Journal of Computing & Biomedical Informatics
ISSN: 2710 - 1606
Volume 04 Issue 02
2023

Research Article
https://doi.org/10.56979/402/2023

Analysis of Deep Learning Algorithms for Detection and Classification of Tomato Leaf Diseases

Muhammad Mubashar Siddique[1], Muhammad Junaid Nazar[2], and Khola Farooq[3]

[1]Department of Creative Technologies, Air University, Islamabad, 44000, Pakistan
[2]Department of Computer Science, Air University, Islamabad, 44000, Pakistan
[3]Department of Computer Science, Lahore Garrison University, Lahore, 54000, Pakistan
*Corresponding Author: Muhammad Mubashar Siddique, Email: hmmubashar07@gmail.com

Received: January 10, 2023 **Accepted:** February 12, 2023 **Published:** March 29, 2023.

Abstract: The utilization of deep learning models has gained significant popularity for identifying and categorizing Tomato Leaf Disease, owing to the intricacy of the task and the proficiency required for precise identification. However, the current models employed for this function are typically judged based on their ability to distinguish between healthy and diseased leaves. This model evaluation approach is limiting. The present models are trained and validated on a small dataset that covers only 2000- 5000 images, referred to as Plant Village. This limited dataset size is deemed unreliable, leading to inefficiency and a lack of robustness in the models. Consequently, it makes agriculture-based applications challenging, particularly when spotting new visual effects of Tomato Leaf Disease in smaller datasets. Additionally, the lack of multiclass classification/detection and the absence of damaged regions' localization in the existing studies constitute a grave concern. Such a deficiency in localization can result in incorrect diagnoses and treatments, leading to crop losses, therefore highlighting the necessity of comprehensive evaluation and validation of these models. This study's primary focus is on addressing the limitations of a small dataset, and it proposes using a more extensive multi-source dataset of around 40000–45000 images. This study performs data preprocessing to balance the dataset's classes using data augmentation techniques. Afterward, the final augmented dataset consisted of 60000 images. In concomitance, it notably conducted to an augmented discriminative aptitude, substantiated by a resultant classification efficacy of 99.97%. Concurrently, this methodical intervention correlated with a salutary diminution in the temporal overhead implicated in the training of the expansive cohort encompassing 40,000 distinct tomato leaf images.

Keywords: Deep Learning; Machine Learning; Data Augmentation; Tomato Leaf Disease;Feature Extraction; Agricultre.

1. Introduction

Plant leaf diseases [1] refers to the condition where plant leaves are damaged or impaired due to several factors, including environmental stressors [2], nutrient deficiency [3], and pathogens. Fungal [4], bacterial [5], and viral infections [6] cause many diseases in ornamental plants and tree crops. Detecting plant diseases has become increasingly important in maintaining agricultural productivity and managing crop damage, and loss [7]. Identifying plant leaf diseases and their stages of development remains a primary concern in agriculture. Implementing digitalized processing techniques can benefit the country as it helps reduce losses [8]. Detecting and diagnosing tomato leaf disease is crucial for cultivating and preserving tomato crops, as it falls under the category of plant leaf diseases. To prevent the spread of such outbreaks, it is necessary to implement effective management and strategies [9].

Over the past few years, computer vision techniques, machine learning, and deep learning have been used for precise diagnosis and early detection of various stages of tomato leaf disease. Additionally, researchers have utilized computer vision methods to detect and recognize the presence of tomato leaf disease [10]. Traditionally, the identification of tomato leaf diseases has depended on manual inspections by farmers or agricultural experts. Nonetheless, the progress in computer vision and machine learning techniques has led to the creation of automated disease detection systems [11]. The use of feature extraction techniques in machine learning has proven advantageous for farmers and has led to improved crop production in the agriculture industry [12]. Recently, several authors have developed a machine learning approach to detect tomato leaf disease, wherein dataset preprocessing, feature extraction, and classification methods are essential steps that contribute significantly to the accurate identification and prediction of tomato leaf disease [13]. Tomato leaf disease outbreaks can lead to the destruction of numerous tomato crops, thereby impacting their productivity. Among the diseases affecting most plants and leaves is the Fusarium oxymoron (FO) disease, identified for detection in a study [14]. Agriculture is the primary source of livelihood and crop production. To minimize the significant losses and damage caused by such diseases, an efficient machine learning-based model was proposed in a study[15] for the early detection of tomato leaf disease. Several types of tomato leaf diseases can affect your plants, including fungal diseases like early blight, late blight, and Septoria leaf spot, as well as viral diseases like tomato mosaic virus and soil-borne diseases like fusarium wilt and verticillium wilt.

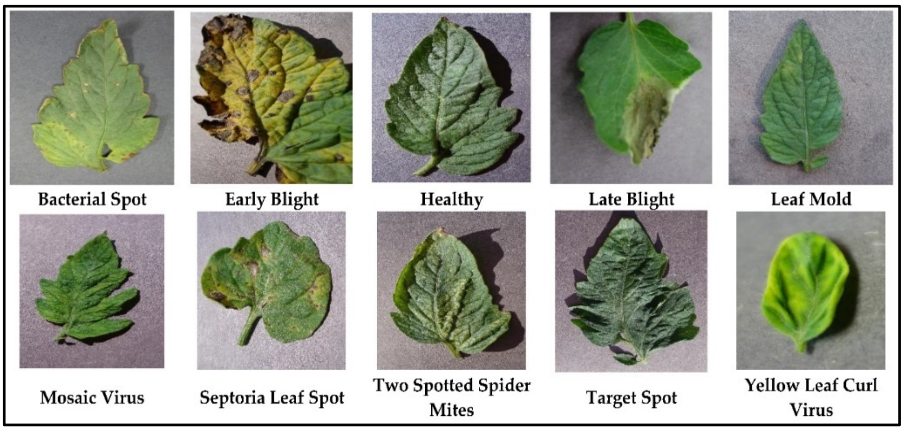

Figure 1. Types of Tomato Leaf Disease

Figure 1 shows the graphical representation of the types of tomato leaf disease. Most of the existing work suffers from model efficiency and robustness due to a single dataset with a smaller number of images ranging from 2000-5000; this is assumed to be too insufficient to depend upon; as a result, the models' applicability on a wide range of datasets needed to be recorded. Additionally, most existing studies did not perform data segmentation and only worked with single performance metrics, i.e., accuracy. This study aims to surpass the current work limitations by amalgamating numerous datasets sources, boosting the number of images uniformly, and preprocessing them with data augmentation to address the issue of imbalanced datasets. Additionally, the study will conduct feature extraction using CNN models. The researchers intend to investigate other performance evaluation metrics, such as specificity and sensitivity, to analyze the model's performance on test data. Below are the contributions and recommendations formulated to attain these goals:

The proposed work aims to tackle the limitations posed by a small dataset by augmenting and diversifying the dataset. The study proposes incorporating multiple sources of images to create a more extensive and diverse dataset consisting of approximately $40000 - 45000$ images.

The study utilized data augmentation techniques for data preprocessing to balance the dataset's classes and address the issue of imbalanced datasets. This approach significantly increases the dataset's size from approximately $2000 - 5000$ images to around 60000, enhancing its diversity. The study expects this larger and more diverse dataset to improve the deep learning models' ability to learn effectively and perform better.

To leverage the power of deep learning models on diverse tomato leaf image datasets, the study implemented three different models. The first model utilized was Inceptionv3, the second model was MobileNet, and the third model was ResNet50.

