



BANGLADESH UNIVERSITY OF BUSINESS AND TECHNOLOGY (BUBT)

A project report on

Tomato Plant Leaf and Disease Recognition

COURSE CODE: CSE 476

COURSE TITLE: Data Mining Lab

SUBMITTED BY

Hasan Al Mahmud	18192103269
Hashibur Rahman	18192103276
Shakil Ahamed Raju	15162103136
Rifatul Islam	18192103187
Tahamina Akther Ruma	18192103225
Jannatul Kobra	17182103296

SUBMITTED TO

Khan Md Hasib

Assistant Professor

BUBT

Submission Date - 01.03.2023

Acknowledgement

We would like to express our heartfelt gratitude to the almighty Allah who offered upon our family and us kind care throughout this journey until the fulfilment of this research.

Also, we express our sincere respect and gratitude to our supervisor, Khan Md Hasib, Assistant professor, Department of Computer Science and Engineering, Bangladesh University of Business and Technology (BUBT). Without his guidance, this research work would not exist. We are grateful to him for his excellent supervision and for putting his utmost effort into developing this project. We owe him a lot for his assistance, encouragement and guidance, which has shaped our mentality as a researcher.

Finally, we are grateful to all our faculty members of the CSE department, BUBT, to make us compatible to complete this research work with the proper guidance and supports throughout the last four years.

Abstract

Humans depend heavily on agriculture, which is the main source of prosperity. The various plant diseases that farmers must contend with have constituted a lot of challenges in crop production. The main issues that should be taken into account for maximizing productivity are the recognition and prevention of plant diseases. Early diagnosis of plant disease is essential for maximizing the level of agricultural yield as well as saving costs and reducing crop loss. In addition, the computerization of the whole process makes it simple for implementation. In this paper, an intelligent method based on deep learning is presented to recognize nine common tomato diseases. To this end, a residual neural network algorithm is presented to recognize tomato diseases. This research is carried out on four levels of diversity including depth size, discriminative learning rates, training and validation data split ratios, and batch sizes. For the experimental analysis, five network depths are used to measure the accuracy of the network. Based on the experimental results, the proposed method achieved the highest F1 score of 99.5%, which outperformed most previous competing methods in tomato leaf disease recognition. Further testing of our method on the Flavia leaf image dataset resulted in a 99.23% F1 score. However, the method had a drawback that some of the false predictions were of tomato early light and tomato late blight, which are two classes of fine-grained distinction.

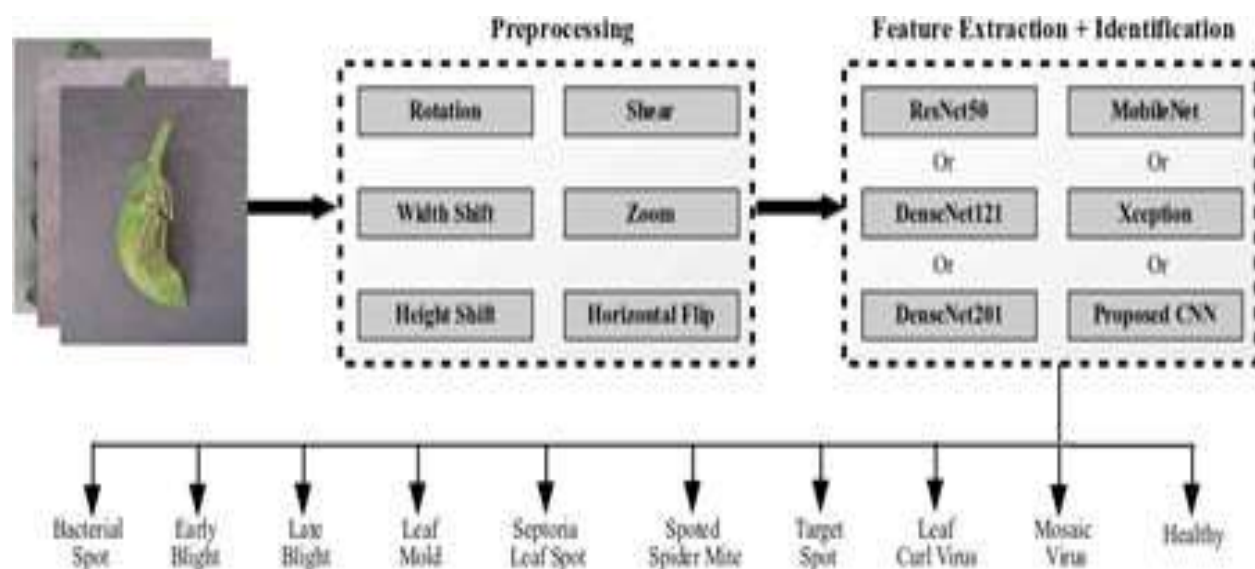
INTRODUCTION

Tomato plants are a widely cultivated crop, known for their high nutritional value and versatile use in cooking. However, tomato plants are also susceptible to various diseases that can significantly reduce yields and quality of the crop. One important aspect of managing and preventing these diseases is the ability to accurately recognize and diagnose them in a timely manner. The use of artificial intelligence (AI) and deep learning can provide fast and accurate identification of diseases, even in the early stages, which can lead to better management and control of the diseases. Tomato plant leaf and disease recognition is an important and timely problem in agricultural research. With the help of machine learning, it can help to identify the disease accurately and quickly, which can help farmers to take timely measures to control the disease and protect their crops. It can also help scientists to better understand the pathogen and develop effective treatments, thus improving the yield and quality of crops.

