patterns and characteristics from photos of diseased plants, enabling accurate classification and localization of diseases in the field of plant disease detection. Deep learning models have the potential to revolutionize the diagnosis and treatment of plant diseases by leveraging large-scale datasets and computational resources to achieve high levels of accuracy and robust generalization to previously unexplored data.

The cutting-edge object detection system known as YOLOv3 (You Only Look Once version 3) is renowned for its speed and accuracy. Instead of using multiple passes over an image as typically required by object detection algorithms, YOLOv3 employs a single-stage approach. It directly predicts bounding boxes and class probabilities from the entire image in a single feedforward pass. YOLOv3 is well-suited for applications requiring low-latency inference, such as plant disease detection in the field, due to its real-time detection capabilities. The foundation of the YOLOv3 architecture is a deep convolutional neural network, such as Darknet-53, followed by several detection layers that predict bounding boxes and associated class probabilities. YOLOv3 has gained popularity for a variety of computer vision tasks, including object recognition in agricultural applications, due to its effective architecture and exceptional speed.

II. RELATED WORK

The effectiveness of YOLOv2, Faster R-CNN, and SSD, among other CNN architectures, has been thoroughly investigated in previous research on automated plant disease diagnosis. This research has highlighted the significance of large-scale annotated datasets and transfer learning techniques for reliable model training. Furthermore, studies on YOLOv3's streamlined single-stage method and real-time object detection capabilities have demonstrated the platform's potential for rapid disease identification across various plant species and environmental conditions. These investigations have highlighted the speed and accuracy advantages of YOLOv3, paving the way for its application in efficient and successful automated plant disease detection systems in agriculture.

[1] Sachin D. Khirade and A. B. Patil propose utilizing vegetation indices from hyperspectral data, coupled with the RELIEF-F algorithm and artificial neural networks (ANN), achieving classification accuracies ranging from 85.2% to 93.5%. However, a limitation arises due to the challenge of implementing RGB applications, necessitating the conversion of images to grayscale.

[2] Rudresh Dwivedi et al. introduce the Grape Disease Detection Network (GLDDN), leveraging attention mechanisms to achieve an impressive accuracy of 99.93% on a public grape disease dataset, aiming for real-time detection and robotic arm-based solutions.

[3] Monzurul Islam and colleagues focus on potato disease detection through image segmentation and multiclass support vector machines (SVM), achieving a testing accuracy of 95% on the Plant Village Dataset, yet challenges persist in automatically estimating disease severity.

[4] TAN NHAT PHAM et al. propose an early disease classification system for mango leaves using a combination of feed-forward neural networks (CNN) and hybrid metaheuristic feature selection methods, though sensitivity to evolving disease patterns remains a concern.

[5] CHANGJIAN ZHOU et al. introduce a restructured deep residual dense network for tomato leaf disease identification, achieving a balance between accuracy and efficiency, yet concerns arise regarding image quality variations and model reliance on updates.

[6] YUXIA YUAN et al. address crop disease leaf segmentation using a spatial pyramid-oriented encoder-decoder cascade CNN, enhancing segmentation accuracy, but challenges persist with diverse backgrounds and lighting conditions in real-world scenarios.

[7] R. Kavitha Lakshmi and Nickolas Savarimuthu propose PLDD, a deep learning-based system for plant leaf disease detection, utilizing Efficient Set to automatically estimate disease severity.

[8] MAJJI V APPLALANAIDU and G. KUMARAVELAN conduct a comprehensive review of machine learning approaches in plant leaf disease detection and classification, highlighting the efficacy of deep learning models with CNN architecture but emphasizing the need for further research in pest recognition.

[9] LILI LI et al. review plant disease detection and classification methods leveraging deep learning techniques, emphasizing the benefits of using GANs and visualization for enhanced detection, yet challenges remain in model clarity and adaptability to real-world scenarios.

[10] HELONG YU et al. propose a method for corn leaf disease diagnosis utilizing k-means clustering and deep learning, achieving up to 96% accuracy, with future work focusing on optimizing results through swarm intelligence methods.

[11] Sunil C. k. et al. introduce an approach for cardamom plant disease detection using EfficientNetV2, achieving 98.26% accuracy, with potential extensions for identifying disease severity and nutrition deficiencies.