This study utilized deep learning models to accomplish two tasks concurrently. The first task entailed identifying diseases in tomato leaves by distinguishing between healthy and affected samples. The second task involved the multi-class classification of ten distinct categories. By capitalizing on the strengths of deep learning, we successfully tackled both tasks in parallel, showcasing the adaptability and efficacy of this methodology.

The researchers used well-established evaluation metrics, including accuracy, specificity, sensitivity, recall, precision, and F1-score, to assess the model's performance. The proposed model exhibited a remarkable accuracy rate of 99%, surpassing current industry benchmarks and effectively identifying and classifying Tomato Leaf Disease.

This paper is effectively organized into several sections for readability and accessibility. The ”Literature Review” covers previous studies in detail. The ”Dataset” section briefly describes the dataset selection process. The ”Proposed Approach” section outlines the study's proposed approach. The ”Experimental Analysis and Results” section details the experimental analysis and findings. Finally, the paper concludes with a ”Conclusion” section, examining the research's potential limitations and avenues for future research.

2. Related work

Much work has been done for tomato leaf disease detection, prediction, and diagnosis of the various stages using machine learning, deep learning, and computer vision techniques. This section covers all the related work done by Machine learning, deep learning, and ensemble models.

2.1. Tomato Leaf Disease Prediction Using Machine LearningModels

An effective method for classifying tomato leaf disease based on multiple features was suggested by [15]. The approach involved utilizing a standard Plant Village Dataset and digital image processing (DIP) techniques to select features such as color, shape, and texture from the images. Support Vector Machines and K-mean clustering were employed to execute the proposed model, which achieved an 86.6% accuracy through cross-validation. A model based on machine learning was suggested in [13] for detecting tomato leaf disease. The model utilized the plant village dataset and applied a histogram feature extraction technique to normalize and flatten the image matrix. Eight machine learning algorithms, including decision tree, random forest, support vector machine, logistic regression, and others, were used in the proposed model. Among them, random forests achieved the highest accuracy of 98% compared to the other algorithms. A machine learning-based approach was developed in [16] for the various stages of tomato leaf disease. The proposed method utilized a standard plant village dataset, which included images of leaves with high, medium, and low severity. The approach involved feature extraction techniques such as shape, color, and texture. Five machine learning algorithms were implemented: Support Vector Machine (SVM), K-nearest neighbors (KNN), Naïve Bayes, Decision tree, and LDA. Among them, the Support Vector Machine (SVM) algorithm achieved an accuracy of 78.44%.

2.2. Tomato Leaf Disease Prediction Using Deep Learning Models

Using deep learning models. Study in [17] utilized a standardplant village dataset and applied a CNN (convolutional neural networks) automatic features extraction technique to develop a deep learning-based model. The model achieved an accuracy of 99.6% through five-fold cross-validation by implementing deep learning models, including CNN1 and CNN2 (Inceptionv3, Mobile Net). Research work in [18] proposed an automatic detection deep learning-based system for tomato leaf disease using a dataset of images. They performed data augmentation as preprocessing and implemented deep learning models such as T-leaf Net, Alex Net, MobileNetV2, and VGG16. T-leaf Net achieved the highest accuracy in a shorter time. A study in [11] developed a contemporary method for detecting various plant diseases using the 2598 plant village dataset. The model implemented a deep learning model including CNN, Inceptionv3, and Transfer learning and obtained an accuracy of 96.37%. The research in [19] proposed a novel approach for detecting tomato leaf disease using pre-trained deep learning models such as CNN, Inceptionv3, and Inception ResNet V2. They used a standard plant village dataset of 5225 images and achieved the highest results of 99.22% accuracy. Research in [20] proposed a pipeline for the automated detection of tomato leaf disease using the standard plant village dataset. They performed feature selection using statistical measures and CNN models and implemented Knearest neighbor (KNN) and support vector machine (SVM). The model achieved the highest accuracy of 99.92%. The research [21] proposed a deep CNN model for detecting tomato leaf disease using a standard plant village dataset of 418160 images. The model automatically learned the features from the dataset and achieved the highest accuracy of 98.40%. The work in [22] proposed an intelligent-based approach to identify and detect tomato leaf disease using a hybrid model of optimized k-mean clustering (OKMC), CNN, VGG16, and Resnet. The proposed model used a standard plant village dataset and performed automatic feature extraction from the images.

2.3. Tomato Leaf Disease Prediction Using Ensemble Model

In [20] study, a Convolutional Neural Network (CNN) with Transfer Learning (TL) feature selection algorithm was employed to detect and classify plant diseases using the Plant Village dataset. This approach utilizes a combination of statistical measures and machine learning algorithms with TL, resulting in a high accuracy rate of 99.92%and99.90%. However, there remains a need for validating the approach on larger and more diverse datasets to ensure the reliability and robustness of the models. A voting ensemble model was suggested by Sayed et al. [21] as a unique method for classifying tomato leaf disease. bacterial and target areas, as well as early bling, were detected using RGB imaging. preprocessed images, adding shape, color, and texture. Implement the random forest, decision tree, SVM (support vector machine), KNN (k-nearest neighbors), naive bayes, and discriminative analysis algorithms among the six machine learning algorithms. achieved accuracy ranging from 44.42% to 91.53%, with ensemble voting approaches achieving a 95.58% accuracy. Overall, this work contributes meaningfully to advancing plant disease detection and classification, providing a solid foundation for future research.

Several other methods, including Naive Byes, SVM, DT, Random Forest, KNN, AdaBoost, Fuzzy classifiers, and NN rules-based classification, were utilized by certain researchers to identify and categorize diseases. Every strategy has its drawbacks. Others concentrated on only two classes—healthy or unhealthy—and those are utilized public datasets. They are not categorized as a particular illness kind, thus combining diverse approaches is a smart idea to get beyond their limitations. To get reliable findings as quickly as possible, we combined three strategies to create the suggested strategy.

To capture the picture, this technique was designed. In order to eliminate over-feting, it has been discovered that employing technology-based data preparation and data augmentation produces positive outcomes and improves performance. Just five disease classifications may be identified by this investigation, along with healthy tomato leaves. The suggested network model must be trained using an increasing amount of fresh data in order to identify more illness classifications.

3. Materials and Methods

predictive deep-learning model for Tomato Leaf Disease. This would be achieved using a multi-source, varied dataset. The proposed approach comprises several stages aimed at generating the model. These phases involve creating a dataset using multiple sources, specifically, Plant Village, Tomato Leaf Multi-Source, and Tomato Leaf Dataset (Kaggle), implementing data preprocessing techniques, addressing data balance issues, utilizing feature extraction, and developing model predictions using deep learning algorithms. The proposed approach employs a diverse multi-source dataset illustrated in **Figure 2**.

