2024 3rd IEEE International Conference on Artificial Intelligence for Internet of Things (AIIoT 2024)

A Cutting Edge Deep Learning Models for Paddy Leaf Disease Detection and Classification

A .Dhanalakshmi
Department of Computer Applications
Kalasalingam Academy of Research and Education
KrishnanKovil,Tamil Nadu , India
dhanuswathi020@gmail.com

K.Ponmozhi
Department of Computer Applications
Kalasalingam Academy of Research and Education
KrishnanKovil,Tamil Nadu , India
ponmozhi@klu.ac.in

A.Ajmal
Department of Data Science
Vellore Institute of Technology(ICER)
Bangalore,Karanataka, India
ajmalimra.a@gmail.com

2024 3rd International Conference on Artificial Intelligence For Internet of Things (AIIoT) | 979-8-3503-7212-0/24/$31.00 ©2024 IEEE | DOI: 10.1109/AIIoT58432.2024.10574681

Abstract— **The world's main grain crop is rice (Oryza sativa). It provides the majority of the world's population with nourishment and energy. Numerous rice leaf diseases are a major threat to the world's agricultural output because they deteriorate crop quality in addition to lowering crop yield. Traditional methods of classifying and managing diseases mostly rely on laborious and subjective manual examination procedures. However, recent advancements in computer vision methods have made it possible to construct automated systems that can recognize illnesses from picture and video data. The paper emphasizes the potential application of cutting-edge models like Detectron2 to address issues with agricultural leaf disease detection. Together with other cutting-edge models, Detectron2 provides a promising means of enhancing the accuracy and effectiveness of disease diagnosis and control techniques. The use of automated disease detection systems holds the potential to reduce subjectivity in the evaluation of illnesses, optimize the allocation of resources in agriculture, and reduce dependence on human labour. These systems are able to continually learn from and adapt to changing disease patterns by incorporating complex algorithms and machine learning techniques, ensuring robust and stable agricultural operations.**

Keywords— Detectron2, Deep learning, Rice leaf diseases

I. INTRODUCTION

A significant part of the Indian economy, agriculture comes in second in terms of rice output. Nearly every state in India—Tamil Nadu, West Bengal, Punjab, Uttar Pradesh, Assam, Bihar, etc. Practices rice production. About 19.9% of the gross domestic product comes from the agriculture industry. In India, rice is one of the main grains consumed. Diseases have an impact on the development and quality of rice plants, which further suggests that cultivation is profitable. Different types of illnesses can affect specific rice crops, and farmers may find it challenging to detect them due to their limited experience and understanding. An expert system that processes automated data is essential for the accurate and timely detection of plant diseases. As a result, the cultivation is successful and healthy [1]. In order to address a variety of problems in agriculture, including fruit counting, root segmentation, plant disease classification, weed and seed identification, and more, the powerful deep learning algorithm has been introduced [2, 3]. A significant quantity of data is successfully trained using deep learning, an improvement in machine learning techniques that generates output based on decision rules and automatically learns the characteristics of the input. In order to understand the spatial connection of the input data and complete a classification assignment, CNN is good at processing visual information. A convolutional neural network that has already been trained can be used for a new task using a technique called transfer learning. This results in a shorter training period for the model than a model created from scratch and improves the suggested model's performance. By removing the last fully connected layers or optimising the final few layers to be more relevant to the dataset in question, transfer learning may be used to build a model that can be used as a fixed feature extractor. The deep learning model has been fine-tuned to achieve increased accuracy in classifying the many types of illnesses that affect rice leaves [4].

