TOMATO DISEASE FUSION AND CLASSIFICATION USING DEEP LEARNING

Patrick Ansah[1], Sumit Kumar Tetarave[1], Ezhil Kalaimannan[2] and Caroline John[2]

[1]School of Computer Applications, Kalinga Institute of Industrial Technology, India
[2]Department of Computer Science,University of West Florida, FL 32514, USA

ABSTRACT

Tomato plants' susceptibility to diseases imperils agricultural yields. About 30% of the total crop loss is attributable to plants with disease. Detecting such illnesses in the plant is crucial to avoid significant output losses.This study introduces "data fusion" to enhance disease classification by amalgamating distinct disease-specific traits from leaf halves. Data fusion generates synthetic samples, fortifying a TensorFlow Keras deep learning model using a diverse tomato leaf image dataset. Results illuminate the augmented model's efficacy, particularly for diseases marked by overlapping traits. Enhanced disease recognition accuracy and insights into disease interactions transpire. Evaluation metrics (accuracy 0.95, precision 0.58, recall 0.50, F1 score 0.51) spotlight balanced performance. While attaining commendable accuracy, the intricate precision-recall interplay beckons further examination. In conclusion, data fusion emerges as a promising avenue for refining disease classification, effectively addressing challenges rooted in trait overlap. The integration of TensorFlow Keras underscores the potential for enhancing agricultural practices. Sustained endeavours toward enhanced recall remain pivotal, charting a trajectory for future advancements.

KEYWORDS

Disease Fusion, Deep Learning Classification,Tomato Leaf Diseases, TensorFlow Keras, Disease Recognition

1. INTRODUCTION

Tomato (Solanum lycopersicum), one of the most extensively cultivated and economically significant crops worldwide, faces numerous diseases that adversely affect its leaves, ultimately compromising its overall health and productivity. These diseases often manifest as symptoms such as discoloration, lesions, spots, and leaf deformities. Various pathogens, including fungi, bacteria, viruses, and environmental factors, can trigger these ailments. Examples of tomato diseases encompass bacterial spots, early blight, target spot, late blight, leaf mold, yellow leaf curl virus, septoria leaf spot, and spotted spider mite. To ensure sustainable tomato production, comprehending and effectively managing these diseases is crucial, given the pivotal role that tomato leaves play in photosynthesis and the overall vitality of the plant.

Image processing involves the intricate manipulation and analysis of visual data captured in images. Various algorithms and techniques are employed to enhance, transform, or extract valuable insights from digital images. This interdisciplinary field integrates aspects of computer science, mathematics, and engineering to process images for diverse applications, including image enhancement, object recognition, pattern discovery, analysis of plant disease imagery, and medical imaging. In disease analysis, image processing enables efficient analysis and quantification of various elements, such as disease symptoms, patterns, and their correlations with environmental factors. Image processing aims to enhance and interpret visual input using

Bibhu Dash et al: ISTECH, ITEORY, CSITAI -2023
pp. 31-43, 2023. IJCI – 2023 DOI:10.5121/ijci.2023.120703

computational methods, enabling computers to perceive and comprehend images analogous to human vision.

There is always a necessity to use scientific techniques because it is exceedingly challenging and ineffective to diagnose diseases by eye [1]. Farmers in the majority of our native environments are compelled to think creatively about traits or characteristics that have evolved, and these enable them to develop appropriate mechanisms that may help in the eradication of underlying diseases or pest attacks [2]. Formal training is necessary to give people the scientific information to understand this issue wisely and generate cutting-edge solutions to remove it [3].

Although several machine learning models are utilized for the recognition and classification of images, the potential of conventional image recognition algorithms is often only partially realized. This field of Deep Learning study has lots of potential in terms of increased accuracy, especially given the expanding growth of Deep Learning technology, which has benefitted the agricultural industry. Precision agriculture regularly uses image processing. Agriculture-related image processing was covered in-depth in the literature. Image processing is employed in various applications, including identifying, quantifying, and categorizing plant illnesses [4] and phenotyping plant disease signs [5].

Deep learning, being a prolific tool, has been frequently utilized in plant disease detection, with a specific emphasis on its application in diagnosing diseases affecting tomato plants. Mohanty et al. [6] used deep learning to detect illnesses in the leaves of several plants. The paper's approach is tailored to address the detection of prevalent tomato plant diseases, including bacterial leaf spot, septoria leaf spot, and numerous others. It can classify input leaf images into specific disease categories or ascertain their health status. A dataset derived from a subset of PlantVillage [7] was utilized for evaluation, encompassing 15 directories pertaining to three distinct crop varieties. The subgroup includes around 16,024 images of tomato leaf diseases. Similarly, authors in [8] augmented the data, which consists of 18 diseases of tomatoes.