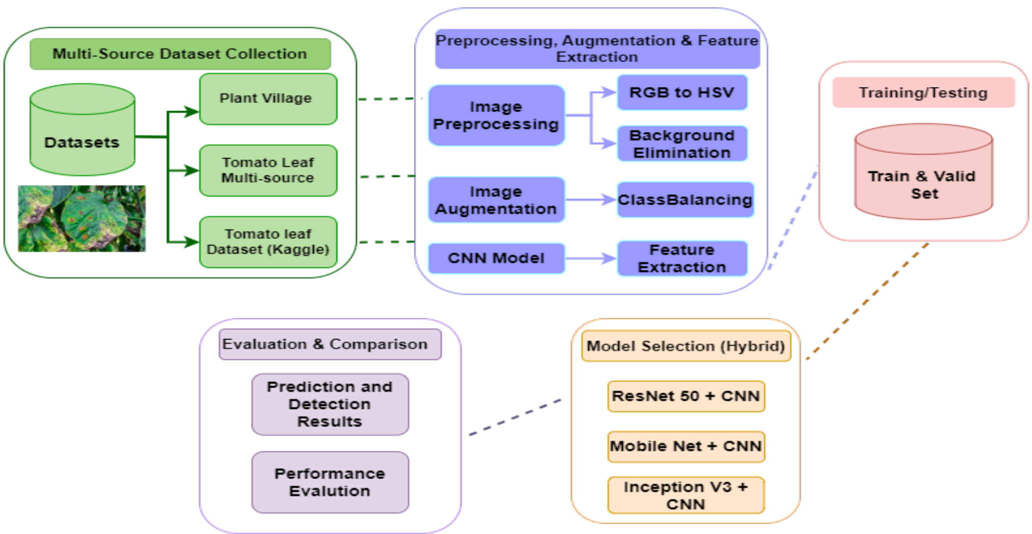

Figure 2. Proposed Methodology

The initial phase of the proposed approach involves collecting data from multiple sources to address the existing issue of a small dataset. The data comprises approximately 40000-45000 images spread across ten classes, although each class has varying images. To balance the dataset's classes, the study employed data augmentation techniques that augmented the dataset to approximately 60000 images. Once the dataset was balanced with data augmentation, the study implemented three deep learning models to analyze the data. These models were Inceptionv3, Mobile Net, and ResNet50. The aim was to use these highly regarded models to provide insights and understand the Tomato Leaf Disease classification process. The models were developed to work with the varying aspects of the dataset, so each performance was measured independently. Finally, the study assessed the models' performance using various evaluation metrics to validate the models' precision, recall, sensitivity, specificity, F1 score, and overall accuracy. This proposed approach's collection and analysis stages aimed to overcome the limitations of a small dataset and develop more accurate and effective methods of identifying and categorizing Tomato Leaf Disease. The proposed deep learning-based framework 1 implements a deep learning model for image classification using the Inceptionv3 architecture and an Image Data Generator for data augmentation. It defines the input size of an image and batch size and uses several functions, such as check valid image, feature extractor, classifier, and prediction layer, to build the model. After defining the model, it unfreezes some layers for fine-tuning and compiles the model with the RMSprop optimizer and Categorical Cross entropy loss function. It then creates callbacks to monitor the validation loss and accuracy, save the best model weights, and reduce the learning rate on the plateau. Finally, it trains the model using the model. fit () function call with generators, epochs, steps per epoch, validation steps, batch size, and callbacks as necessary parameters.

Algorithm 1: Pseudo code of Proposed Deep Learning Framework For Tomato Leaf Disease Detection and Prediction

1**: Input**: shape for an image 224x224 and batch size 64.
2**: Output**:Multiclass detection
3: **Evaluation:** Evaluation Measure: Accuracy, Precision, Recall, F1-Score,Sensitivity and specificity
4: Define functions:
5: check valid image(file path) - to check if an image file is valid
6: feature extractor(input tensor) - to extract image features using Inceptionv3 architecture.
7: classifier(x) - to apply dense layers with regularization techniques to the extracted features.
8: prediction layer(x) - to add a dense layer with softmax activation function for predictions.
9**: Augmentation:** Implement an ImageDataGenerator for data augmentation.
10: Create train and validation generators using the train and validation directories as inputs to ImageDataGenerator.
11: **Model Building:** Build a deep learning model with the following steps:
12: Define inputs and call feature extractor with these inputs
13: Call classifier with feature extractor output as input.
14: Call prediction layer with classifier output as input.
15: Create a Model using the inputs and prediction layer output as outputs.
16: Unfreeze some of the later layers for fine-tuning.
17: Compile the model with RMSprop optimizer and CategoricalCrossentropy loss function.
18: Define callbacks to save the best model weights, monitor validation loss and accuracy, and reduce the learning rate on the plateau.
19: Call model.f it() to train the model with necessary parameters, including generators, epochs, steps per epoch, validation steps, batch size, and callbacks.

3.1. Data Description

This section provides a detailed overview of dataset creation from multiple sources. Further information and descriptions about this topic can be found in Table I. The three datasets have been merged to create a single dataset with ten classes, consisting of approximately 40, 000 − 45, 000 images.

Table 1. Multi-Source Data with Descriptions

S. No	Data Name	No of Images	classes
1.	Tomato Leaf Multiple Source	25,000 above images	11
2.	Mendeley Data (Tomato Leaves)	14,000 above images	10
3.	Tomato leaf dataset	10,000 images	10

3.2. Data Preprocessing:

Following the creation of the dataset, it was necessary to perform data preprocessing due to the unbalanced image sizes across the different classes, which could affect the model's performance. Both the photos we gathered and those found in the Plant village files are RGB and may be of any size. Scaling the image to 256x256 while maintaining the same aspect ratio was necessary since many neural network methods need a rectangular input picture. leveling off the intensity and decreasing the low-frequency background noise. Image editing entails cropping of all the images by hand, producing a rectangle surrounding each leaf. It is certain that the image comprises every pertinent detail for the collection of features.

1. Convert RGB into HSV
2. Background Elimination

Then, data augmentation technique was utilized to balance the number of images in each class and increase the overall dataset size to 60, 000. The augmented dataset was then split into two sets, namely training and testing sets. A visualization of the counts of the augmented images per class is shown in **Figure 3**. The x-axis shows the classes, and the y-axis shows the image count.

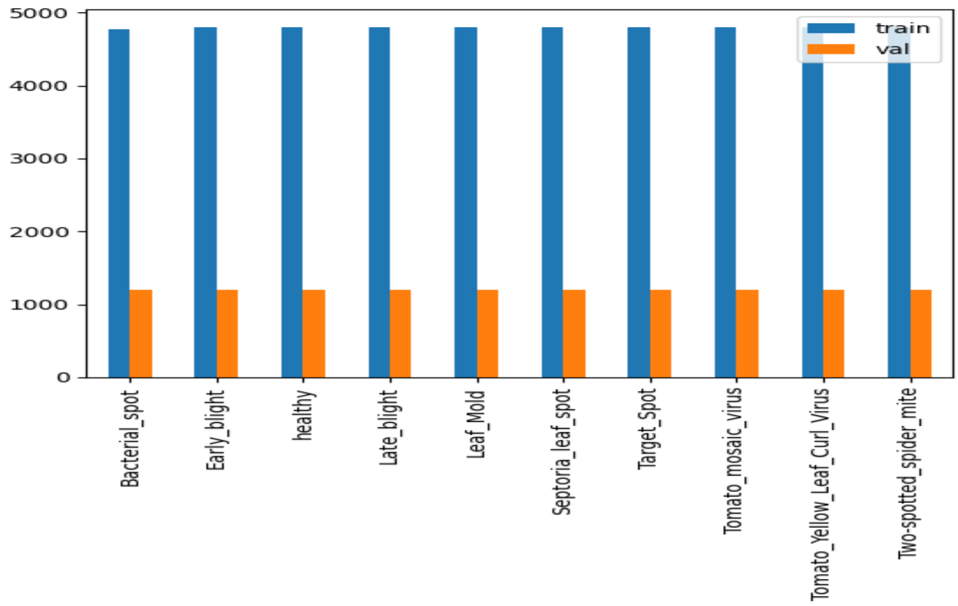

Figure 3. Augmented Training and Validation Images of Each

4. Experimental Results and Discussion

This section thoroughly analyzes the proposed deeplearning framework for Tomato Leaf Disease. The experimental dataset is split into two sets, with 80% used for model training and 20% used for model validation. Deep learning models are employed to train the system, and its performance is evaluated using various evaluation metrics such as accuracy, precision, recall, F1-score, and confusion matrix. Detailed discussions on the results obtained from the experiments are also presented in this section.