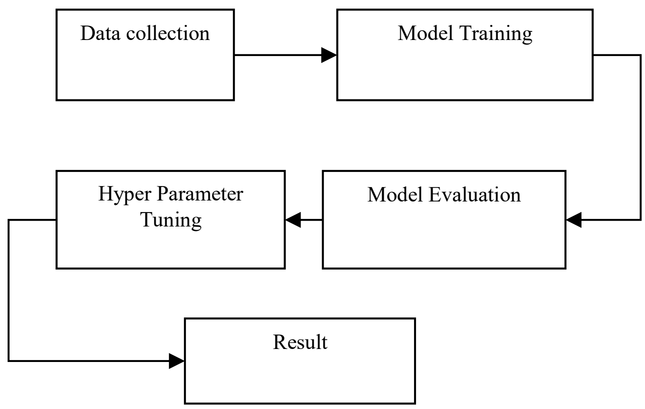

Figure 1: Flow chart

II. RELATED WORK

This section covers the other earlier research on utilising a deep learning algorithm to forecast leaf

979-8-3503-7212-0/24/$31.00©2024 IEEE

Authorized licensed use limited to: VIT University- Chennai Campus. Downloaded on July 03,2024 at 05:58:21 UTC from IEEE Xplore. Restrictions apply.

illnesses. The publication describes how to identify rice disease using agro-meteorological data and leaf disease using data collecting methods, specifically using MLP architecture. CNN, the algorithm utilised in the study, has a 95% prediction rate[5]. A approach for data preprocessing and data augmentation for the detection of plant diseases is presented in [6]. The process is driven by the CNN algorithm, which has a 93% accuracy rate. Last but not least, ResNet 50 offers 66% accuracy while ResNet 101 offers 70% accuracy. Furthermore, utilising object identification models such as Mask CNN, annotated photographs from the dataset will be able to be used in future research to detect sickness severity levels. While VGG 16 delivers 89%, Double Gan is the algorithm that produces 86% [7]. The limitation of the research is Subsequent investigations will concentrate on broadening the dataset to encompass more illnesses and difficult photographs, as pictures with an abundance of leaves can be incorrectly categorised. The techniques for identifying plants and pests, which involve gathering data, preprocessing data associated with the images, and using CNN and VGG 16 algorithms. Both methods give 95% of the accuracy [8]. The primary limitation of the research is the lack of fine-grained data, such as geographic features or the capacity to handle several items simultaneously. Wheat leaves are the main focus of crop disease diagnosis, together with a particular crop's preprocessing, support and query sets, picture encodings, and attention block are among the methods used [9]. Subsequent research ought to concentrate on enlarging the proposed system to encompass other plants and crops.

For the detection of plant diseases and pests, techniques such as data synthesis and production, transfer learning and fine-tuning classical network model, fine-grained identification, and data amplification are used [10]. Utilising algorithms like Resnet and GooLeNet structure, which provide results with 97 percent accuracy, restricts the use of deep learning methods for identifying plant diseases and pests. In order to achieve more, attention processes may be utilised to effectively choose information and allocate more resources towards areas of interest in the project's three alternatives for small-scale and long-term research on the early detection of plant diseases and pests. Techniques for analysing fruit plants, such as peaches and strawberries, include fine-grained identification, transfer learning, and the amplification, synthesis, and production of data. The accuracy of the L1-ELM (Extreme Learning Machine) approach is 98.5% [11].The limitation was not many labelled subjects are accessible. LSTM algorithm which has an 84% accuracy rating is used to forecast the data first. Yolo v5 and v7 are employed in [12] using an approach centered on transfer learning and SVM data augmentation. The dataset only contains three illnesses, which is a limitation of the work. But in order to give more illness forecasts, improve up to six diseases in the future. The following technique is used to explain the fruit plant and crop plant dataset: pretrained deep models, features, CNN, extraction, image processing, leaf image, disease diagnosis, pesticide recommendation, and Conventional neural network, an algorithm that achieves 93% accuracy [13]. In [14] model's task is to identify leaf diseases; to this end, they make use of the "plant village" dataset. The Gated self-Attentive Convoluted MobileNetV3 (GSAtt-CMNetV3) model yields 97% accuracy. The work's downside is the image's incapacity to distinguish several leaves because to the disease's poor spotting. The method used in this work has the potential to facilitate the segmentation and classification process. Further, the Field Plant dataset to identify leaf diseases [15]. In their model, they use CNN and MobileNet, which offer 90% accuracy. One of the benefits of this approach in the job is its effective classification of leaf diseases. One limitation of the work is the extremely poor accuracy of illness detection, which prevents from correctly classifying a large number of leaves. The comparative study of recent studies is shown in table 1.