Moreover, Prajwala et al. [9] worked on unstructured images and classified them. They performed experiments using AlexNet [10] and GoogleNet [11], and the best results were obtained during the use of LeNet architecture [12]. The Gabor wavelet transformation technique has been used to extract and identify tomato diseases [13]. Ashqar et al. [14] performed a controlled laboratory environment to produce images of healthy and unhealthy tomato leaves. Llorca et al. [15] collected the images from Google Images to identify the different diseases. Recently, scientists developed several customized convolutional neural network models and transfer learning (TL)-based models to identify tomato leaf disease [16-19].

Reviewing previous research has overwhelmingly demonstrated that the majority of deep learning studies focused on predicting tomato diseases have typically centered on predicting a single disease in each test. We found that the monopoly detection mechanisms mostly predicted different diseases in the presence of another disease on the same tomato leaf. We proposed an augmentation process on the PlantVillage dataset to confuse the existing prediction tools. We propose a four-way method for this work: Data Acquisition, Pre-processing, Data Splitting and Fusing, and Classification. The final stage of the classification utilizes TensorFlow's Keras, which has an intuitive and powerful framework commonly used to build classification models.

The rest of the research paper is structured as follows: Section 2 focuses on the existing research in the relevant field. Section 3 explains the methods for acquiring the required findings alongside the model and proposed technique. Section 4 presents the results and examines the suggested methodology. The conclusion and outlines of the future research directions are presented in Section 5.

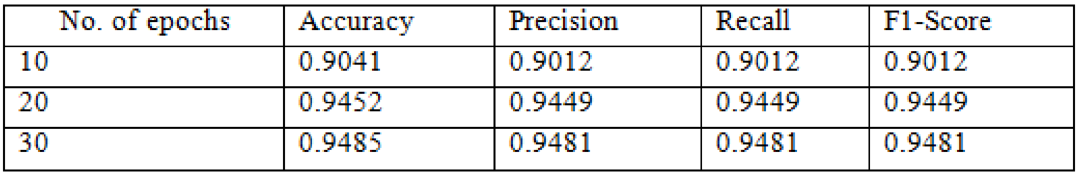
2. Literature Survey

Plant leaf disease detection constitutes a significant research domain, where the amalgamation of image processing and deep learning techniques has resulted in precise classification. This article delves into prevalent methods incorporated within the literature in this domain.

In the paper [8], the authors used a dataset that contained 18 diseases of tomatoes. They used augmentation to increase the dataset from 13,112 to 41,263. They employed five types of Convolutional Neural Networks for classification based on the training they are DenseNet_Xception, Xception, Resnet50, MobileNet, and ShuffleNet. They comprise layers that perform convolutional operations to extract local features from the input data. The accuracy during training was obtained as 97.10%, 93.10%, 86.56%, 80.11%, and 83.68% from DenseNet_Xception, Xception, Resnet50, MobileNet, and ShuffleNet, respectively. The best recognition accuracy of DenseNet_Xception is 97.10%, though the parameters for this method were the most in number. ShuffleNet employed the least number of parameters but recorded an accuracy of 83.68%. ShuffleNet was trained using an augmented data set with optimized parameters and performed well during training and validation. The effect of epoch on the accuracy consistently rises and converges at 1 without any significant drops, typically indicating that the neural network has learned the training data exceptionally well. The model achieved learning success by keeping the Converging about the 1 point through to 1000, 2000, and 3000 epochs. If accuracy increases, a loss is expected to depreciate in the manner of optimum training.

In the paper [9], the authors began with Data Acquisition, Data preprocessing, and classification using Convolutional Neural Networks (CNN) to develop the model that worked on unstructured images and converted them to corresponding classification output labels. The model was trained using 10, 20, and 30 epochs, which are mentioned in Table 1. Validation was performed using a Confusion Matrix. From the outcome, the observation was that, as the number of training processes (epochs) increased, so did the accuracy, prediction, recall, and consequently, the F1-Score.

Table 1. Evaluation of epochs



No. of epochs	Accuracy	Precision	Recall	F1-Score
10	0.9041	0.9012	0.9012	0.9012
20	0.9452	0.9449	0.9449	0.9449
30	0.9485	0.9481	0.9481	0.9481

The methodology detailed in the paper [13] presented how the Gabor wavelet transformation technique was employed to extract distinctive features that helped to identify diseases in tomato leaves. Subsequently, these extracted features are input into a Support Vector Machine (SVM) classifier for training, enabling the determination of the specific disease affecting the tomato leaves. Before the feature extraction, the preprocessing stage involves image resizing, noise reduction, and background elimination tasks. The research used the Gabor transformation to capture textual patterns inherent in the affected leaves and extract relevant features. Disease classification conducted using SVM with varying kernel functions was performed. There was cross-validation on performance evaluation. TheROC curve of SVM using the invmult kernel produced an AUC of 0.90705, while the one using the Laplacian kernel produced an AUC of 0.99679. Experimental results indicate an impressive accuracy rate of 99.5% achieved by the proposed system.