4.1. Experimental Settings

To conduct the research experiment, specific technologies and tools were used. Python version 3.9 was selected as the programming language because of its extensive range of libraries and tools for data processing, analysis, and visualization. The Kaggle Notebook development framework was utilized for the experiment, with Python 3.9 as the selected language. Windows was chosen as the operating system due to its reliability and efficiency in running Python applications. The experiment was conducted on a Hair

11y c laptop with a powerful processor and ample memory. An Nvidia 1060 graphics processing unit (GPU) was also integrated into th experimental setup, facilitating fast and efficient parallel processing and accelerating the training and evaluation of deep learning models. **Table 2.** summarizes the various tools and technologies utilized in the experimental setup, aiding in comprehending the setup and the resources involved in the deep learning experiment.

Table 2. Experimental Settings

Parameters	Values
Framework	Kaggle
Operating System	Windows
Hardware Platforms	Dell Spectre
GPU	Nvidia 1060
Programming Language	Python 3.9

4.2. Evaluation Matrix

The present investigation utilizes an array of evaluation metrics to meticulously assess the performance of a model, comprising accuracy, precision, recall, sensitivity, specificity, F1-score, and confusion matrix, to achieve unprecedented levels of analytical precision and accuracy. Accuracy, the most comprehensible metric, gauges the proportion of accurately classified instances concerning the total number of instances. This fundamental metric can be calculated using the equation provided:

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{2TP+FP+FN+TN}a} \quad (1)$$

As a performance metric, precision reveals the ratio of true positive predictions to all positive predictions, thereby providing insight into the model's ability to avoid false positives. This preeminent metric can be ascertained utilizing the equation provided:

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \quad (2)$$

Recall, commonly referred to as sensitivity, is a preeminent performance metric that measures the model's ability to detect all actual positive instances in the dataset. This invaluable metric can be ascertained using the equation provided:

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \quad (3)$$

Specificity is a pivotal performance metric that measures the model's ability to detect all negative instances in the dataset. It reveals the ratio of true negative predictions to all negative instances in the dataset and is calculated using the equation provided:

$$\text{Specificity} = \frac{\text{TN}}{\text{TN+FP}} \quad (4)$$

F1-score, as an excellence metric, amalgamates into a harmonious whole the aforementioned precision and recall metrics, resulting in an informative and comprehensive measure of the overall model performance. This pivotal metric is fundamental in evaluating the system's performance and can be ascertained using the equation presented:

$$\text{F1 Score} = \frac{2*\text{precession}*\text{Recall}}{\text{Precession + Recall}} \quad (5)$$

4.3. Multi-Class Confusion Matrix

A multi-class confusion matrix serves as an evaluation table for machine learning models capable of classifying examples into more than two distinctive categories. It is structured such that each column represents the predicted class and each row the true class and is imbued with an array of metrics that provide insights into the model's performance. Correct classifications for each class are found within the diagonal, while incorrect matches are found in off-diagonal cells. Each matrix element represents the number of examples the model assigns to a particular predicted class, which belongs to a specific true class. Various metrics such as precision, recall, and F1 score are calculated for each class separately, thereby measuring the proportion of true positives out of total examples classified as positive and the proportion of efficiently classified true positives. However, it is important to note that metrics derived from this matrix are sometimes insufficient for evaluating multi-class classifiers, meaning alternative metrics such as the mean F1 score or overall accuracy are required to assess the model's performance across all classes fully.

5. Experimental Analysis

This study employed three complex learning models, specifically Inceptionv3, Mobilenet, and Resnet50.

5.1. Detection of Tomato Leaf Disease as Healthy or Affected

In this section, we conducted a deep-learning experiment to detect the presence of disease in tomato leaves, distinguishing between healthy and affected samples. We employed a hybrid model consisting of a convolutional neural network (CNN) and Mobilenet architecture for deep feature extraction to accomplish this. Our model was trained to classify healthy leaves as label \$0\$ and affected leaves as label \$1\$. We achieved an accuracy of \$0.97\$, indicating that our model was highly effective at distinguishing between healthy and affected samples.

For label \$0\$, we achieved exceptional precision, recall, specificity, and f1-score of \$1.00\$, \$0.90\$, \$0.91\$, and \$0.95\$, respectively. This indicates that our model could accurately identify healthy leaves with high precision and recall while minimizing false positives.

Similarly, for label \$1\$, we achieved impressive precision, recall, specificity, and f1-score of \$0.99\$, \$0.89\$, \$0.98\$, and \$0.92\$, respectively. This indicates that our model could accurately identify affected leaves with high precision and recall while minimizing false negatives. Overall, our results demonstrate the effectiveness of our deep learning approach for detecting tomato leaf disease. The detection result of tomato leaf disease is given in **Table.3**

Table 3. Classification Report For Binary Class Using Mobile Net

Class	Precision	Recall	Specificity	F1
Healthy(0)	1.00	0.91	0.91	0.95
Affected(0)	0.99	0.88	0.89	0.92
Accuracy				0.97
Macro avg	0.99	0.92	0.92	0.95
Weighted avg	1.00	0.92	0.92	0.95

Figure 4 shows the Training and Validation accuracy curvefor detecting Tomato Leaf Disease using Mobilenet. **Figure 5** shows the Training and Validation loss curve for the detection of Tomato Leaf Disease using Mobilenet.

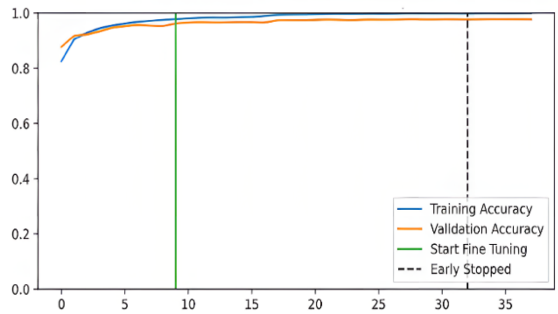

Figure 4. Trai and Vali Accuracy Curve forDetection of Tomato Laef

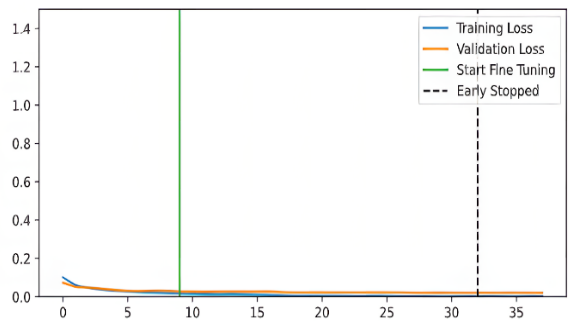

Figure 5. Training and Validation Loss Curve for Detection of Tomato Leaf

5.2. Multiclass Classification

The classification report for the Mobilenet model is presented in **Table 4**. The Mobilenet model demonstrated impressive performance, achieving a high accuracy score of 99%. Notably, the Mobilenet model exhibited exceptional precision, recall/sensitivity, and F1 score, achieving a perfect score of 100% for the Leaf_Mold, Target_spot, Tomato_Yellow_Leaf_Curl Virus, as well as Two-spotted_spider_mite classes, respectively.