Table 1: Comparative study

Author	Year	Aim	Dataset & Method used	Acc
[12]	2023	Rice leaf disease detection	Rice leaf Diseases Yolo v5	76%
[9]	2023	Wheat Disease Classification	CGIAR Crop Disease Dataset Efficient Net Model	93%
[15]	2023	Leaf Diseases Detection	Field plant dataset CNN, MobileNet	90%
[6]	2024	Plant Disease Detection	Real time plant disease dataset CNN,ResNet50, ResNet101 models are Used.	93%,66%, 70%
[14]	2024	Leaf Diseases Detection	Plant Village dataset Gated Self-Attentive Convoluted MobileNetV3(GSAtt-CMNetV3)	96%

III. MATERIALS AND METHODS

As indicated in Fig. 1 and Fig. 2, this section breaks down the planned work's technique into five parts for rice leaf disease classification.

A. Data Collection

The rice leaf picture dataset utilised in this experiment was obtained via the Google Images website and the Kaggle website. The collection includes 3355 photos representing the healthy group and three categories of illnesses, including bacterial blight, brown spot, and leaf blast. Every picture contains a single disease, although some images have many diseases. The quantity of photos in the training, testing, and validation sets is the same.

B. Model training

Configuration Establishment:

Choose the architecture and configuration file: Select the appropriate configuration file and the model's architecture (such as Mask R-CNN or Faster R-CNN). These configuration files define the hyperparameters, ROI

Authorized licensed use limited to: VIT University- Chennai Campus. Downloaded on July 03,2024 at 05:58:21 UTC from IEEE Xplore. Restrictions apply.

a.Brownspot b.Hispa c.Healthy

Figure 2: Samples of rice crop leaf images.

(Region of Interest) heads, backbone network, region proposal network (RPN), and other parameters. Get the dataset ready: Verify that the formatting requirements for Detectron 2 are met by your dataset. This usually means placing the dataset in a specified directory structure with functional annotations (such bounding boxes and masks) in a COCO JSON format.

C. *Activate the dataset:*

Register the dataset with Detectron2 by using the Dataset Catalogue and Metadata Catalogue. Detectron2 may thus access and use the dataset during training.

Construct the model: Build_model(cfg) is used to create an instance of the model that is given in the configuration file.Set the model weights at zero: Either start from scratch or use weights that have already been trained to initialize the model. With cfg, pre-trained weights can be loaded WEIGHTS MODEL Set the trainer's initials: Create an instance of the DefaultTrainer class with the specified parameters to begin the training process.

D. *Begin of training:*

To begin the training process, call the trainer's train() method. The trainer iterates over the dataset, feeds the model batches of data, computes the loss, and modifies the model's parameters using an optimisation strategy (such SGD or Adam).

E. *Analysis the model:*

Utilise measures like as accuracy, mean average precision (mAP), and others to evaluate the model's performance on a validation or test set after training. The Detectron2 classes COCO Evaluator and Detection Evaluator can be used for evaluation.

F. *Examine the outcome:*

Examine the evaluation results in order to determine areas that want improvement and to comprehend the model's advantages and disadvantages.

G. *Fine-tune the model:*

It takes further training on certain datasets or tasks to improve the model's performance. To do this, a variety of optimization strategies, regularization techniques, and hyperparameter tweaks may be applied.