[12] Vibhor Kumar Vishnoi et al. focus on apple plant disease detection using CNN, achieving 98% accuracy on the Plant Village dataset, highlighting the need for more diverse leaf images to improve disease detection.

[13] Emmanuel Moupojou et al. introduce FieldPlant, a dataset of field plant images for disease detection and classification using deep learning, emphasizing the need for more accurate models tailored to field conditions.

[14] Nikolaos Ploskas explores machine learning and deep learning techniques for plant disease classification and detection using datasets comprising apple leaf images, achieving high accuracy but lacking fine-grained spatial details in classification.

[15] Emerson Ajith Jubilson compares multiple deep learning models for plant leaf disease detection and classification, achieving 99.69% accuracy, yet facing limitations such as high parameters and slow detection.

III. METHODOLOGY

The methodology for using YOLO in automated plant disease detection involves systematic steps to train a robust model that allows accurate disease detection in plants from images. It starts with building a diverse dataset of images of healthy plants and plants with various diseases, followed by labelling bounding boxes to indicate disease location and

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 05 | May - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

categories. The YOLO architecture is then configured by selecting the appropriate hyperparameters and adapting the network architecture. Data augmentation techniques increase the diversity of the dataset and the robustness of the model. Appropriate loss functions and optimizers are used for training, and performance is monitored by metrics such as precision and recall. Evaluation with a particular data set assesses the effectiveness and generality of the model. Fine-tuning and optimization can be followed to improve performance or optimize model performance. Finally, the trained YOLO model is used for automatic detection of diseases in real agricultural environments to help in preventive management. This method uses the efficiency and precision of YOLO to promote precision agriculture, which improves crop health and productivity.

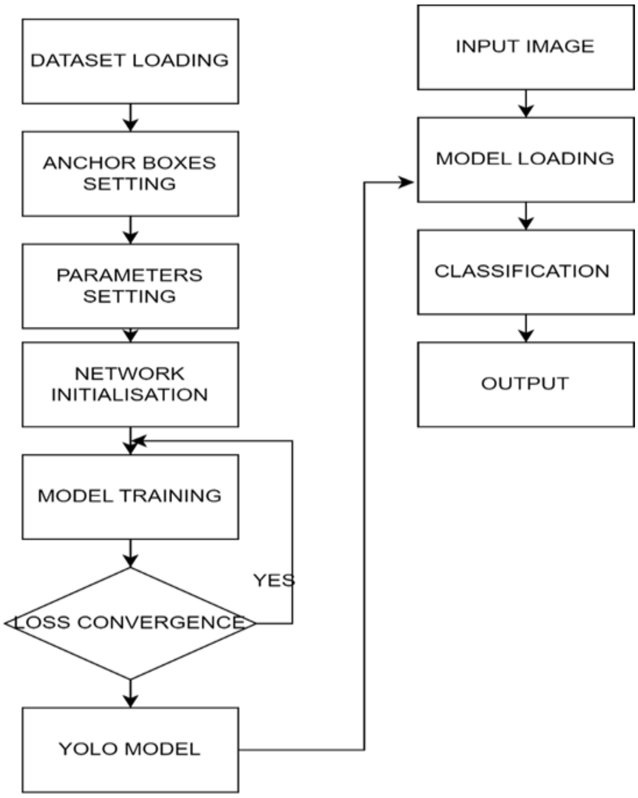

Fig. 1. Architecture

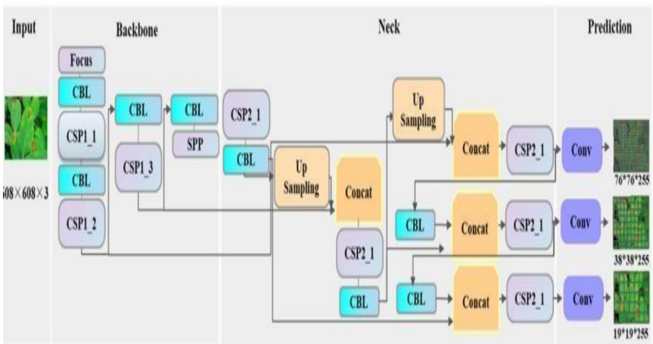

Fig. 2.YOLO Network

A. *Data Collection:*

Dataset helps you to organize unstructured data collected from multiple sources to get the target outcome. Initial data that you give to an algorithm for learning is usually called a training dataset. Training data is a foundation for further development that determines how effective and useful your Machine Learning system will be. However, all initial datasets are flawed and require some preparation before using them for training. For mapping data to the features valuable precisely for your business, you need to label it and make it clean. It will help you exclude useless elements and files, increasing the ML model’s chances of becoming smart. The labelling process used by Exposit usually includes the following steps:

- Data Analysis
- Creation of Data Labelling rules
- Data labelling
- QA Step
- Neural Network training
- Measurement of the output quality

Collecting and labelling images to create a high-quality dataset from scratch requires a lot of resources. If you need to do research or create MVP, you can use publicly available datasets with already labeled data that can include up to 80 categories of different objects. Remember that if you use the same dataset for training, validation, and testing, you won’t be able to evaluate the efficiency of your solution objectively. At Exposit we are more likely to use new and unseen data for testing to ensure excellent performance.

Image Dataset: A well-prepared training dataset drives the quality of your Machine Learning model and effectiveness in fulfilling business purposes. The more quality and accurate results you use for company decision making, the more relevant business strategies you can apply. A good dataset can also help you to save resources on future Machine Learning implementations as you will already have the quality input data. The first and most 44 important stage in training a deep learning model is to gather the necessary images and prepare the dataset on our own, or to select relevant existing datasets for the task and use them. For a Neural Network model, a collection of labeled images as a dataset is used to train, test and assess the performance of the model. Convolutional neural networks are thought to learn from the images in the dataset. The dataset is image-processed before being input into the training module, which is constantly monitored for training accuracy and loss at each epoch. The dataset used for training the model is made up of 2321 RGB images of different plant leaf diseases gathered from the Kaggle website.

Different classes of dataset: Dataset is classified into 30 different classes Apple Scab Leaf, Apple leaf, Apple rust leaf, Bell_pepper leaf spot, Bell_pepper leaf, Blueberry leaf, Cherry leaf, Corn Gray leaf spot, Corn leaf blight, Corn rust leaf, Peach leaf, Potato leaf early blight, Potato leaf late blight, Potato leaf, Raspberry leaf, Soyabean leaf, Squash Powdery mildew leaf, Strawberry leaf, Tomato Early blight leaf, Tomato Septoria leaf spot, Tomato leaf bacterial spot, Tomato leaf late blight, Tomato leaf mosaic virus, Tomato leaf yellow virus, Tomato leaf, Tomato mold leaf, Tomato two spotted spider mites leaf, grape leaf black rot, grape leaf. Some of the images in the dataset are shown in fig.8.1 1. Apple Scab Leaf – 158 images 2. Apple leaf – 232 images 3. Apple rust leaf – 167 images 4. Bell_pepper leaf spot – 312 images 5. Bell_pepper leaf – 248 images 6. Blueberry leaf – 796 images 7. Cherry leaf – 218 images 8. Corn Gray leaf spot – 72 images 9. Corn leaf blight – 356 images 10. Corn rust leaf – 117 images 11. Peach leaf – 579 images 12. Potato leaf early blight – 11 images 13. Potato leaf late blight– 301 images 14. Potato leaf – 235 images 15. Raspberry leaf – 539 images 45 16. Soybean leaf- 246 images 17. Soybean leaf -15 images 18. Squash Powdery mildew leaf – 243 images 19. Strawberry leaf – 438 images 20. Tomato Early blight leaf – 193 images 21. Tomato Septoria leaf spot – 402 images 22. Tomato leaf bacterial spot – 373 images 23. Tomato leaf late blight – 266 images 24. Tomato leaf mosaic virus – 204 images 25. Tomato leaf yellow virus - 225 images 26. Tomato leaf - 759 images 27. Tomato mold leaf – 279 images 28. Tomato two spotted spider mites leaf– 2 images 29. grape leaf black rot – 201 images 30. grape leaf – 125 images.

B. Dataset Collection and Preprocessing:

The initial step involves gathering a diverse dataset comprising images of healthy plants and plants affected by various diseases. Each image is annotated with bounding boxes to indicate the location and classification of the disease. Preprocessing techniques are applied to standardize the image sizes, formats, and quality, ensuring uniformity across the dataset.