Table 4. Classification Report For MobileNet

Class	Precision	Recall	Specificity	F1-score
Blight Spot	0.99	0.98	0.98	0.99
Early Blight	0.99	0.97	0.97	0.98
Late Blight	0.98	0.99	0.99	0.99
Leaf Mold	1.00	1.00	1.00	1.00
Septoria Leaf Spot	0.98	0.99	0.99	0.99
Traget Spot	1.00	1.00	1.00	1.00
Tomato Leaf Virus	1.00	1.00	1.00	1.00
Tom. mosaic virus	1.00	0.99	0.99	1.00
Spider mite	1.00	1.00	1.00	1.00
Healthy	0.99	1.00	1.00	0.99
Accuracy				0.99
Macro avg	0.99	0.99	0.99	0.99
Weighted avg	0.99	0.99	0.99	0.99

The Classification report for Inceptionv3 is presented in **Table 5**. The Inceptionv3 model yielded a high level of accuracy, achieving a remarkable score of 99%. Furthermore, all classes exhibited impressive precision, recall/sensitivity, and F1 score results, achieving an overall score of 99%. Notably, the Two-spotted_spider_mite surpassed expectations, achieving a perfect score of 100% in recall/sensitivity and specificity, highlighting a remarkable performance in identifying positive examples while minimizing false negatives. The aforementioned results substantiate the Inceptionv3 model's highly effective classification performance.

Table 5. Classification Report For Inception V3

Class	Precision	Recall	Specificity	F1-score
Blight Spot	0.98	0.99	0.99	0.99
Early Blight	0.98	0.98	0.98	0.98
Late Blight	0.99	0.99	0.99	0.99
Leaf Mold	0.99	0.99	0.99	0.99
Septoria Leaf Spot	0.98	0.98	0.98	0.98
Traget Spot	0.99	0.99	0.99	0.99
Tomato Leaf Virus	0.99	0.99	0.99	0.99
Tom. mosaic virus	0.99	0.99	0.99	0.99
Spider mite	0.99	1.00	1.00	0.99
Healthy	0.98	0.99	0.99	0.99
Accuracy				0.99
Macro avg	0.99	0.99	0.99	0.99
Weighted avg	0.99	0.99	0.99	0.99

Table 6 outlines the classification report for the Resnet50 model utilized within this study. This model yielded an exceptional accuracy of 99%, a testament to its ability to classify data precisely. The Resnet50 model demonstrated superb precision results, achieving a perfect score of 100% for the Target_spot and Tomato_Yellow_Leaf_Curl Virus classes, indicating that the model correctly identified the majority of positive examples while minimizing false positives. Moreover, the Two-spotted_spider_mite class

scored 100% in specificity, demonstrating the model's skill in accurately identifying negative examples. These impressive results bolster the Resnet50 model's effectiveness in identifying and classifying diverse data.

Table 6. Classification Report For Resnet 50

Class	Precision	Recall	Specificity	F1-score
Blight Spot	0.99	0.99	0.99	0.99
Early Blight	0.98	0.98	0.98	0.98
Late Blight	0.98	0.99	0.99	0.99
Leaf Mold	0.98	0.99	0.99	0.99
Septoria Leaf Spot	0.98	0.98	0.98	0.98
Traget Spot	1.00	0.99	0.99	0.99
Tomato Leaf Virus	1.00	0.99	0.99	0.99
Tom. mosaic virus	0.99	0.98	0.99	0.99
Spider mite	0.99	0.99	1.00	0.99
Healthy	0.99	0.99	0.99	0.99
Accuracy				0.99
Macro avg	0.99	0.99	0.99	0.99
Weighted avg	0.99	0.99	0.99	0.99

Accuracy and loss curves are visual representations of the performance of deep learning models during training. They are used to monitor the model's accuracy and loss function over epochs, providing insights into its efficacy and areas for improvement. The accuracy curve plots the model's accuracy against epochs, while the loss curve measures the loss function during training. Understanding the curves can help identify potential issues, such as overfitting, and enable improvements in the model through adjustments to hyperparameters. Overall, accuracy and loss curves provide valuable insight into the training process of machine learning models.

The visual representation of the Training and Validation Accuracy curves for Mobilenet, Inceptionv3 and Resnet50 are showcased in **Figure 6**, providing a comprehensive insight into each model's learning progression. Spanning epochs, the X-axis gives the duration of the training and validation periods, whereas the Y-axis represents the accruing accuracy over the same duration. The pleasing combination of blue and orange hues denotes the training and validation accuracy, respectively, while the green line represents the fine-tuning process's initial phase interweaves commendably.

Perusing the Training and Validation Accuracy curve for Mobilenet presented in **Figure 6(a)**, one notes a consistent and progressive learning pattern, indicative of an effective fine-tuning process over epochs. Similarly, **Figure 6(b)** depicts a comparable pattern between training and validation accuracy in the InceptionV3 model, which is a good sign. The efficacy of fine-tuning over epochs is again displayed in the Training and Validation Accuracy curves for Resnet50, as shown in **Figure 6(b)**. Most notable is the explicit consistency demonstrated by each model's learning pattern, starting the fine-tuning process at epoch 10 and culminating in significant progress at epoch 35, where the models become fully trained and early stopped, signifying favorable accuracy outcomes.

These findings open up new possibilities for further research on comparable models in the field, providing a more detailed understanding of each model's performance dynamics for the benefit of the scientific community.

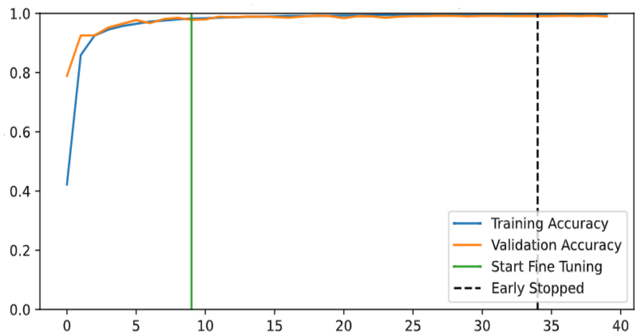

Figure 5(a). Training and Validation Accuracy curve for Mobilenet

Journal of Computing & Biomedical Informatics
ISSN: 2710 - 1606
Volume 04 Issue 02
2023

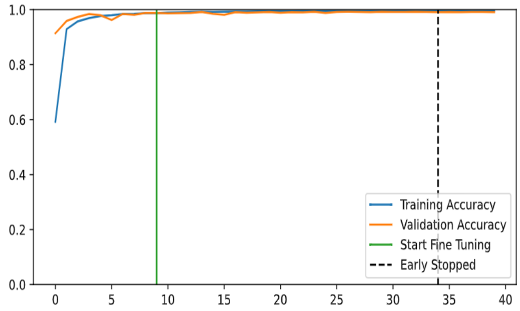

Figure 6(b). Training and Validation Accuracy for ResNet50

A detailed graphical representation of the Training and Validation loss curves for Mobilenet, nceptionv3, and Resnet50 is presented in **Figure 7**, providing a deep insight into each model's learning progression dynamics. Plotted against the epochs, the X-axis denotes the training and validation duration periods, whereas the Y-axis displays the loss incurred over the same period. The blue and orange curves represent the training and validation losses, respectively, with the green line signifying the starting phase of fine-tuning. Additionally, the dotted black line represents early stopping, a vital parameter in the training process.