However, it is necessary to note the utilization of Gabor transformation for feature extraction comes with the limitation of computational intensiveness. In the paper [14], the authors employed a dataset comprising 9000 images of infected and healthy tomato leaves. All images were produced within a controlled laboratory environment. This dataset, obtained from the PlantVillage repository, was utilized for the purpose of categorizing five specific diseases: leaf curl, bacterial spot, septoria leafspot, early blight, and leaf mold. A comprehensive color model was utilized for disease spot classification to facilitate data visualization, while a grayscale model was employed to capture the underlying leaf shapes and visual disease patterns. The results revealed that the full-color model achieved superior accuracy compared to the grayscale model. Notably, the captured results were obtained under precisely controlled conditions within the PlantVillage dataset, thereby introducing a potential limitation to the model's applicability in a more diverse setting.

In the paper [15], the authors used a dataset comprising 2,779 images from Google Images. These images encompassed various instances of hornworms, powdery mildew, cutworms, early blight, and whiteflies. Notably, each disease category contained 550 images. This quantity is considered limited for robust training, potentially giving rise to overfitting concerns. The author employed data augmentation techniques to mitigate the overfitting challenge, including vertical flips and random scaling. These techniques introduce diversity and variability into the dataset, contributing to a more generalized and resilient model.

While Convolutional Neural Networks (CNNs) are well suited for deciphering image content, it is worth noting that training a CNN from scratch demands substantial computational resources and a vast dataset. The author opted for a transfer learning approach, utilizing the Google Inception model as a foundation. By leveraging pre-trained weights and features from the Tensorflow Inception V3 model, the author was able to capitalize on its learned representations and optimize the training process, effectively sidestepping the need for extensive data and computational power after the model achieved an accuracy of 88.90%.

The dataset for the paper [16] was obtained from Ehime University in Matsuyama. The scientists developed several simultaneous convolutional neural networks with various topologies to identify tomato leaf disease. To significantly improve the network's performance, they used the activation layers Swish, LeakyReLU-Swish, ReLU-Swish, Elu-Swish, and ClippedReLU-Swish in addition to the Batch Normalization-Instance Normalisation layer. That allowed them to achieve classification accuracy of over 99.0% with training datasets, 97.5% with validation datasets, and 98.0% with testing datasets. Although different performance metrics were observed, none of the suggested networks overfit the validation dataset.

They also employed a variety of methods to visualize network performance. It showed how the networks (Network 1, Network 2, Network 3, Network 4, Network 5) learn from the training dataset and could show infected leaf areas with high confidence scores under actual circumstances. In terms of network stability and illness location visualization, Network 1 performed the best. The shortcomings of Networks 4 and 5 in predicting the Healthy class could be resolved by computing, summarising, and rating the output of several parallel convolutional neural networks [16].

Reference [17] introduces a study that put forward a custom convolutional neural network (CNN) model with a lightweight architecture and employed Transfer Learning models VGG-16 and VGG-19 to classify tomato leaf diseases. This research utilized eleven classes, with one type dedicated to healthy leaves, to replicate various tomato leaf disease scenarios. Furthermore, the study conducted an ablation study to pinpoint the most influential parameters for the proposed model. Moreover, evaluation metrics were used to analyze and compare the performance of the proposed model with the TL-based model. By applying data augmentation techniques, the proposed model achieved the highest accuracy and recall of 95.00% among all the models. Finally, the best-performing model was utilized to construct a Web-based and Android-based end-to-end (E2E) system for tomato cultivators to classify tomato leaf disease.

3. PROPOSED METHODOLOGIES

3.1. SYSTEM OVERVIEW

This research aims to create a model using deep learning that can predict several diseases of tomatoes on the same leaf. Therefore, splitting and fusing distinct diseased leaves would be beneficial. The proposed Split and Fuse model seeks to employ the procedures shown in Figure 1.

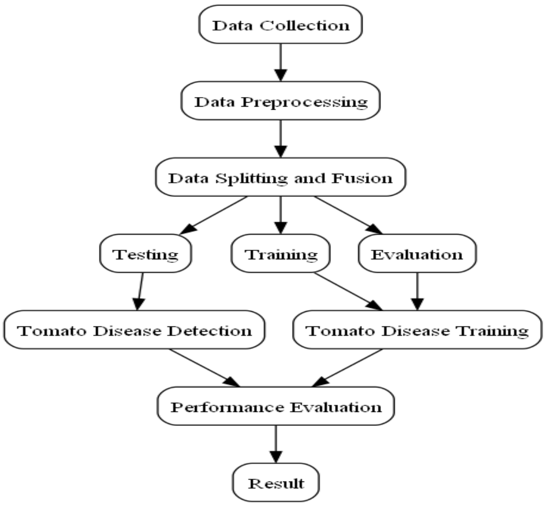


Figure1. The proposed procedure from data collection to evaluation of the results.

3.2. Data Collection

The dataset comprises images of tomato leaf diseases sourced from the Plant Village repository [7]. With a collection of about 16,024 images, it encompasses ten distinct classes. These classes encompass a comprehensive range of leaf diseases that have the potential to impact tomato crops. Some of these diseases are Bacterial spot, Early blight, Late blight, Leaf mold, septoria leaf spot, Target spot, etc.