C. Model Configuration:

YOLO's architecture is configured by selecting appropriate hyperparameters and adjusting the network architecture. This involves determining the number of anchor boxes and detection layers based on the characteristics of the dataset. Fine-tuning may involve optimizing parameters such as the learning rate and batch size to enhance the model's performance during training.

D. Data Augumentation:

Data augmentation is crucial for enriching the training dataset and improving the model's generalization ability. Techniques such as random rotation, flipping, scaling, and brightness adjustment are applied to introduce diversity and robustness to the training data, helping the model learn invariant features across different conditions.

E. Training:

YOLO is configured and is trained on the augmented dataset suitable appropriate loss functions and optimizers. During training, batches of images are fed through the network, and the model's weights are updated iteratively to minimize the loss and improve performance. Common loss functions include binary cross-entropy or focal loss, while popular optimizers include Adam or stochastic gradient descent (SGD).

F. Evaluation:

After training, trained YOLO model is evaluated on a separate validation dataset to assess its performance and generalization ability. Evaluation metrics such as precision, recall, and F1-score are computed to quantify the accuracy of the model in detecting plant diseases. Qualitative evaluation involves visually inspecting the model's predictions overlaid on test images to identify any discrepancies or areas for improvement.

G. Fine-tuning and Optimization:

Based on the evaluation results, fine-tuning and optimization steps may be performed to further improve the model's performance or optimize its efficiency for deployment. This may include adjusting hyperparameters, exploring different loss functions or optimizers, or fine-tuning the model architecture. Techniques such as model ensembling, pruning, and quantization may also be employed to enhance model efficiency and resource utilization.

H. Deployment:

Finally, the trained and optimized YOLO model deployed in real-world agricultural settings for automated plant disease detection. Integration with user-friendly interfaces enables farmers and agricultural experts to interact with the system, receiving actionable insights and recommendations based on disease detection results. Continuous monitoring and feedback mechanisms ensure the reliability and effectiveness of the deployed system, facilitating proactive management practices and enhancing crop health and productivity.

IV. EXPERIMENTAL RESULTS

Results and Analysis for YOLO

The results obtained from implementing YOLO for object detection provide valuable insights into the model's performance in accurately detecting and localizing objects within images. Through a detailed analysis of classification and localization metrics, such as precision, recall, and Intersection over Union (IoU), the effectiveness and robustness of the YOLO model can be thoroughly assessed.

Fig. 3. Disease of Potato Leaf

Fig. 4. Disease of Apple Leaf

Fig. 5. Soyabean Leaf

Localization Performance with YOLO

The localization performance of YOLO evaluates the accuracy of the model in predicting the coordinates of bounding boxes around detected objects. The Intersection over Union (IoU) metric is particularly critical for evaluating YOLO's localization performance because it quantifies the overlap between the predicted and ground truth bounding boxes. High Intersection over Union (IoU) values indicate that YOLO accurately localizes objects within images, with the predicted bounding boxes closely aligning with the ground truth.

Analysis of YOLO Results

Analyzing the results obtained from YOLO enables a comprehensive assessment of its performance and efficacy. High precision and recall values in classification signify YOLO's proficiency in accurately identifying objects and assigning them to their respective classes. Furthermore, high Intersection over Union (IoU) values indicate the accuracy of YOLO in localizing objects within images. However, areas for improvement may be identified through detailed analysis, such as instances of false positives or false negatives. These insights can inform further refinement of the YOLO model, guiding adjustments to optimize its performance and address any shortcomings. Overall, the results and analysis of YOLO provide valuable feedback for assessing its effectiveness in object detection tasks and guiding future enhancements to maximize its utility and accuracy.