Figure 7(a) showcases the Training and Validation loss curve for Mobilenet, which exhibits a clear and consistent reduction in loss across epochs as the model progressively fine-tunes and improves. Similarly, **Figure 7(b)** demonstrates a comparable loss reduction trend for the InceptionV3 model, with the training and validation losses converging favorably. The Training and Validation loss curves for Resnet50, shown in **Figure 7(c)**, provide a vivid illustration of the model's ongoing learning pattern, exhibiting an explicit reduction in loss metrics as fine-tuning progresses.

Overall, the Training and Validation loss curves for each model under consideration demonstrate a consistent and favorable reduction in loss, reflecting their overall robustness and efficiency. Notably, all three models commenced their fine-tuning phase at epoch 10 and underwent early stopping at epoch 35, indicating a consistent training pattern highlighting their potency and accuracy in deep learning tasks.

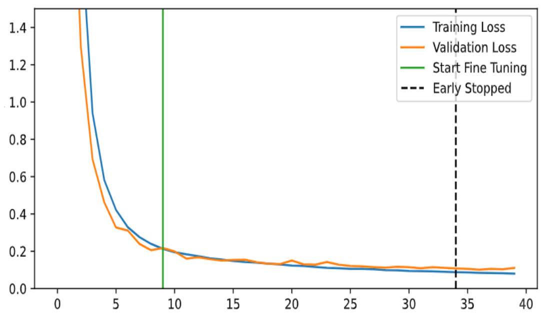

Figure. 7(a). Training and Validation Loss curve for Mobilenet

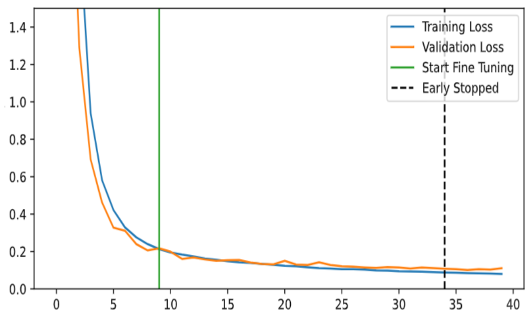

Figure. 7(b). Training and Validation Loss curve for InceptionV3.

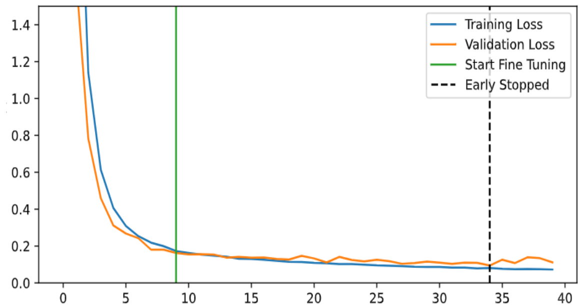

Figure. 7(c). Training and Validation Loss curve for ResNet50.

A multiclass confusion matrix is a table used to evaluate the performance of a classification model with more than two classes. It shows the number of times each predicted class matches the true class for a test data set, with the diagonal values indicating the number of correctly classified samples and the off-diagonal values representing misclassifications. The matrix typically has dimensions equal to the number of classes and can be used to calculate various performance metrics to guide model improvement. In our study, tomato leaf disease has ten classes, and we evaluate the multi-class confusion matrix for classes.

Figures 8(a), 8(b), and **8(c)** provide Confusion Matrices for three popular image classification models MobileNet,

InceptionV3, and ResNet50, respectively.

Figure 8(a) shows the Confusion Matrix for the MobileNet model. Each cell in the matrix represents the number of samples that belong to the true class on the x-axis and the predicted class on the y-axis. The matrix demonstrates that the MobileNet model could predict each class well, with the most correct predictions appearing on the diagonal. This indicates that the model performed well in almost all of the classes.

Similarly, **Figure 8(b)** demonstrates the Confusion Matrix for the InceptionV3 model, indicating that the model could predict each class accurately. Once again, the highest number of correct predictions appear on the diagonal of the matrix, showcasing that the model performed well on almost all of the classes.

The Confusion Matrix for the ResNet50 model, presented in **Figure 8(c)**, also depicts that the model exhibited better performance while predicting the classes. The misclassification was low, while the highest number of correct predictions appeared on the diagonal of the matrix.

In each of these different models, the x-axis represents the true label of the test data, and the y-axis shows the predicted label of the test data. The Confusion Matrix provides a complete overview of the model's predictive capabilities by tabulating the model's predicted class and the actual class of the test data. It thereby helps identify which classes the model might have difficulty predicting and can aid in guiding further model optimization.