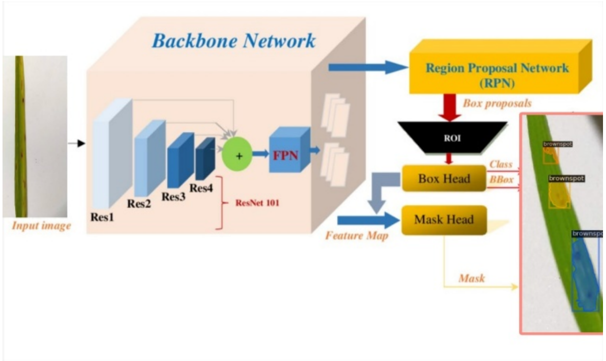

Figure 3: Architecture of Detectron2

IV. METHODOLOGY

Through the use of the present deep learning model to analyse pertinent studies. Detectron 2, a recently developed algorithm, is effective in correctly categorizing illnesses. On the other hand, detectron 2 operates at maximum iterations, displaying the usefulness of this model in a very short period of time. Other deep learning algorithms operate in epochs, so it takes several hours to reveal the result. In contrast, the method runs on the CPU in just 45 minutes.

A. *Hyper Parameter Tuning*

The Detectron2 hyperparameter tuning strategy involves changing certain parameters that are not learnt during training in order to enhance the model's performance. They include things like learning rate, batch size, iteration count, and more. The code snippet you offer sets some of these hyperparameters for instance segmentation model training using the Mask R-CNN architecture. Regarding instance segmentation, the model's architecture is determined by this hyperparameter. This example employs a Mask R-CNN architecture with a ResNet-101 backbone and a Feature Pyramid Network (FPN) with a 3x training schedule. This line creates the path to the configuration file (.yaml), based on the architecture that has been chosen. A number of the hyperparameters and settings required to train the model are included in the configuration file, such as learning rate, ROI heads,and backbone network. This hyperparameter controls how frequently the model is evaluated on the validation set during training. In this case, evaluation is done every 200 iterations. Frequent model assessments help monitor the model's performance and spot potential issues like under- or overfitting. This hyperparameter determines the training base learning rate of the model. The learning rate dictates the step size at which the model parameters are changed during optimization. While lower learning rates can result in more stable training but may require more iterations to attain convergence, higher learning rates have the potential to cause instability but can potentially accelerate convergence. A hyperparameter determines the number of classes in the dataset. It determines the output dimension of the model's final classification layer. This parameter must be set to match the number of classes in the dataset in order for the dataset to be correctly trained and evaluated. Hyperparameter tweaking involves testing

Authorized licensed use limited to: VIT University- Chennai Campus. Downloaded on July 03,2024 at 05:58:21 UTC from IEEE Xplore. Restrictions apply.

with various values to find the combination of these parameters that offers the model the best performance on the given task and dataset. To find the best configuration, this process often requires iterative testing, training many models with different hyperparameter values, and performance evaluation.

V. IMPLEMENTATION OF LEAF DISEASE DETECTION

A. *Dataset Description*

The Data is hand collected from various websites with each and every label verified and sourced Kaggle.

- 3355 Images

B. *Pre-processing*

Using an image dataset requires little preparation on our part. We may say that one of the preprocessing techniques we use to increase our accuracy is data augmentation.

Data Augmentation: By providing the model with a wider variety of training instances, data augmentation enhances the model's capacity to generalise to new data and mitigates overfitting. It is particularly helpful when there is a limited quantity of dataset available or when the intricacy of the job causes the model to overfit. Reliable and noteworthy findings about leaf diseases Effective data preparation techniques are necessary for prediction.

C. *Models*

Conventional neural Network (CNN): One well-liked type of deep learning models for image analysis and recognition is called convolutional neural networks, or CNNs. Convolutional layers are used to extract features from input pictures, and pooling layers are used to reduce dimensionality. CNNs are adept at tasks like object detection, classification, and segmentation in images because they lower prediction errors by modifying their parameters through forward and backward propagation. CNNs use supervised learning, in which they are trained on labelled datasets to find connections.