3.3. Data Pre-processing

The dataset we obtained featured images that exhibited minimal noise, obviating the requirement for noise removal as a preliminary procedure. The images were resized to dimensions of 255 × 255 to accelerate the training phase and render model training computationally viable. The neural representation of resizing and rescaling is shown in Figure 2.

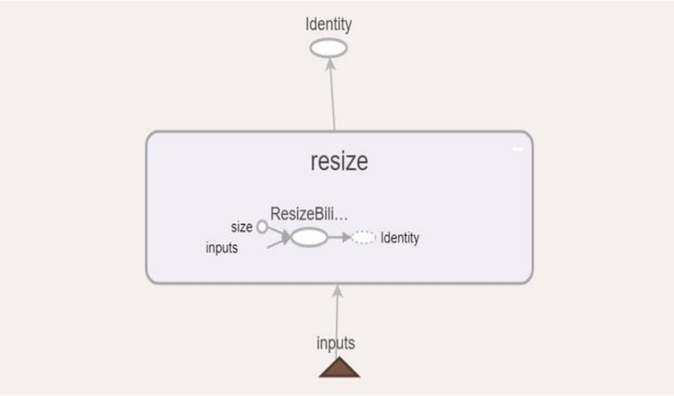


Figure2. The process or pre-processing of the proposed method

Deep learning models can effectively train on small-sized images; therefore, reducing the size of the input images would contribute to the effective training of the model. However, it is important to note that while the images are scaled down slightly, the essential features and patterns necessary for the disease classifications get preserved due to the minimal reduction in size. That enables the training process to be more resource-effective without significantly sacrificing the accuracy of the model's predictions.

3.4. Data Splitting and Fusing

Conventional data augmentation techniques aim at diversifying training data by applying transformations to existing samples. Commonly used augmentation techniques include rotations, flips, translations, changes in brightness and contrast, and more. These transformations help the model become more robust by exposing it to variations of the same data. The method needs to incorporate a seasoned approach. The brain behind this work is to transform the data by splitting each image into two halves from a particular class and fusing each half with another disease image from a different class without duplication to maintain uniqueness. Figure 3 shows the process of splitting and fusing the images.

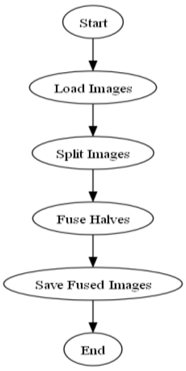


Figure 3. The process of splitting and fusing the images

Through data splitting and fusing, an image from a particular class undergoes division into halves and subsequent fusion with corresponding halves sourced from different ones. The overarching objective of this technique is to replicate intricate interactions across diverse disease classes, thereby generating fresh instances that might not naturally arise. This approach seeks to facilitate the model's training in discerning nuanced disease characteristics, ultimately bolstering its capacity to generalize effectively to real-world scenarios.

With a focus on maintaining distinctiveness, the dataset was constructed by implementing the split-and-fuse method on a selected number of classes. This undertaking resulted in the creating of a total of 20 new sub-classes for the designated activity. An illustrative example is presented in Figure 4, showcasing the fusion of half of an image depicting Tomato Leaf Mold. The Tomato Leaf Mold class is with half of the Healthy class. Additionally, it is noteworthy that a hybrid image featuring half of the septoria leaf spot has been seamlessly integrated with late blight, as exemplified.

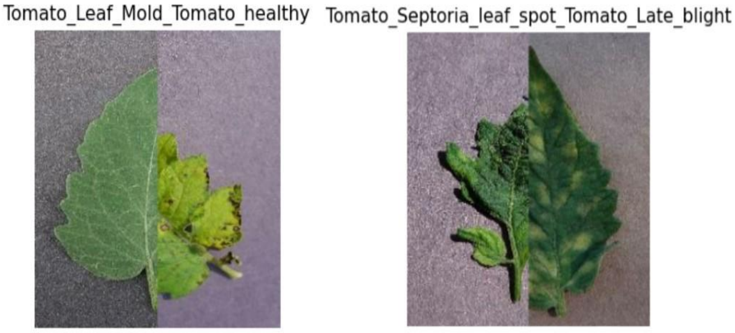

Tomato_Leaf_Mold_Tomato_healthy
Tomato_Septoria_leaf_spot_Tomato_Late_blight

Figure 4. Fused diseases from different classes

3.5. Augmentation

In the context of Convolutional Neural Networks within the domain of Deep Learning, achieving a more robust outcome is mapped to the availability of a substantial volume of data [8]. Even with the possession of a dataset comprising 15,000 instances, the strategic implementation of methodologies aimed at data augmentation emerges as a pivotal avenue for enhancing the quality of this study. Data augmentation represents a canonical strategy employed to amplify the expanse of the dataset corpus, concurrently affording a mitigation mechanism against the potential pitfalls of overfitting [16]. The significance of this approach lies in its ability to integrate modified images into the training dataset, utilizing a variety of techniques like image flipping, rotation, color manipulation, stochastic cropping, and other methods, as cited in reference [17]. Figure 5 illustrates how images are flipped and rotated.