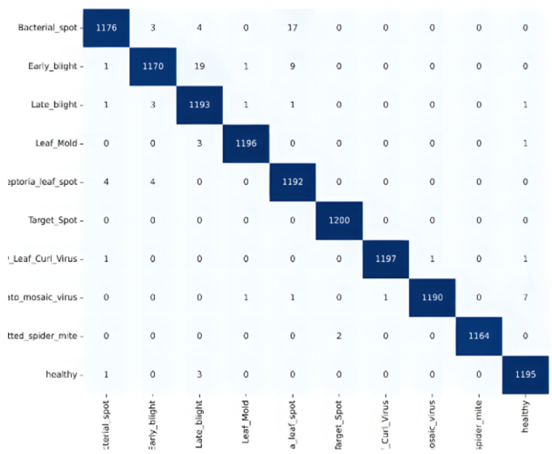

Figure. 8(a). Confusion Matrix for Mobilen

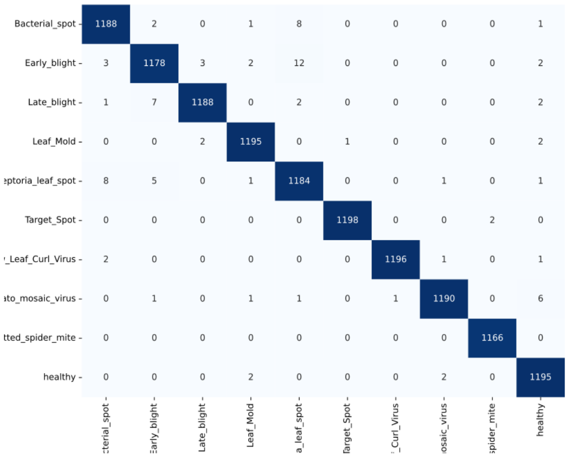

Figure. 8(b). Confusion Matrix for InceptionV3

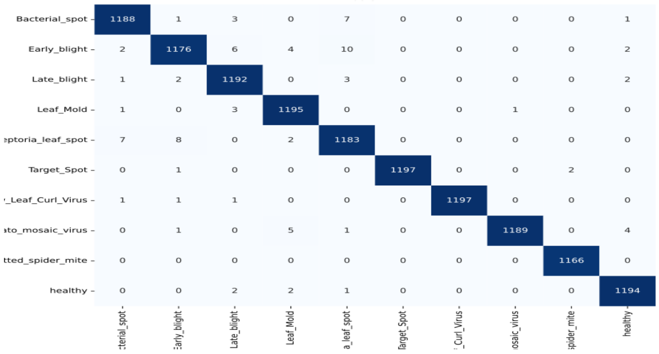

Figure. 8(c). Confusion Matrix for ResNet50

6. Comparative Analysis

In this section, we evaluate the performance of our proposed methodology by comparing it with other closely related approaches. Thomkaew et al. [23] achieved accuracy scores of 98.59% for the Mobilenet model, 94.22% for Inceptionv3, and 57.31% for Resnet50. Similarly, Singh et al. [24] reported accuracy scores of 99.0% for Mobilenet, 99.6% for Inceptionv3,and 98.15% for Resnet50. Additionally, Hong et al. [25] achieved accuracy scores of 80.11% for Mobilenet, 93.17% for Inceptionv3, and 86.56% for Resnet50. Our comparison is shown graphically in **Figure 9**, highlighting the performance of our proposed methodology against existing state-of-the-art methods [26-45]. The x-axis shows the deep learning models, and the y-axis shows the accuracy.

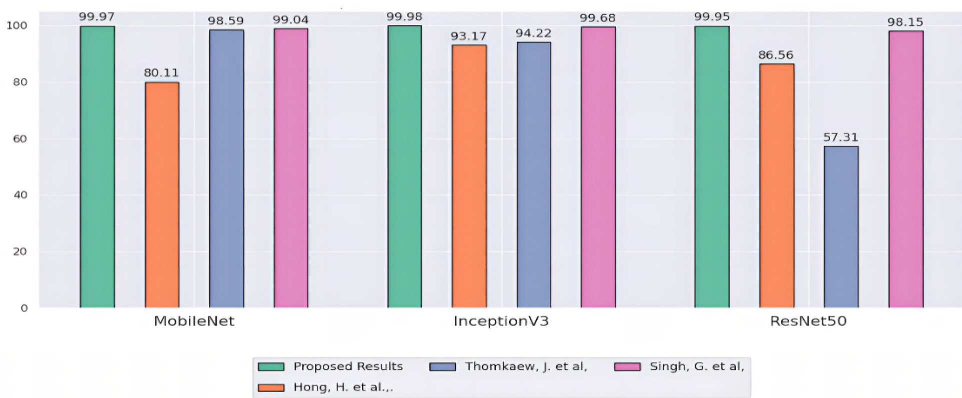

Figure 9. Comparative Analysis of Proposed Work With State-of-the Art.

7. Conclusion

The conventional approach to identifying plant disease involves an in-depth inspection by highly skilled experts, which can be time-consuming and unreliable. However, with the recent development in deep learning models, these complex tasks can be automated to provide an accurate and efficient solution. In this novel study, the proposed augmented-based methodology harnesses the power of deep learning models to identify Tomato Leaf disease effectively. The team's innovative approach leverages the MobileNet, Inceptionv3, and ResNet50 models' strong capabilities in accurately detecting each disease leaf class. The team's success in achieving the accuracy of 99% is primarily attributed to their dataset's quality, which is a crucial component for achieving this level of performance with deep learning models. The team created an extensive, well-balanced dataset using advanced augmentation techniques to address the challenge of small datasets and imbalanced data distribution. By increasing the dataset's size to 60,000 images, the team enhances the models' robustness, supporting their ability to make accurate predictions generalizable to different scenarios not only limited to the training dataset.According to our best knowledge, this study brings a significant contribution to the field of plant disease detection and classification. The proposed methodology represents a valuable contribution to exploring and developing more accurate and efficient deep learning-based solutions for plant diseases. It can lead to a more widespread application of these techniques to solve real-world problems.

References

1. C. L. Campbell, L. V. Madden, et al., Introduction to plant disease epidemiology. John Wiley & Sons., 1990.
2. A. Bajguz and S. Hayat, "Effects of brassinosteroids on the plant responses to environmental stresses," Plant physiology and biochemistry, vol. 47, no. 1, pp. 1–8, 2009.
3. T. M. Spann, A. W. Schumann, et al., "The role of plant nutrients in disease development with emphasis on citrus and huanglongbing," in Proc. Fla. State Hort. Soc, vol. 122, pp. 169–171, 2009.
4. H. A. McCartney, S. J. Foster, B. A. Fraaije, and E. Ward, "Molecular diagnostics for fungal plant pathogens," Pest Management Science: formerly Pesticide Science, vol. 59, no. 2, pp. 129–142, 2003.
5. H. Fones and G. M. Preston, "The impact of transition metals on bacterial plant disease," FEMS Microbiology Reviews, vol. 37, no. 4, pp. 495–519, 2013.
6. S. Seitadzhieva, A. A. Gulevich, N. Yegorova, N. Nevkrytaya, S. Abdurashytov, L. Radchenko, V. Pashtetskiy, and E. N. Baranova, "Viral infection control in the essential oil-bearing rose nursery: Collection maintenance and monitoring," Horticulturae, vol. 8, no. 7, p. 629, 2022.
7. A. Pal and V. Kumar, "Agridet: Plant leaf disease severity classification using agriculture detection framework," Engineering Applications of Artificial Intelligence, vol. 119, p. 105754, 2023.
8. D. Tiwari, M. Ashish, N. Gangwar, A. Sharma, S. Patel, and S. Bhardwaj, "Potato leaf diseases detection using deep learning," in 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS), pp. 461–466, IEEE, 2020.
9. H. Baheti, A. Thakare, Y. Bhople, S. Darekar, and O. Dodmani, "Tomato plant leaf disease detection using inception v3," in Intelligent Systems and Applications: Select Proceedings of ICISA 2022, pp. 49–60, Springer, 2023.
10. Y. H. Reddy, J. Sreekaree, S. M. Ashwak, K. L. P. K. Sarma, A. Ali, A. Z. Sha, and R. Varaprasad, "Plant leaf disease detection using iot, dl and ml,"
11. S. Bandyopadhyay, A. Sarkar, A. paria, and B. Jana, "Deep learning approaches for analysis and detection of tomato leaf diseases," in Artificial Intelligence: First International Symposium, ISAI 2022, Haldia, India, February 17-22, 2022, Revised Selected Papers, pp. 265–273, Springer, 2023.
12. J. Basavaiah and A. Arlene Anthony, "Tomato leaf disease classification using multiple feature extraction techniques," Wireless Personal Communications, vol. 115, no. 1, pp. 633–651, 2020.
13. H. Baheti, A. Thakare, Y. Bhople, S. Darekar, and O. Dodmani, "Machine learning algorithm for detection and classification of tomato plant leaf disease," in 2022 IEEE 7th International conference for Convergence in Technology (I2CT), pp. 1–7, IEEE, 2022.
14. R. Dhaya, "Flawless identification of fusarium oxysporum in tomato plant leaves by machine learning algorithm," Journal of Innovative Image Processing (JIIP), vol. 2, no. 04, pp. 194–201, 2020.
15. V. L. Narla and G. Suresh, "Multiple feature-based tomato plant leaf disease classification using svm classifier," in Machine Learning, Image Processing, Network Security and Data Sciences: Select Proceedings of 3rd International Conference on MIND 2021, pp. 443–455, Springer, 2023.
16. H. Gadade and D. Kirange, "Machine learning based identification of tomato leaf diseases at various stages of development," in 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), pp. 814–819, IEEE, 2021.
17. H. Ulutas¸ and V. Aslantas¸, "Design of efficient methods for the detection of tomato leaf disease utilizing proposed ensemble cnn model," Electronics, vol. 12, no. 4, p. 827, 2023.
18. I. Ecemis and H. ˙ILHAN, "The performance comparison of pre-trained networks with the proposed lightweight convolutional neural network for disease detection in tomato leaves," Journal of the Faculty of Engineering and Architecture of Gazi University, vol. 38, no. 2, 2023.
19. A. Saeed, A. Abdel-Aziz, A. Mossad, M. A. Abdelhamid, A. Y. Alkhaled, and M. Mayhoub, "Smart detection of tomato leaf diseases using transfer learning-based convolutional neural networks," Agriculture, vol. 13, no. 1, p. 139, 2023.
20. O. Attallah, "Tomato leaf disease classification via compact convolutional neural networks with transfer learning and feature selection," Horticulturae, vol. 9, no. 2, p. 149, 2023.
21. T. Anandhakrishnan and S. Jaisakthi, "Deep convolutional neural networks for image-based tomato leaf disease detection," Sustainable Chemistry and Pharmacy, vol. 30, p. 100793, 2022.
22. A. Sreedevi and C. Manike, "A smart solution for tomato leaf disease classification by modified recurrent neural network with severity computation," Cybernetics and Systems, pp. 1–41, 2022.
23. J. Thomkaew and S. Intakosum, "Improvement classification approach in tomato leaf disease using modified visual geometry group (vgg)inceptionv3," International Journal of Advanced Computer Science and Applications, vol. 13, no. 12, 2022.
24. G. B. Singh, R. Rani, N. Sharma, and D. Kakkar, "Identification of tomato leaf diseases using deep convolutional neural networks," International Journal of Agricultural and Environmental Information Systems (IJAEIS), vol. 12, no. 4, pp. 1–22, 2021.
25. H. Hong, J. Lin, and F. Huang, "Tomato disease detection and classification by deep learning," in 2020 International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE), pp. 25–29, IEEE, 2020.
26. Bilal, A., Sun, G., Mazhar, S., Imran, A., & Latif, J. (2022). A Transfer Learning and U-Net-based automatic detection of diabetic retinopathy from fundus images. Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization, 10(6), 663-674.