METHODOLOGY

There are various techniques available for crop disease identification, but deep learning techniques are mostly used now a day as discussed in the introduction section. Therefore, this study uses a deep convolutional neural network for tomato crop diseases identification. The overall system for tomato crop disease identification involves three major steps namely data preprocessing, feature extraction, and identification. Each step is briefly discussed in this section. Figure 1 shows the workflow diagram of this work.

Figure 1.
Workflow diagram



1. Data collection: The first step in tomato plant leaf and disease recognition is to collect data. This data should include images of tomato leaves with different diseases, as well as labels indicating the type of disease present.
2. Image preprocessing: Image preprocessing is necessary to prepare the images for analysis. This includes cropping, resizing, contrast enhancement, and noise reduction.
3. Feature extraction: Feature extraction is used to extract relevant features from the images. This can be done using techniques such as local binary pattern (LBP) or Haar wavelet transform.
4. Model building: Once the features have been extracted, a model can be built to recognize the disease. This can be done using techniques such as support vector machine (SVM) or convolutional neural networks (CNN).

Data Preprocessing

The preprocessing technique is used to enhance the image data. The deep learning model trained on enhanced images can improve the identification accuracy of the model and overcome it from overfitting (Jiang, Peng, et al., 2019). If every image has different variants then, the model can learn more irrelevant patterns during the training phase. The expansion of the dataset by creating different variants of the existing image is called data augmentation. In this step; we have used width shift, height shift, rotation, horizontal flip, shear, and zoom as a data augmentation parameter. We have chosen a parameter range between 0.1-0.5 that is maximum 5% by assuming that this range will not actually change the shape of the image. Then we tested the performance of the model with a different combination of values in the given range. And the values of each parameter on which the model performed better are listed in Table 1.

Parameter	Value
Rotation range	5
Width shift range	0.1
Height shift range	0.1
Shear range	0.2
Zoom range	0.1
Fill mode	'nearest'
Horizontal flip	True

A shift in an image refers to moving pixels in one direction either towards height or width, while the dimension of an image remains the same. A horizontal flip means exchanging row pixels with column pixels and vice-versa. Images are randomly rotated in a clockwise direction using a rotation parameter. Zoom parameter in an augmentation process randomly zoom in an image or insert new pixels around it. The original images are resized to 128×128 pixels to minimize training time. Original images are in RGB format. RGB coefficients range from 0 to 255, which will be higher and complex for our model to learn. Therefore, coefficients are rescaled between 0 and 1 by multiplying each pixel with 1/255.

Pretrained Deep CNNs

This section discusses standard pre-trained deep convolutional neural networks, which have been used to compare the performance of the proposed deep convolutional neural network. The pre-trained models are already trained on a large benchmark dataset ‘ImageNet’. These models don’t need to train from scratch. Some of the well-known pre-trained DCNNs models are AlexNet, VGG, DenseNet, MobileNet, and Xception. These models are differing by their configurations, kernel size, depth, and a number of neurons. This study uses top-5 DCNNs models having higher identification accuracy on the tomato leaf dataset. The top-5 DCNNs having higher identification accuracy are ResNet50, DenseNet121, DenseNet201, MobileNet, and Xception, which are discussed in detail in this section.

Proposed Work

This research work proposed a deep convolutional neural network model to recognize different types of diseases in tomato crops. The proposed model is influenced by the standard pre-trained model AlexNet. According to a review of the state of the art model for tomato disease identification, AlexNet has higher identification accuracy, but the number of parameter is more. Hence, AlexNet is considered as the basic network for this work. This work improves the recognition accuracy of model and minimizes the network parameter. The AlexNet model is less deep than ResNet, DenseNet, Inception, MobileNet, and Xception, but the number of parameters is much higher than these models. So, the proposed model minimizes the number of parameters by changing number of neurons, kernel size and number of kernels, and rearranging max-pooling and convolution layers. The AlexNet and proposed model have been briefly discussed in this section.

Proposed Deep CNN

The proposed model consists of convolution, Maxpooling, batch normalization, flattening, dropout, and fully connected layer. Figure 6 shows the detailed configuration of the proposed deep convolutional neural network.

The main aim of convolution operation is to extract features like corners, edges, and colors from an image. The convolution operation is performed by continuous sliding of filter (kernel) over image pixels and taking the dot product of the corresponding pixel of filter and input image pixel. The proposed model contains six convolution layers with a kernel size of 3×3 and each followed by a rectified linear unit (ReLU) activation function. The ReLU activation function has been used to make the input neuron capable of learning more complex and complicated features. The ReLU activation function also rectifies the vanishing gradient problem. Due to the increase in number of convolution layers, the network parameter increases exponentially. So, pooling is performed to decrease the dimension of the feature-map. It extracts essential features from the feature map by removing non-essential features. The proposed model uses max-pooling, due to its better performance and greater convergence. The max-pooling is done by simply taking the max value in the pooling window. The training of a deep convolutional neural network is a challenging task due to the overfitting problem. There are mainly two ways to overcome from over fitting namely regularization and dropout operation. Therefore, batch normalization has been used to include the regularization effects in the network. It also boosts the training speed and performance of the model. The batch normalization mainly standardizes the input by scaling it in a similar range. In the proposed network, batch normalization operation has been performed after every activation operation and a dense layer. The model also includes dropout operations with a dropout rate of 25%. Dropout operation refers to the deactivating some randomly chosen neurons during training. It means, temporarily deactivates neurons from the network and also its incoming and outgoing edges.