VGG 16: The 16-layer deep convolutional neural network architecture, or VGG16, combines convolutional and fully linked layers. It stands out for being both profound and basic. It is trained on large image datasets such as ImageNet using gradient descent and backpropagation, and it employs supervised learning to extract hierarchical features from pictures. VGG16 convolves input pictures using varying-sized filters in order to extract features. Then, max-pooling layers are utilised to lower dimensionality, and fully linked layers are employed for classification or regression tasks.

Mobile Net V2: MobileNetV2, a lightweight convolutional neural network architecture, is notable for its low computational complexity and excellent accuracy. It is designed to be deployed effectively on mobile and embedded devices. It applies techniques like depth-wise separable convolutions and inverted residuals to minimise parameters and processing costs without compromising performance. It uses supervised learning, often with labelled datasets. MobileNetV2 applies batch normalisation, ReLU activations, depth-wise separable convolutions, and linear bottlenecks in that sequence to enable efficient feature extraction and classification on platforms with constrained resources.

ResNet 50: A deep convolutional neural network architecture called ResNet, short for Residual Networks, was created to solve the disappearing gradient issue during training. It makes use of labeled datasets and supervised learning, concentrating on residual mappings rather than the intended underlying translation. In order to improve convergence and preserve performance, ResNet uses shortcut connections, also known as skip connections, to facilitate gradient flow through the network. This allows for the training of extremely complex neural networks with hundreds of layers.

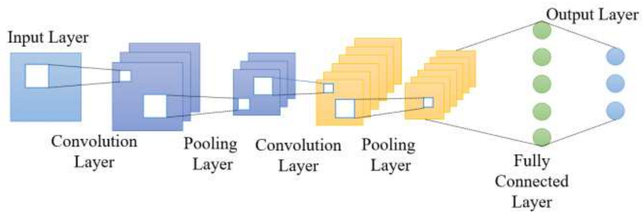

Figure 4: Architecture of CNN

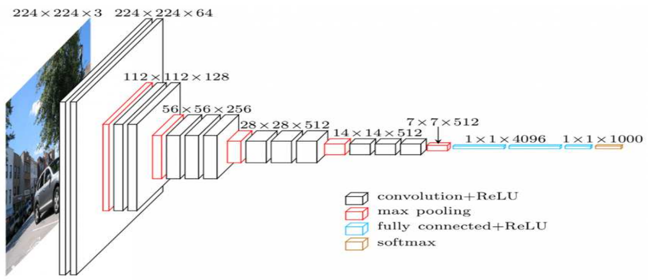

Figure 5: Architecture of VGG 16

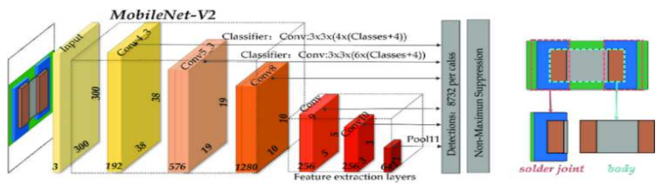

Figure 6: Architecture of Mobile Net v2

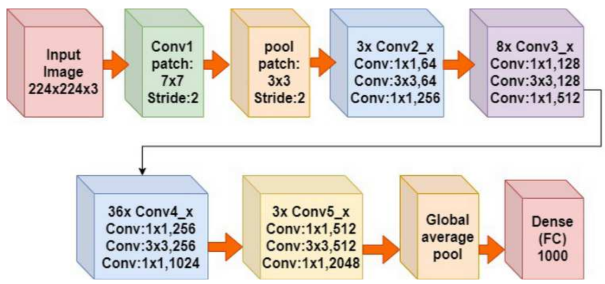

Figure 7: Architecture of Resnet 50

Authorized licensed use limited to: VIT University- Chennai Campus. Downloaded on July 03,2024 at 05:58:21 UTC from IEEE Xplore. Restrictions apply.