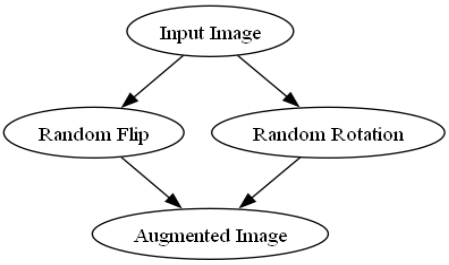


Figure 5. The proposed method to flip and rotate the images

We leveraged the Keras API within TensorFlow. Through this approach, we established a preprocessing pipeline that takes care of the resizing and rescaling input images, ensuring their proper formatting and scaling before their inclusion in the neural network model for training. Keras, an open-source high-level neural network API has been designed for creating effective deep learning models. The pixel values were also rescaled by dividing them by 255, bringing them into the range of [0,1] for normalization.

The neural network arrangement for resizing and rescaling is shown in Figure 6.

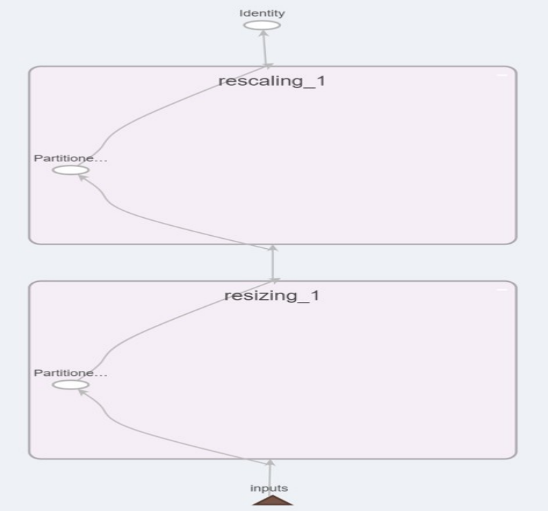


Figure 6. The resize and rescale process

4. CLASSIFICATION MODEL

We adopt the TensorFlow Keras API for the image classification. The arrangement of layers of the Convolutional Neural Networks used for creating the proposed model is described as follows.

4.1. Initial Convolutional Layers

The model's inception involves a convolutional layer that incorporates 32 filters. Each filter is configured with a 3x3 kernel and operates with the rectified linear unit (ReLU) activation function.It's one of artificial neural networks' most commonly used activation functions, especially in deep learning architectures. ReLU is a non-linear activation function that adds non-linearity to the network, enabling it to learn intricate patterns and representations from the input.After each convolution, a max-pooling layer with a 2x2 window size is employed to down-sample the spatial dimensions of the feature maps.

4.2. Intermediate Convolutional Layers

Following the initial convolutional layer, a second convolutional layer is adopted with 64 filters utilizing a 3x3 kernel and ReLU activation. Another max-pooling layer, again with a 2x2 window size, is then applied further to reduce the dimensions of the feature maps. A final convolutional layer utilizes 64 filters with a 3x3 kernel and ReLU activation before a last pooling operation.

4.3. Flattening and Dense Layers

After the convolutional layers, the resultant feature maps get flattened into a 1D vector. A fully connected dense layer comprising 64 units and the ReLU activation function gets utilized to capture higher-level patterns and representations.

4.4. Output Layer

The model concludes with a dense layer with some units equal to 20 classes. The Softmax activation function is employed, which permits the model to yield class probabilities as outputs, i.e., it is to produce the final output or predictions of a neural network. We also incorporated dense layers to act as intermediate layers to perform feature extraction and transformation.The purpose of feature extraction and transformation by the dense layer in the network is to convert raw input data into a more meaningful and representative form in the form of reduced dimension. In our case, it reduces the data input. The dense layer, therefore, enables our neural network to learn and represent relevant information from the input data, facilitating its ability to make accurate predictions or classifications.

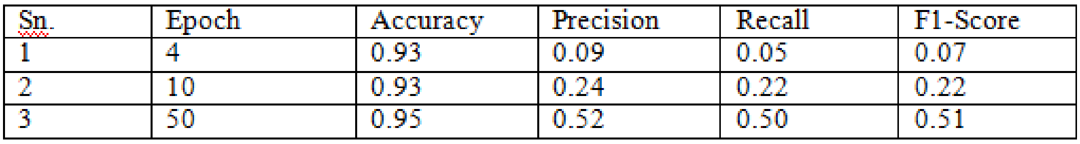
This proposed architecture, constructed through the Sequential API, is tailored to accommodate input data with dimensions of (batch size, image size, image size, and channels). The proposed model leverages convolutional layers to extract hierarchical image features and subsequently applies fully connected layers to enable accurate classification across multiple classes. In addition, to minimize the loss function and enhance the model's performance on the training data, three sets of epochs were executed to facilitate the model's training.

It anticipates that this model will significantly contribute to accurately categorizing images by the specified research objectives.