Deep CNN advantages

The advantages of the proposed model over the standard pre-trained models are as follows-

- It contains less number of parameters, hence requires less time for training.
- There is very little chance of vanishing information before reaching to the output layer of the network as the proposed architecture is less deep.
- The batch normalization feature prevents over fitting and also makes the network training process faster.
- The error rate for prediction of disease is also very less as compare to other networks, while it has a lesser number of parameters.

Layer Name	Input Shape	Output Shape	# Parameters
Conv2D_1	(128, 128, 3)	(126, 126, 12)	3584
Conv2D_2	(126, 126, 128)	(124, 124, 128)	147584
Maxpooling2D_1	(124, 124, 128)	(62, 62, 128)	0
Batch_Normalization_1	(62, 62, 128)	(62, 62, 128)	512
Conv2D_3	(62, 62, 128)	(60, 60, 256)	295168
Conv2D_4	(60, 60, 256)	(58, 58, 256)	590080
Maxpooling2D_2	(58, 58, 256)	(29, 29, 256)	0
Batch_Normalization_2	(29, 29, 256)	(29, 29, 256)	1024
Conv2D_5	(29, 29, 256)	(27, 27, 384)	885120
Conv2D_6	(27, 27, 384)	(25, 25, 256)	884992
Maxpooling2D_3	(25, 25, 256)	(12, 12, 256)	0
Batch_Normalization_3	(12, 12, 256)	(12, 12, 256)	1024
Flatten_1	(12, 12, 256)	36864	0
Dense_1	36864	512	18874880
Dropout_1	512	512	0
Batch_Normalization_4	512	512	2048
Dense_2	512	10	5130

Resizing and normalizing

The various CNN network has input image size requirements. Thus, the images were resized to 299×299 for Inceptionv3 and 224×224 for Resnet18, MobilenetV2, and DenseNet201. Using the mean and standard deviation of the images of the dataset, z-score normalization was used to normalize the images.

Augmentation

Training with an imbalanced dataset will result in a biased model because the dataset is not balanced and does not contain a comparable number of images for the various categories. As a result, data augmentation can aid in the creation of a similar number of images in each class, resulting in reliable results, as reported in numerous recent publications [6, 7, 8, 9, 10, 11]. To align the training images, three augmentation techniques (rotation, scaling, and translation) were used. The images were rotated in a clockwise and counterclockwise direction with an angle of 5 to 15 degrees for image augmentation. The scaling process involves enlarging or shrinking the image's frame size, and 2.5 percent to 10% image magnifications were used in this analysis. Image translation was accomplished by converting images by 5–20% horizontally and vertically.