Inception V3: Using supervised learning, Inception V3 is a deep convolutional neural network architecture. It effectively records spatial hierarchies by employing inception modules with different kernel sizes, which helps with feature extraction and classification tasks.

DenseNet 121: DenseNet 121 is the name of a dense convolutional neural network design that uses supervised learning. By feed-forwardly linking each layer to every other layer, it improves gradient flow, promotes feature reuse, and produces compact, very parameter-efficient models.

Xception: Xception is a supervised learning deep convolutional neural network architecture built on the depthwise separable convolution operation. It uses depthwise convolutions to capture spatial and channel-wise correlations, which reduces computational complexity without compromising expressiveness for photo recognition applications.

DenseNet and Xception: Ensemble models combining the DenseNet and Xception architectures employ supervised learning. By leveraging the complementary characteristics of both networks, they enhance feature representation and generalisation, which typically leads to improved results on difficult picture recognition and classification tasks.

EfficientNet B3. In order to optimize resource efficiency, supervised learning was used to construct a class of convolutional neural network architectures known as EfficientNet B3. Even with constrained processing power, the network provides state-of-the-art performance on a range of computer vision applications because to dynamic scaling of its depth, breadth, and resolution.

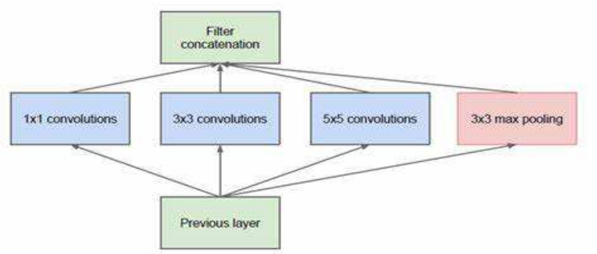

Figure 8: Architecture of Inception V3

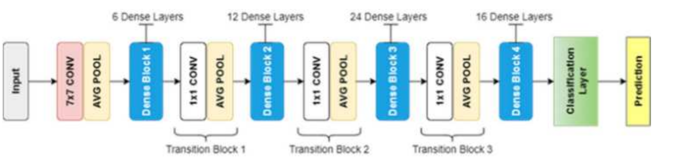

Figure 9: Architecture of DenseNet 121

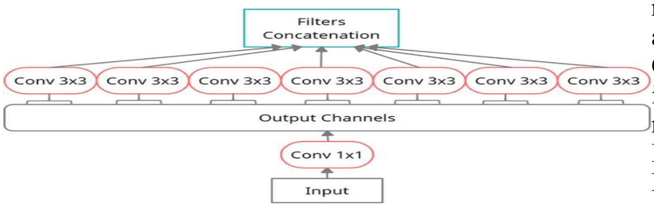

Figure 10: Architecture of Xception

D. *Evaluation*

Any deep learning project must include an evaluation phase since it shows where the model is performing well and where it needs to be improved. Depending on the task and kind of model being evaluated, deep learning frequently employs a range of assessment criteria. This procedure entails assessing how effectively the model predicts the Disease in leaves in order to use deep learning to detect leaf illnesses. Approaches like accuracy are used as assessment metrics in accordance with the particular goals and research design. By giving quantitative insights into the model's ability to generalize to new data, these measures aid in improving the model's predictive accuracy.

Accuracy: In deep learning, accuracy metrics evaluate a model's performance by gauging how well it can generate predictions based on the provided dataset. Accuracy, precision, recall, and F1 score are examples of common accuracy measures. These are calculated using the true positive, false positive, true negative, and false negative predictions. These metrics aid in assessing the model's performance across a range of tasks, including segmentation, object identification, and classification. They also reveal the model's advantages and disadvantages when it comes to managing distinct classes or categories within the dataset.