5. Evaluation

A set of quantitative parameters comprising accuracy, precision, recall, and F1-score are utilized to gauge how effective the proposed model is. The findings have been presented in Table 2, showcasing the highest values of these quantitative metrics achieved up to their corresponding epoch numbers. Both tests with four and ten epochs exhibit the same accuracy of 0.93. These observations suggest that the model's accuracy stabilizes after four epochs, and additional epochs do not substantially improve overall correct predictions on the dataset.

Table 2. Evaluation of parameters



Sn.	Epoch	Accuracy	Precision	Recall	F1-Score
1	4	0.93	0.09	0.05	0.07
2	10	0.93	0.24	0.22	0.22
3	50	0.95	0.52	0.50	0.51

Figure 7 shows the abovementioned phenomenon, where the model's predictions deviated from expected outcomes when tested against datasets. These two parameters significantly contribute to the weakness observed in the Prediction and Recall metrics provided by the confusion matrix. Hence, the model requires enhancement to attain higher accuracy, a pivotal aspect in determining its utility in practical applications and considering Figure 7 when the model created from the epoch of four predicted otherwise due to low precision.

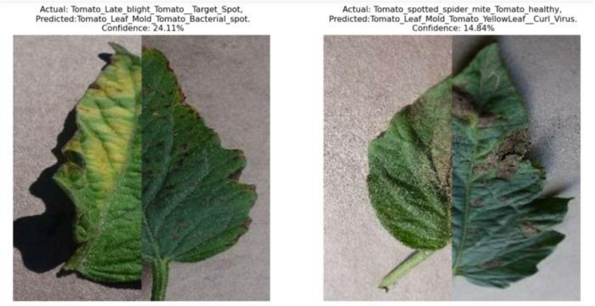

Actual: Tomato_Late_blight_Tomato__Target_Spot,
Predicted:Tomato_Leaf_Mold_Tomato_Bacterial_spot.
Confidence: 24.11%
Actual: Tomato_spotted_spider_mite_Tomato_healthy,
Predicted:Tomato_Leaf_Mold_Tomato_YellowLeaf__Curl_Virus.
Confidence: 14.84%

Figure 7. Incorrect predictions with low confidence percentages

Epoch graphs with accuracy and validation are significantly meant to provide insights into a model's training progress and help identify overfitting. With four epochs, the recall is 0.05. That means that the model correctly identified only 5% of the positive instances in the dataset as positive. With ten epochs, the recall increased to 0.22. That indicates the model now captures a large portion of actual positive instances in its predictions. The influences of these instances are seen in the figure below, where training and validation struggle to exhibit their dominance in the model creation regarding accuracy and loss. Figure 8 depicts the training and validation accuracies of the model, which shows that ten epochs properly work when compared to the epoch of 4.

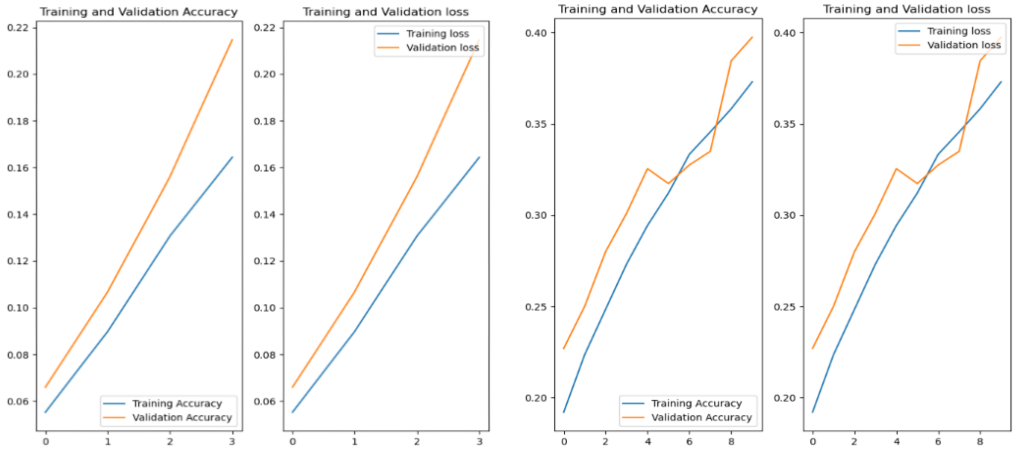


Figure 8. Training and validation accuracies withepoch of 4 and epoch of 10

Considering four epochs in Figure 8, the F1-score is 0.07. The F1-Score is calculated as the harmonic mean of precision and recall. It is valuable for evaluating a model's ability to correctly classify positive instances while minimizing false positives and false negatives. A low F1 score indicates how the model strives to balance false positives and negatives. On the other hand, the F1-score has increased to 0.23 with ten epochs. That suggests that the model's precision and recall have improved in a way that positively affects the overall balance captured by the F1-score. That balance was realized during its test using a dataset, as shown in Figure 9.