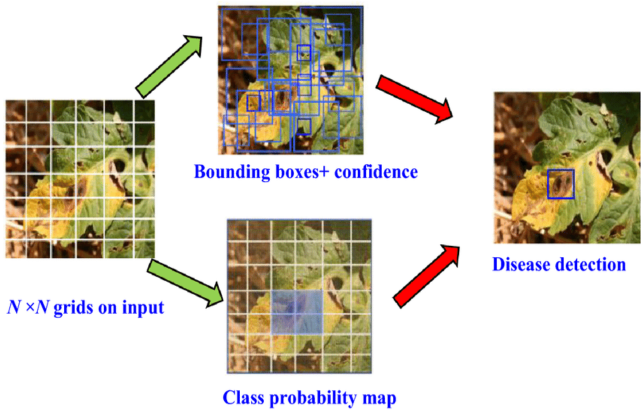

Fig. 6. Working of YOLO Algorithm

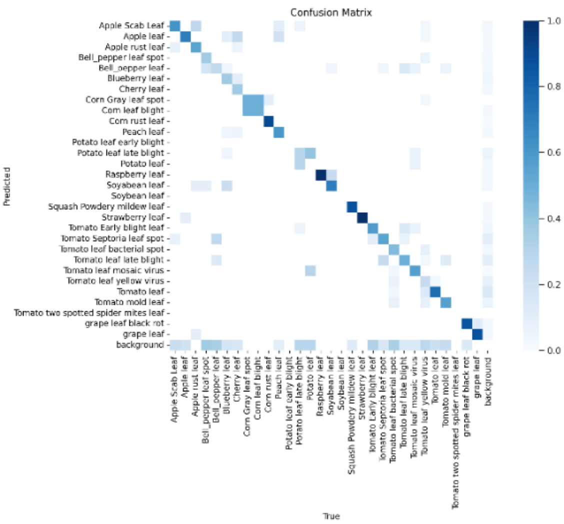

Fig. 7. Confusion Matrix for built YOLOv8 model

V. CONCLUSION

Crops and plants are an important part of agriculture and must be protected. This requires a thorough knowledge of the plant species cultivated and the possible diseases that the plants may suffer. To achieve the desired results in the research, we developed an automated disease detection model that uses image processing techniques such as enhancement, segmentation, feature extraction and classification. The YOLO network was used to determine whether the leaves of the plant were affected or not, which proved to be more accurate. These technologies help farmers detect diseases at an early stage so that they can monitor and warn against them. In order for every farmer to benefit from this approach, the research domain will be expanded to include more classifiable diseases in the future.Due to the increased size of the dataset, it is necessary to compare the YOLOv5 and YOLOv8 models. The YOLOv5 model is optimized as a lightweight YOLOv5 model and can then be compared to the latest YOLOv8 model. Transfer learning must be enabled to allow the model to adapt to new changes.Once a lightweight version of YOLOv5 is obtained, it is compared with the latest YOLOv8 model on several performance parameters, including accuracy. , speed of reasoning and use of resources. Taking into account variables including model complexity, computational needs, and realistic implementation considerations, this comparison attempts to determine whether the model is better suited to handling a larger data set.

VI. FUTURE SCOPE

The validation process involved evaluating the effectiveness of the project against the original requirements and objectives. This process allowed the development team to identify areas that needed improvement and optimize the project accordingly. The test and validation metric is important because it ensures that the project will work as planned, meet initial requirements and be ready for deployment. In addition, the test and validation metrics are a roadmap for future development and allow developers to identify areas that need improvement and optimize the project accordingly. The test and validation metrics are generally a critical part of the project development process. This ensures that the project meets the initial requirements, works as planned and is ready for use. The testing and validation chapter also provides a roadmap for future development, ensuring that the project continues to meet the changing needs of its users.

Going forward, the future of YOLO object recognition offers a mature landscape with development and application opportunities in many different fields. Going forward, researchers and developers are ready to explore new ways to improve YOLO's capabilities, focusing on improving model architectures to achieve superior performance in object detection in complex environments. In addition, there is a growing demand for fine-grained object detection capabilities, which could be addressed by further research on adapting YOLO to extract complex object details. Real-time video object tracking is another promising area for future research, as it uses YOLO's high-speed processing capabilities to track objects seamlessly between video frames. In addition, adapting YOLO to domain-specific tasks such as agriculture or healthcare offers significant opportunities to optimize model performance and respond to the unique challenges of specialized domains. As the field progresses, attention will also be directed towards enhancing YOLO's robustness to adversarial attacks, improving interpretability and explainability, and optimizing deployment on edge devices for efficient real-world applications. By delving into these avenues for future research and development, YOLO is poised to remain at the forefront of object detection technology, driving innovation and enabling transformative advancements in computer vision and beyond.

VII. REFERENCES

[1] Wu, N.; Weng, S.; Chen, J.; Xiao, Q.; Zhang, C.; He,Y. Deep convolution neural network with weighted loss to detect rice seeds vigor based on hyperspectral imaging under the sample-imbalanced condition. Computer.Electron. Agric. 2022, 196, 106850

[2] Xu, W.; Zhao, L.; Li, J.; Shang, S.; Ding, X.; Wang, T. Detection and classification of tea buds based on deep learning. Computer. Electron. Agric. 2022, 192, 106547.