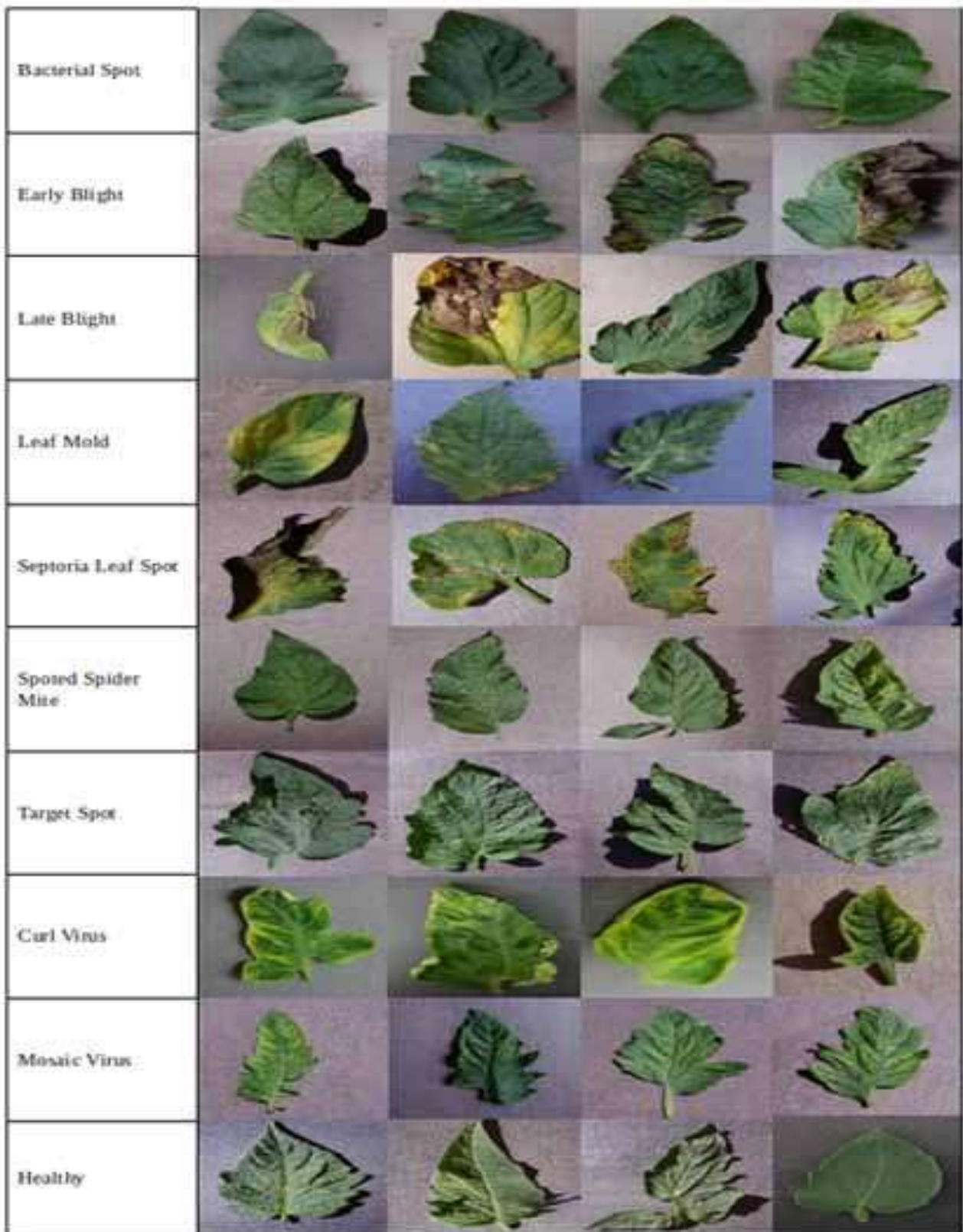
Dataset

We need a suitable dataset at every step of our research work. So, we retrieved the plantVillage dataset from the publicly available repository plantVillage organization (<https://plantvillage.psu.edu/>). The plantVillage organization helps farmers working in a remote area. The dataset contains 18,160 images of diseased and healthy tomato leaves. Each image is of size 256×256 pixels. The images are labeled under the supervision of an agricultural expert, according to their disease categories(TM, Prajwala, et al., 2018). Images are divided into 10 classes, in which nine classes contain diseased leaf images and one class contains healthy images. Classes representing nine categories of diseases are bacterial spot, early blight, curl virus, septoria leaf spot, mosaic virus, target spot, spotted spider mite, late blight, and leaf mold. Figure 7 shows the sample images of various tomato leaf diseases.

Experimental Setup

The proposed methodology is implemented on the google cloud platform. Google provides free GPU resources for AI developers, which is known by Google Colab. Google Colab uses a jupyter notebook environment. The hardware specification of Google Colab is as follows: uses 1×Tesla K80 GPU, 13 GB GDDR5 VRam, and 33 GB disk space. The models are implemented using the Keras library and Tensorflow framework. The dataset used for this research work contains 18,160 images of 9 common tomato leaf diseases and a healthier one. The dataset is split into training and validation sample in the percentage split ratio of 75:25. Table 3 lists the count of training and testing sample details of each class label.

Sample images of the tomato leaf dataset



Accuracy and Loss Comparison

To compare the performance of the proposed deep convolutional neural network various standard pre-trained deep convolutional neural networks, ResNet50, DenseNet121, DenseNet201, and are applied for tomato crop diseases identification. This study mainly uses the accuracy and loss parameter for the evaluation of models.

The models are trained on training image samples and tested on test image samples to compare the performance of models. Throughout the training process, Adam optimizer is used for the pre-trained model and stochastic gradient descent (SGD) with momentum is used for the proposed deep convolutional neural network.

The Adam optimizer is basically considered as a combination of stochastic gradient descent (SGD) and RMSprop with momentum. Like RMSprop, it uses a squared gradient to map the learning rate. And like stochastic gradient descent it uses the moving average of the gradient in place of using the gradient itself. The optimizer randomly selects a group of images for training, which is known as batch size. The batch size value depends on the capacity of resources used for training. This work uses a batch size value of 32. The proposed model uses a learning rate value of 0.01 with a momentum value of 0.09. And pre-trained models use learning rate value as 0.001. The momentum defines how fast gradients move towards the optimum point. To find the appropriate weight for the network, weight is updated using a back propagation algorithm.

Details of training and validation samples of tomato leaf diseases

Disease	Class Label	Training Images	Validation Images	Total Images
Bacterial Spot	A	1596	531	2127
Early Blight	B	750	250	1000
Late Blight	C	1432	477	1909
Leaf Mold	D	714	238	952
Septoria Leaf Spot	E	1329	442	1771
Spoted Spider Mite	F	1257	419	1676
Target Spot	G	1053	351	1404
Leaf Curl Virus	H	4018	1339	5357
Mosaic Virus	I	280	93	373
Healthy	J	1194	397	1591

Evaluation Metrics, Results, and Discussion

This section presents the metrics used in evaluating the results of this research, the detailed results, and relevant discussion

Evaluation Metrics

The accuracy, precision, recall, and f1-score of the proposed method were all evaluated. The proposed plant recognition system's accuracy has been calculated using the following expression, which incorporates numerical details such as true positive (TP) (the number of correctly identified leaf images), false positive (FP) (the number of incorrectly detected leaves), true negative (TN) (the number of correctly detected leaf images), and false negative (FN) (it is a parameter for representation of the number of leaf images that are correctly recognized).

Accuracy: Accuracy is the number of right predictions that are made by the model with respect to the total number of predictions that were made. It is mathematically represented by Equation (1).