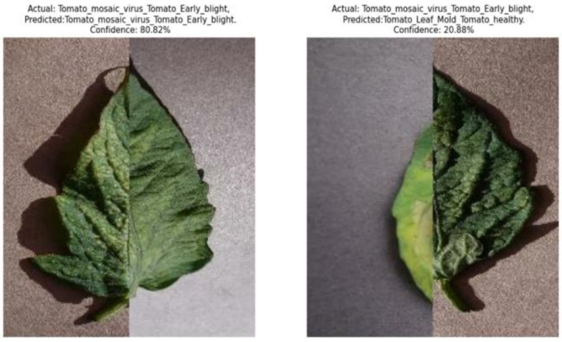

Actual: Tomato_mosaic_virus_Tomato_Early_blight,
Predicted:Tomato_mosaic_virus_Tomato_Early_blight.
Confidence: 80.82%
Actual: Tomato_mosaic_virus_Tomato_Early_blight,
Predicted:Tomato_Leaf_Mold_Tomato_healthy.
Confidence: 20.88%

Figure 9. Correct Prediction vs Wrong Prediction

Figure 9 shows an accurate prediction after the model is provided with a dataset at a confidence of 80.82%. The same model made incorrect predictions with a confidence level of 20.88%. When supplied with the Mosaic virus and Early blight, the model erroneously predicted health, suggesting the need for improvement. Moreover, with an epoch count of 50, the model's overall effectiveness improved compared to epochs of 4 and 10. The model trained for 50 epochs demonstrated a better capability to capture a more significant portion of actual positive instances in its predictions.

While accuracy remains consistent between the 4-epoch and 10-epoch models, the 50-epoch model achieves an improved balance between precision and recall. That implies that extended training enabled the model to become more accurate, particularly in correctly classifying positive instances, resulting in a more balanced and higher-performing model overall. Figure 14 (left side) illustrates training visualization at 50 epochs, demonstrating progressive enhancement compared to 4 and 10 epochs. On the right side of Figure 10, the model achieves highly accurate predictions with confidence levels as high as 99.82% when provided with a test dataset. Nevertheless, the model exhibited enormous weaknesses in other predictions, e.g., in Figure 11.

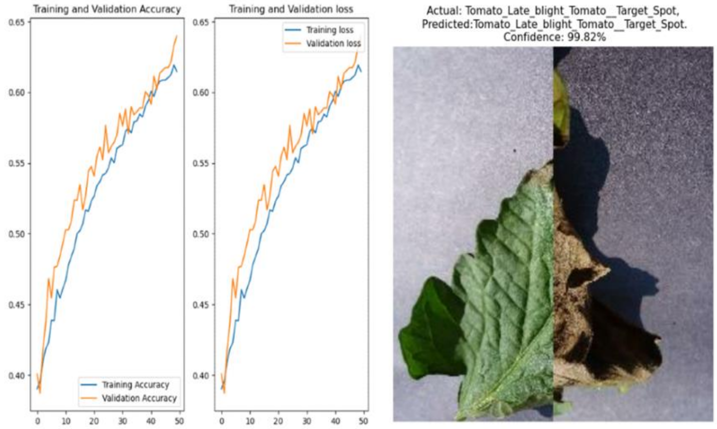


Figure 10. An accurate Prediction with high confidence

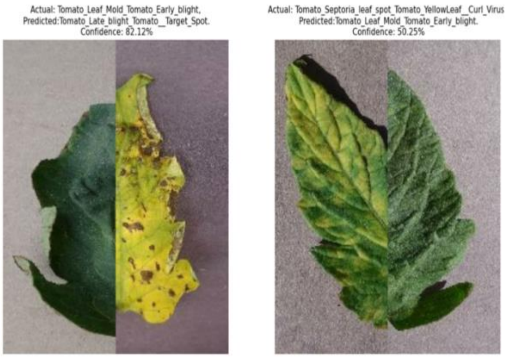
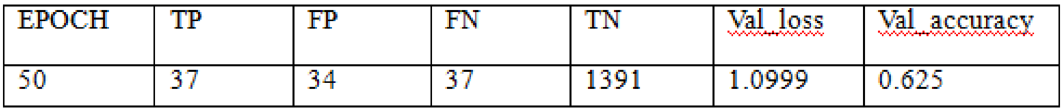

Actual: Tomato_Leaf_Mold_Tomato_Early_blight,
Predicted:Tomato_Late_blight_Tomato__Target_Spot.
Confidence: 82.12%
Actual: Tomato_Septoria_leaf_spot_Tomato_YellowLeaf__Curl_Virus
Predicted:Tomato_Leaf_Mold_Tomato_Early_blight.
Confidence: 50.25%

Figure 11.An example of incorrect predictions

Table 3. Details of 50 epochs



EPOCH	TP	FP	FN	TN	Val_loss	Val_accuracy
50	37	34	37	1391	1.0999	0.625

Table 3 shows the recorded validation loss value of 1.0999 and a validation accuracy of 0.65, providing insights into the model's capacity to generalize its acquired knowledge to previously unseen data. The loss value indicates the model's proficiency in minimizing errors during validation, while the validation accuracy affirms its ability to sustain high performance beyond the training dataset. A relatively elevated count of True Negatives (1391) signifies the model's competence in accurately identifying the absence of the target class. Consequently, this proficiency also increases the likelihood of generating inaccurate predictions. These metrics underscore the positive influence of the model's architecture and training process on its performance, thus highlighting its suitability for addressing the research objectives, albeit not at an optimal level.