[3] Joseph Redmon∗ , Santosh Divvala∗, Ross Girshick , Ali Farhadi∗University of Washington∗ , Allen Institute for AI† , Facebook AI Research, You Only Look Once: Unified, Real-Time Object Detection.

[4] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K.I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. International Journal of Computer Vision, 111(1):98–136, Jan. 2015.

[5] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. CoRR, abs/1409.4842, 2014.

[6] Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 779–788.

[7] Plant Disease Detection and Classification Method Based on the Optimized Lightweight YOLOv5 Model Haiqing Wang, Shuqi Shang *, Dongwei Wang, Xiaoning He, Kai Feng and Hao Zhu.

[8] Zheng, Q.; Chen, Y. Interactive multi-scale feature representation enhancement for small object detection. Image Vis. Comput. 2021, 108, 104128.

[9] S. D. Khirade and A. B. Patil, "Plant Disease Detection Using Image Processing," *2015 International Conference on Computing Communication Control and Automation*, Pune, India, 2015, pp. 768-771, doi: 10.1109/ICCUBEA.2015.153.

[10] Y. Yuan, Z. Xu and G. Lu, "SPEDCCNN: Spatial Pyramid-Oriented Encoder-Decoder Cascade Convolution Neural Network for Crop Disease Leaf Segmentation," in *IEEE Access*, vol. 9, pp. 14849-14866, 2021, doi: 10.1109/ACCESS.2021.3052769.

[11] C. Zhou, S. Zhou, J. Xing and J. Song, "Tomato Leaf Disease Identification by Restructured Deep Residual Dense Network," in *IEEE Access*, vol. 9, pp. 28822-28831,2021,doi: 10.1109/ACCESS.2021.3058947.

[12] L. Li, S. Zhang and B. Wang, "Plant Disease Detection and Classification by Deep Learning—A Review," in *IEEE Access*, vol. 9, pp. 56683-56698, 2021, doi: 10.1109/ACCESS.2021.3069646.

[13] T. N. Pham, L. V. Tran and S. V. T. Dao, "Early Disease Classification of Mango Leaves Using Feed-Forward Neural Network and Hybrid Metaheuristic Feature-Selection,"in *IEEEAccess*,vol.8,pp.189960-189973,2020,doi: 10.1109/ACCESS.2020.3031914.

[14] R. Dwivedi, S. Dey, C. Chakraborty and S. Tiwari, "Grape Disease Detection Network Based on Multi-Task Learning and Attention Features," in *IEEE Sensors Journal*, vol. 21, no. 16, pp. 17573-17580, 15 Aug.15, 2021, doi: 10.1109/JSEN.2021.3064060.

[15] E. Moupojou *et al*., "FieldPlant: A Dataset of Field Plant Images for Plant Disease Detection and Classification With Deep Learning," in *IEEE Access*, vol. 11, pp.35398-35410, 2023,doi:10.1109/ACCESS.2023.3263042.

[16] H. Yu *et al*., "Corn Leaf Diseases Diagnosis Based on K-Means Clustering and Deep Learning," in *IEEE Access*, vol. 9, pp. 143824-143835, 2021, doi: 10.1109/ACCESS.2021.3120379.

[17] V. Balafas, E. Karantoumanis, M. Louta and N. Ploskas, "Machine Learning and Deep Learning for Plant Disease Classification and Detection," in *IEEE Access*, vol. 11, pp. 114352-114377, 2023, doi: 10.1109/ACCESS.2023.3324722.

[18] R. K. Lakshmi and N. Savarimuthu, "PLDD—A Deep Learning-Based Plant Leaf Disease Detection," in *IEEE Consumer Electronics Magazine*, vol. 11, no. 3, pp. 44-49, 1 May 2022, doi: 10.1109/MCE.2021.3083976.

[19] C. Madhurya and E. A. Jubilson, "YR2S: Efficient Deep Learning Technique for Detecting and Classifying Plant Leaf Diseases," in *IEEE Access*, vol. 12, pp. 3790-3804, 2024, doi: 10.1109/ACCESS.2023.3343450.