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

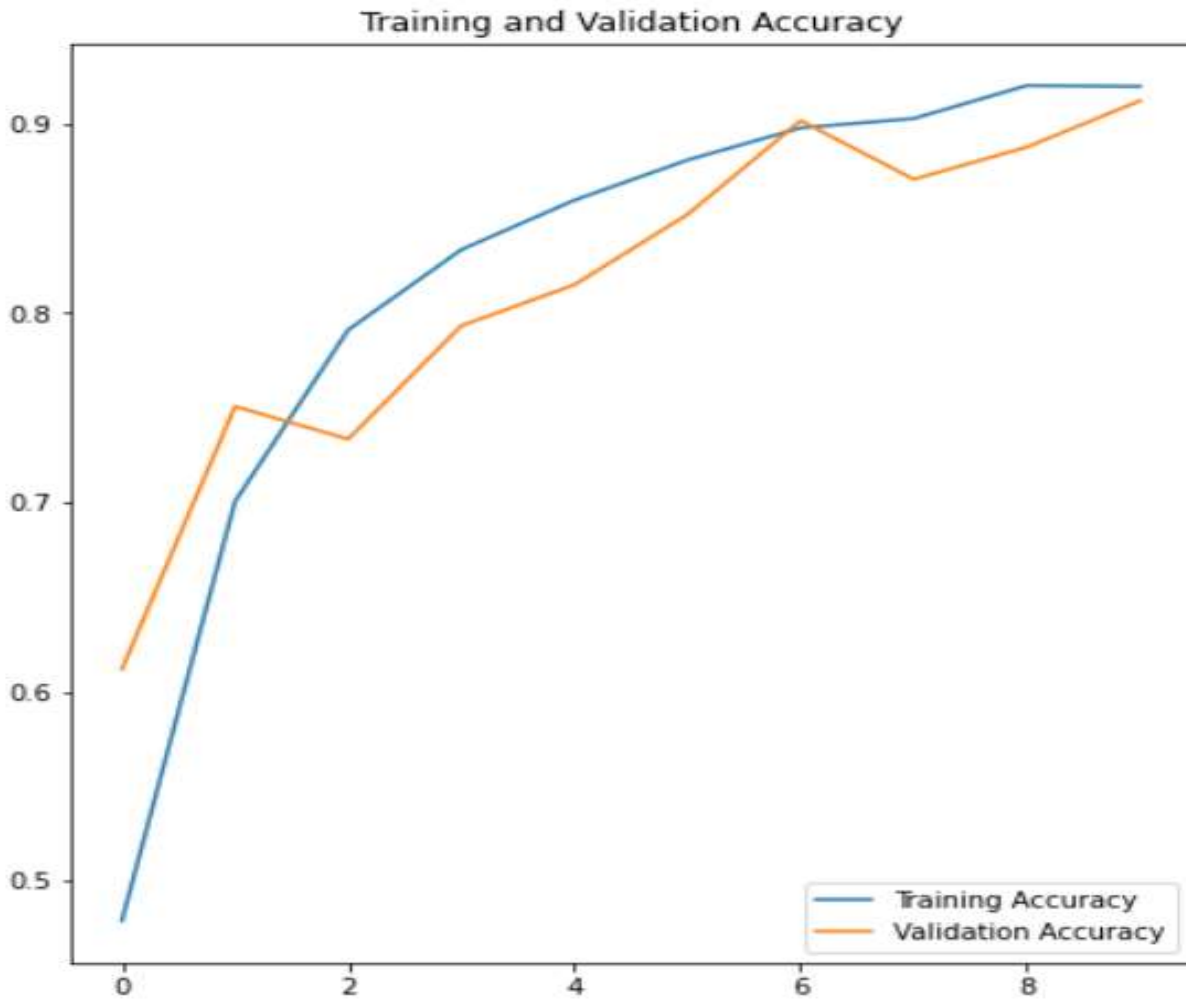
Precision: Precision is defined as the number of true positive results (TP) divided by the number of positive results (TP + FP) that are predicted by the model. The range of the precision is between 0 and 1 and is calculated using Equation (2). It is used to find the proportion of positive identifications that is true.

Recall: The recall is the number of true positives (TP) divided by the number of all relevant sample data (TP + FN). Equation (3) represents the mode of calculation of the recall. It is used to determine the proportion of actual positives that were correctly identified. These concepts are represented mathematically by Equations (2) and (3), respectively:

$$\text{Precision} = \frac{TP}{TP+FP}$$

$$\text{Recall} = \frac{TP}{TP+FN}$$

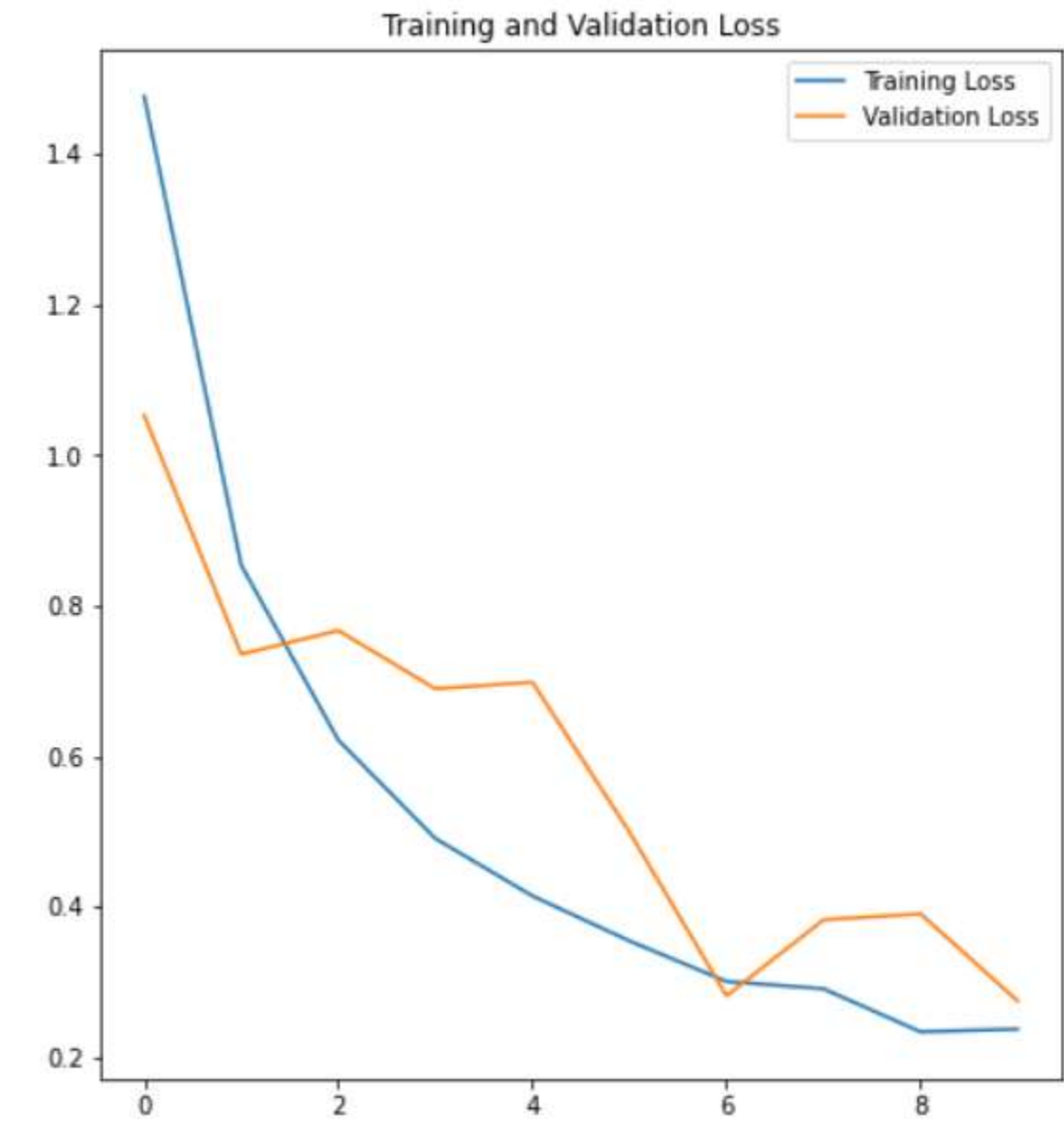
Training And Validation Accuracy



Each model is trained for 200 epochs. The epoch defines the number of times the model will learn on training samples. The training and validation accuracy curve is plotted to visualize the performance of the model during training. Figure 8 represents the accuracy (training and testing) comparison of ResNet50, DenseNet121, DenseNet201, MobileNet, Xception, and proposed model for each epoch.

The X-axis of the accuracy plot represents epoch number (1-200) and Y-axis represents the accuracy values corresponding to each epoch. The loss plot of each model is shown in Figure 9. The X-axis of the loss plot represents the epoch number (1-200) and Y-axis shows the loss value corresponding epochs. When validation loss is much higher than training loss then, the network is considered as over fitting. And if training loss value is much greater than validation loss then, the network is considered as under fitting.

Training And Validation Loss



The performance parameter includes training accuracy, validation accuracy, validation loss, Precision, Recall, and F1-Score. The training accuracy is the number of images correctly classified on which it was trained. And the validation accuracy is the number of unseen images correctly classified. The Precision is proportion of positive identifications was actually correct. Recall is proportion of actual positives was identified correctly. And F1-Score is the weighted average of Precision and Recall. As observed from Table 4, among all pre-trained deep convolutional neural networks, Xception outperformed ResNet50, DenseNet121, DenseNet201, and MobileNet with identification accuracy and loss of 99.60% and 1.73% respectively. But MobileNet models have lesser network parameters among all models. The proposed deep convolutional neural network realizes higher performance in terms of identification accuracy and loss. And also it has lesser parameters among all pre-trained models except DenseNet121 and MobileNet.

Now, the proposed model is analyzed using a re-sampling technique known as k-fold cross-validation. The value k represents number of folds dataset to be split. And also it represents the number of groups. In each group, one fold is used for testing and the remaining k-1 folds involve in training. In this approach, we have used 5-fold cross-validation. The dataset is split into 5-folds. Each folds get a chance to train for 4 times. Each group is trained for 100 epochs. The result obtained from each group and the average result is shown in Table 5. The model achieves average validation accuracy and loss of 99.42% and 0.0220.