6. CONCLUSION

Contemporary research employs deep learning in image processing and pattern recognition, facilitated by the intricacy of its designed neural architectures. These approaches have been proven effective in identifying diseases that affect plant leaves. We revealed by our observations that splitting and subsequently merging different leaves afflicted by distinct diseases results in a complex structure suitable for training through convolutional networks. Augmenting the training dataset also enhances the quantity of data available for training, leading to improved model performance. Increasing the number of epochs impacts model accuracy. Consequently, an efficient model capable of swiftly detecting multiple diseases on a plant leaf gets modeled using this concept, demonstrating a confident prediction accuracy of 99.82% for two diseases.

In future research, with our work's objective in mind, we propose creating a more robust model for tomato disease prediction. This model should be capable of predicting the number of diseases present on an image while assessing their severity in terms of how much affected the leaves are using the data fusion method. Additionally, constructing a more robust convolutional neural network that can comprehend complex structures such as Disease fusion would be valuable for future research, which is pertinent as most studies often involve only a single disease per leaf when modeling with CNNs. The employment of a separable convolutional network addition of dropouts with batch transformation could be included in the model to enhance the training phase with more profound learning abilities.

REFERENCES

[1] Shijie, Jia, Jia Peiyi, and Hu Siping. "Automatic detection of tomato diseases and pests based on leaf images." 2017 Chinese automation congress (CAC). IEEE, 2017.

[2] Chen, Hui-Ling, et al. "Support vector machine based diagnostic system for breast cancer using swarm intelligence." Journal of medical systems 36 (2012): 2505-2519.

[3] Blancard, Dominique. Tomato diseases: identification, biology and control: a colour handbook.CRC Press, 2012.

[4] Arnal Barbedo, Jayme Garcia. "Digital image processing techniques for detecting, quantifying and classifying plant diseases." SpringerPlus 2.1 (2013): 1-12.

[5] A. M. Mutka and R. S. Bart, "Image-Based Phenotyping of Plant Disease Symptoms," Frontiers in Plant Science, vol. 5, pp. 1-8, 2015.

[6] S. P. Mohanty, D. Hughes and M. Salathe, "Using Deep Learning for Image-Based Plant Disease Detection," eprint arXiv:1604.03169, 2016.

[7] Kaggle.com,https://www.kaggle.com/datasets/arjuntejaswi/plant-village?select=PlantVillage, accessed on September 2023.

[8] Hong, Huiqun, Jinfa Lin, and Fenghua Huang. "Tomato disease detection and classification by deep learning." 2020 International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE). IEEE, 2020.

[9] Tm, Prajwala, et al. "Tomato leaf disease detection using convolutional neural networks." 2018 eleventh international conference on contemporary computing (IC3). IEEE, 2018.

[10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. In: Advances in neural information processing systems.2012, pp. 1097–1105.

[11] Szegedy, Christian, et al. "Going deeper with convolutions." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.

[12] Yann LeCun et al. “Backpropagation applied to handwritten zip code recognition”. In: Neural computation 1.4 (1989),

[13] Usama Mokhtar et al. “Tomato leaves diseases detection approach based on support vector machines”. In: Computer Engineering Conference (ICENCO), 2015 11th International. IEEE. 2015, pp. 246–250.

[14] Ashqar, Belal AM, and Samy S. Abu-Naser. "Image-Based Tomato Leaves Diseases Detection Using Deep Learning." IJARWpublication(2018).

[15] Llorca, Charmaine, May Elsbeth Yares, and Christian Maderazo. "Image-based pest and disease recognition of tomato plants using a convolutional neural network." Proceedings of international conference technological challenges for better world. 2018.

[16] Islam MP, Hatou K, Aihara T, Seno S, Kirino S, Okamoto S. “Performance prediction of tomato leaf disease by a series of parallel convolutional neural networks.” Smart Agricultural Technology, 2022; 100054: 25 pp.

[17] Showmick Guha Paula, Al Amin Biswas, Arpa Saha, Md. Sabab Zulfiker, Nadia Afrin Ritu. “A real-time application-based convolutional neural network approach for tomato leaf disease classification.” Array19(2023)100313, 2023, 14 pp

[18] Geetharamani G. a, Arun Pandian J.“Identification of plant leaf diseases using nine-layer deep convolutional neural network.” Computers and Electrical Engineering 76 (2019) 323-338. https://doi.org/10.1016/j.compeleceng.2019.04.011

[19] Zhang Y-D, Dong Z, Chen X, Jia W, Du S, Muhammad K, et al. “Image based fruit category classification by 13-layer deep convolutional neural network and data augmentation.” Multimed Tools Appl 2019;78(3):3613-32.