Table 2: Representing the comparison of the performance Metric

MODEL	ACCURACY FOR 25 EPOCHS (%)	ACCURACY FOR 50 EPOCHS (%)
CNN	89	92
VGG 16	57	76
MobileNetV2	65	91
ResNet 50	32	50
Inception V3	85	80
Densenet121	84	92
Xception	76	92
Densenet& XCeption	88	91
Efficient b3	19	32

VI. RESULTS AND DISCUSSION

TensorFlow is an open-source AI structure created by Google. Keras empowers clients to put less accentuation on minute execution subtleties and more on model design and trial and error It upholds convolutional and intermittent brain organizations. Detectron2 is an open-source profound learning system with cutting edge object recognition and division calculations.

Detectron2 represents a major development in the discipline by introducing a revolutionary algorithm. In addition to this innovation, the landscape is enhanced by a number of pre-existing algorithms, each with unique uses and capabilities. The underlying technique is Convolutional Neural Networks (CNNs), which have an impressive 92% accuracy rate. The repertory of current methods is further enhanced by VGG16, MobileNetV2, InceptionV3, Densenet121, Xception, and the hybrid Densenet-Xception model.

Authorized licensed use limited to: VIT University- Chennai Campus. Downloaded on July 03,2024 at 05:58:21 UTC from IEEE Xplore. Restrictions apply.

Table 3: . The model Comparing between the existing and new model

MODEL	ITERATION	ACCURACY
Detectron 2	2000	90%
	3000	97%

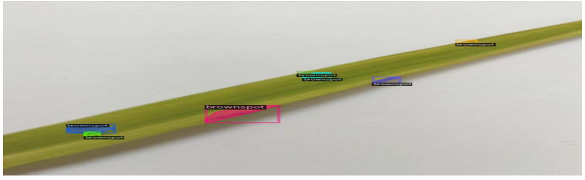

Figure 11: Training phase of the Detectron2

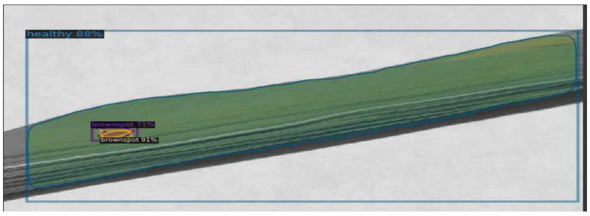

Figure 12: Testing phase of the Detetcron2

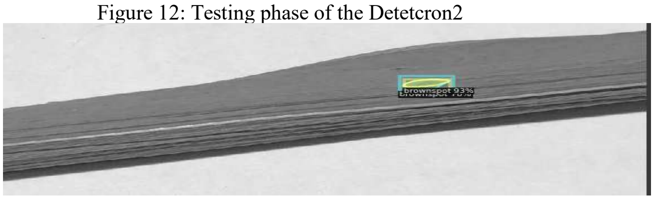

Figure 13: The accuracy is depends upon how clear the image is

These algorithms are not all that effective; their accuracy ranges from 76% to 95%. Detectron2 is a notable performance that stands out due to its outstanding accuracy rate of 97%. Despite being well-established, the EfficientNetB3 model performs worse, with an accuracy of only 32%. This wide range of algorithms highlights the continuous research and development in the field of computer vision.

Although greater accuracy is ideal, training time must be balanced with accuracy. Effective models for deployment include DenseNet121, Xception, and the ensemble model, which all produce acceptable accuracies with manageable training times. Our Detectron2 gives the highest accuracy and runs 3000 iterations in minimum number of time. In contrast, due to their lesser accuracies, models such as ResNet50 and EfficientNetB3 might need more investigation or improvement.