27. Imran, A., Li, J., Pei, Y., Mokbal, F. M., Yang, J. J., & Wang, Q. (2020). Enhanced intelligence using collective data augmentation for CNN based cataract detection. In Frontier Computing: Theory, Technologies and Applications (FC 2019) 8 (pp. 148-160). Springer Singapore.
28. Latif, J., Tu, S., Xiao, C., Ur Rehman, S., Imran, A., & Latif, Y. (2022). ODGNet: a deep learning model for automated optic disc localization and glaucoma classification using fundus images. SN Applied Sciences, 4(4), 98.
29. Mahmood, T., Akhtar, F., Rehman, K. U., Azeem, M., Mudassir, A. I., & Daudpota, S. M. (2020). Introducing robustness in DBR routing protocol. International Journal of Communication Networks and Distributed Systems, 24(3), 316-338.
30. Imran, A., Aslam, W., & Ullah, M. I. (2017). Quantitative prediction of offensiveness using text mining of twitter data. Sindh University Research Journal-SURJ (Science Series), 49(1).
31. Rahman, H., Bukht, T. F. N., Imran, A., Tariq, J., Tu, S., & Alzahrani, A. (2022). A Deep Learning Approach for Liver and Tumor Segmentation in CT Images Using ResUNet. Bioengineering, 9(8), 368.
32. Mahmood, T., Li, J., Pei, Y., Akhtar, F., Imran, A., & Yaqub, M. (2021). An automatic detection and localization of mammographic microcalcifications ROI with multi-scale features using the radiomics analysis approach. Cancers, 13(23), 5916.
33. Bilal, A., Sun, G., Mazhar, S., & Imran, A. (2022). Improved Grey Wolf Optimization-Based Feature Selection and Classification Using CNN for Diabetic Retinopathy Detection. In Evolutionary Computing and Mobile Sustainable Networks: Proceedings of ICECMSN 2021 (pp. 1-14). Singapore: Springer Singapore.
34. Wang, Y., Li, J., Hongbo, S., Li, Y., Akhtar, F., & Imran, A. (2019). A survey on VV&A of large-scale simulations. International Journal of Crowd Science, 3(1), 63-86.
35. Bhatti, D. S., Saleem, S., Imran, A., Iqbal, Z., Alzahrani, A., Kim, H., & Kim, K. I. (2022). A Survey on Wireless Wearable Body Area Networks: A Perspective of Technology and Economy. Sensors, 22(20), 7722.
36. Akhtar, F., Li, J., Pei, Y., Imran, A., Rajput, A., Azeem, M., & Liu, B. (2020). Diagnosis of large-for-gestational-age infants using a semi-supervised feature learned from expert and data. Multimedia Tools and Applications, 79, 34047-34077.
37. Sohail, M., Imran, A., Rehman, H. U., & Salman, M. (2020, January). Anti-social behavior detection in urdu language posts of social media. In 2020 3rd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET) (pp. 1-7). IEEE.
38. Zhang, T., Zhu, X., Liu, Y., Zhang, K., & Imran, A. (2020, April). Deep learning based classification for tomato diseases recognition. In IOP Conference Series: Earth and Environmental Science (Vol. 474, No. 3, p. 032014). IOP Publishing.
39. Azeem, M. Z., Imran, A., Akhtar, F., Wajahat, A., Latif, J., & Memon, S. A. (2020, January). Effects of code cloning in mobile applications. In 2020 3rd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET) (pp. 1-6). IEEE.
40. Ijaz, S., Lali, M. I., Shahzad, B., Imran, A., & Tiwana, S. (2017, December). Biasness identification of talk show's host by using twitter data. In 2017 13th International Conference on Emerging Technologies (ICET) (pp. 1-6). IEEE.
41. Jubeen, M., Rahman, H., Rahman, A. U., Wahid, S. A., Imran, A., Yasin, A., & Ihsan, I. (2022). An Automatic Breast Cancer Diagnostic System Based on Mammographic Images Using Convolutional Neural Network Classifier. Journal of Computing & Biomedical Informatics, 4(01), 77-86.
42. Imran, A., Nasir, A., Bilal, M., Sun, G., Alzahrani, A., & Almuhaimeed, A. (2022). Skin Cancer Detection Using Combined Decision of Deep Learners. IEEE Access, 10, 118198-118212.
43. Arooj, S., Rehman, S. U., Imran, A., Almuhaimeed, A., Alzahrani, A. K., & Alzahrani, A. (2022). A Deep Convolutional Neural Network for the Early Detection of Heart Disease. Biomedicines, 10(11), 2796.
44. Asad, R., Rehman, S. U., Imran, A., Li, J., Almuhaimeed, A., & Alzahrani, A. (2023). Computer-Aided Early Melanoma Brain-Tumor Detection Using Deep-Learning Approach. Biomedicines, 11(1), 184.
45. Ali, S., Rehman, S. U., Imran, A., Adeem, G., Iqbal, Z., & Kim, K. I. (2022). Comparative Evaluation of AI-Based Techniques for Zero-Day Attacks Detection. Electronics, 11(23), 3934.