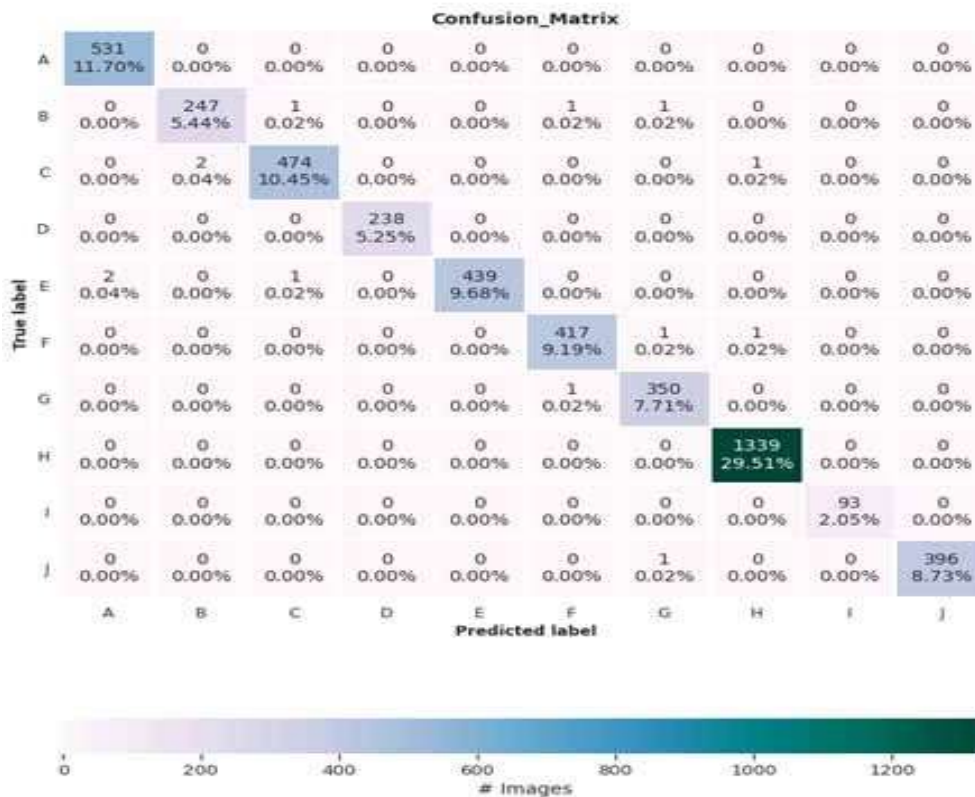
Group No.	Training Accuracy	Validation Accuracy	Validation Loss	Precision	Recall	F1-Score
1.	99.77	99.27	0.0249	0.99	0.99	0.99
2.	99.94	99.38	0.0291	0.99	0.99	0.99
3.	99.86	99.59	0.0132	0.99	0.99	0.99
4.	99.81	99.45	0.0164	0.99	0.99	0.99
5.	99.75	99.42	0.0266	0.99	0.99	0.99
Average:	99.83	99.42	0.0220	0.99	0.99	0.99

Finally, we have also tested the performance of the proposed model on splitting the dataset into the train, validate, and test. The dataset is split in to the train, validation, and test ratio of 70:15:15. The model is trained for 100 epochs. The performance of the proposed model on the given split

Training Accuracy	Validation Accuracy	Test Accuracy	Validation Loss	Test Loss	Precision	Recall	F1-Score
99.74	99.23	99.12	0.0269	0.0315	0.99	0.99	0.99

Confusion Matrix

The confusion matrix is a table in which rows represent the true class labels and columns correspond to the predicted class label. Hence, the confusion matrix helps to visualize the recognition accuracy of the system (Jiang, Peng, et al., 2019). The diagonal elements in the confusion matrix show the number of samples correctly classified and the remaining elements shows the number of samples incorrectly classified in a given set of samples. Figure 10 shows the confusion matrix of the proposed deep convolutional neural network on a given test image samples. The confusion matrix consists of 10 rows and 10 columns. Each row represents the class label of tomato crop diseases. The deeper color represents the higher recognition accuracy of corresponding classes.



Contribution

The major contribution of this paper are as follows

- We proposed light weight deep convolutional neural network for identification of various tomato crop disease. In the proposed deep convolutional neural network number of parameter is minimized by reducing the kernel size and number of neurons. The batch normalization feature is added to overcome from overfitting and to make the training process faster.
 - We presented suitable set of augmentation parameter and its value in order to improve the identification accuracy of tomato crop disease identification model. The parameter and its value is selected by testing on different set of values in given range.
 - The proposed model has been compared with five state-of-the-art deep learning-based models ResNet50, DenseNet121, DenseNet201, MobileNet, and Xception. These models were trained in similar environmental conditions and evaluated on same public benchmark dataset with same training and validation images.
-

FUTURE WORK

1. Data collection: To improve the accuracy of tomato leaf and disease recognition, larger datasets should be collected. This should include images of different diseases, different varieties of tomatoes, and different environmental conditions.
2. Domain adaptation: Domain adaptation techniques can be used to leverage knowledge from other domains to improve the accuracy of tomato leaf and disease recognition.
3. Transfer learning: Transfer learning can be used to leverage knowledge from other tasks to improve the accuracy of tomato leaf and disease recognition.
4. Automated disease diagnosis: Automated disease diagnosis can be achieved using machine learning models. This can reduce the time and effort required for manual diagnosis.
5. Effective treatments: Machine learning models can be used to identify effective treatments for various diseases. This can help to reduce crop losses and improve agricultural productivity.

Conclusion

In conclusion, deep learning approaches have the potential to significantly improve the way tomato plant leaf diseases are recognized and diagnosed. The proposed method of training a CNN on a dataset of tomato leaf images and incorporating the use of remote sensing techniques can provide fast and accurate identification of diseases, even in the early stages, which can lead to better management and control of the diseases. With the continued development of AI technology and deep learning, it is likely that these methods will become increasingly accurate and widely adopted in the future. It is worth noting that this proposed method is still in a research phase and more work need to be done to validate the results and make it useful in practical scenarios.