VII. CONCLUSION

In brief, our article has shown how computer vision techniques, in particular deep learning models like CNNs and Detectron 2, may be used to automate the identification of leaf diseases in agricultural settings. We have created a strong framework for precise and scalable illness detection by tackling issues including environmental robustness, scalability, and dataset annotation. Our method combines domain-specific expertise with cutting-edge computer vision techniques to deliver useful answers for researchers, stakeholders, and farmers. Crop health, production optimization, and sustainable agriculture methods could all advance as long as this field's study and development continue. All things considered, this multidisciplinary endeavor is an important step toward improving agricultural technology and tackling the problems caused by leaf diseases in contemporary agriculture. Adding more disease to predict it and also combine all the process and convert to API or website.

VIII. REFERENCES

[1] L. Li, S. Zhang, and B. Wang, "Plant disease detection and classification by deep learning—a review," *IEEE Access,* vol. 9, pp. 56683-56698, 2021.

[2] J. Liu and X. Wang, "Plant diseases and pests detection based on deep learning: a review," *Plant Methods,* vol. 17, pp. 1-18, 2021.

[3] M. Shoaib *et al.*, "An advanced deep learning models-based plant disease detection: A review of recent research," *Frontiers in Plant Science,* vol. 14, p. 1158933, 2023.

[4] G. Shrestha, M. Das, and N. Dey, "Plant disease detection using CNN," in *2020 IEEE applied signal processing conference (ASPCON)*, 2020: IEEE, pp. 109-113.

[5] R. R. Patil and S. Kumar, "Rice-fusion: A multimodality data fusion framework for rice disease diagnosis," *IEEE access,* vol. 10, pp. 5207-5222, 2022.

[6] D. S. Joseph, P. M. Pawar, and K. Chakradeo, "Real-time Plant Disease Dataset Development and Detection of Plant Disease Using Deep Learning," *IEEE Access,* 2024.

[7] Y. Zhao *et al.*, "Plant disease detection using generated leaves based on DoubleGAN," *IEEE/ACM transactions on computational biology and bioinformatics,* vol. 19, no. 3, pp. 1817-1826, 2021.

[8] V. Balafas, E. Karantoumanis, M. Louta, and N. Ploskas, "Machine learning and deep learning for plant disease classification and detection," *IEEE Access,* 2023.

[9] A. Alharbi, M. U. G. Khan, and B. Tayyaba, "Wheat disease classification using continual learning," *IEEE Access,* 2023.

[10] X. Liang, "Few-shot cotton leaf spots disease classification based on metric learning," *Plant Methods,* vol. 17, pp. 1-11, 2021.

[11] R. Dwivedi, T. Dutta, and Y.-C. Hu, "A leaf disease detection mechanism based on L1-norm minimization extreme learning machine," *IEEE geoscience and remote sensing letters,* vol. 19, pp. 1-5, 2021.

[12] E. Haque, M. Paul, A. Rahman, F. Tohidi, and J. Islam, "Rice Leaf Disease Detection and Classification Using Shallow Trained Yolov7 Active Deep Learning Approach," in *2023 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, 2023: IEEE, pp. 516-522.

[13] S. Pawar, S. Shedge, N. Panigrahi, A. Jyoti, P. Thorave, and S. Sayyad, "Leaf disease detection of multiple plants using deep learning," in *2022 international conference on machine learning, big data, cloud and parallel computing (COM-IT-CON)*, 2022, vol. 1: IEEE, pp. 241-245.

[14] S. S. A. Begum and H. Syed, "GSAtt-CMNetV3: Pepper Leaf Disease Classification using Osprey Optimization," *IEEE Access,* 2024.

[15] E. Moupojou *et al.*, "FieldPlant: A dataset of field plant images for plant disease detection and classification with deep learning," *IEEE Access,* vol. 11, pp. 35398-35410, 2023.

Authorized licensed use limited to: VIT University- Chennai Campus. Downloaded on July 03,2024 at 05:58:21 UTC from IEEE Xplore. Restrictions apply.