REFERENCES

- Agarwal M. Gupta S. K. Biswas K. K. (2020). Development of Efficient CNN model for Tomato crop disease identification. *Sustainable Computing: Informatics and Systems*, 28, 100407. 10.1016/j.suscom.2020.100407
- Agarwal M. Singh A. Arjaria S. Sinha A. Gupta S. (2020). ToLeD: Tomato Leaf Disease Detection using Convolution Neural Network. *Procedia Computer Science*, 167, 293–301. 10.1016/j.procs.2020.03.225
- Alarifi J. S. Goyal M. Davison A. K. Dancey D. Khan R. Yap M. H. (2017, July). Facial skin classification using convolutional neural networks. In *International Conference Image Analysis and Recognition* (pp. 479-485). Springer. 10.1007/978-3-319-59876-5_53
- Chollet F. (2017). Xception: Deep learning with depth-wise separable convolutions. *Proceedings of the IEEE conference on computer vision and pattern recognition*. 10.1109/CVPR.2017.195
- Durmuş, H., Güneş, E. O., & Kırıcı, M. (2017). Disease detection on the leaves of the tomato plants by using deep learning. In *2017 6th International Conference on Agro-Geoinformatics*. IEEE. 10.1109/Agro-Geoinformatics.2017.8047016
- Elhassouny A. Smarandache F. (2019). Smart mobile application to recognize tomato leaf diseases using Convolutional Neural Networks. In *2019 International Conference of Computer Science and Renewable Energies (ICCSRE)*. IEEE. 10.1109/ICCSRE.2019.8807737
- Fu L. Lv X. Wu Q. Pei C. (2020). Field Weed Recognition Based on an Improved VGG With Inception Module. *International Journal of Agricultural and Environmental Information Systems*, 11(2), 1–13. 10.4018/IJAEIS.2020040101
- He K. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*.
- Howard, A. G. (2017) *Mobilenets: Efficient convolutional neural networks for mobile vision applications*. arXiv preprint arXiv:1704.861.
- Huang G. (2017). Densely connected convolutional networks. *Proceedings of the IEEE conference on computer vision and pattern recognition*.
- Ileri D. Belal E. Okinda C. Makange N. Ji C. (2019). A computer vision system for defect discrimination and grading in tomatoes using machine learning and image processing. *Artificial Intelligence in Agriculture*, 2, 28–37. 10.1016/j.aiia.2019.06.001
- Islam, M. (2017). Detection of potato diseases using image segmentation and multiclass support vector machine. In *2017 IEEE 30th canadian conference on electrical and computer engineering (CCECE)*. IEEE. 10.1109/CCECE.2017.7946594
- Jiang B. He J. Yang S. Fu H. Li T. Song H. He D. (2019). Fusion of machine vision technology and AlexNet-CNNs deep learning network for the detection of postharvest apple pesticide residues. *Artificial Intelligence in Agriculture*, 1, 1–8. 10.1016/j.aiia.2019.02.001
- Jiang P. Chen Y. Liu B. He D. Liang C. (2019). Real-time detection of apple leaf diseases using deep learning approach based on improved convolutional neural networks. *IEEE Access: Practical Innovations, Open Solutions*, 7, 59069–59080. 10.1109/ACCESS.2019.2914929
-

- Kamal K. C. Yin Z. Wu M. Wu Z. (2019). Depth-wise separable convolution architectures for plant disease classification. *Computers and Electronics in Agriculture*, 165, 104948. 10.1016/j.compag.2019.104948
- Kamilaris A. Prenafeta-Boldú F. X. (2018). Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture*, 147, 70–90. 10.1016/j.compag.2018.02.016
- Kaur S. Pandey S. Goel S. (2019). Plants disease identification and classification through leaf images: A survey. *Archives of Computational Methods in Engineering*, 26(2), 507–530. 10.1007/s11831-018-9255-6
- Koay, K. L. (2020). *Detection of Plant Leaf Diseases using Image Processing* (Diss.). Tunku Abdul Rahman University College.
- Kour V. P. Arora S. (2019). Particle swarm optimization based support vector machine (P-SVM) for the segmentation and classification of plants. *IEEE Access: Practical Innovations, Open Solutions*, 7, 29374–29385. 10.1109/ACCESS.2019.2901900
- Krisnandi D. Pardede H. F. Yuwana R. S. Zilvan V. Heryana A. Fauziah F. Rahadi V. P. (2019). Diseases Classification for Tea Plant Using Concatenated Convolution Neural Network. *Communication and Information Technology Journal*, 13(2).
- Krithika N. Grace Selvarani A. (2017). An individual grape leaf disease identification using leaf skeletons and knn classification. In 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS). IEEE. 10.1109/ICIIECS.2017.8275951
- Krizhevsky A. Sutskever I. Hinton G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*.
- Kumar A. Kim J. Lyndon D. Fulham M. Feng D. (2016). An ensemble of fine-tuned convolutional neural networks for medical image classification. *IEEE Journal of Biomedical and Health Informatics*, 21(1), 31–40. 10.1109/JBHI.2016.263566328114041
- Meng X. Liu M. Wu Q. (2020). Prediction of Rice Yield via Stacked LSTM. *International Journal of Agricultural and Environmental Information Systems*, 11(1), 86–95. 10.4018/IJAEIS.2020010105
- Nema S. Dixit A. (2018). Wheat Leaf Detection and Prevention Using Support Vector Machine. In 2018 International Conference on Circuits and Systems in Digital Enterprise Technology (ICCSDET). IEEE. 10.1109/ICCSDET.2018.8821098
- Padol P. B. Yadav A. A. (2016). SVM classifier based grape leaf disease detection. In 2016 Conference on advances in signal processing (CASP). IEEE. 10.1109/CASP.2016.7746160
- Priyadharshini R. A. (2019). Maize leaf disease classification using deep convolutional neural networks. *Neural Computing & Applications*, 31(12), 8887–8895. 10.1007/s00521-019-04228-3
- Rangarajan A. K. Purushothaman R. Ramesh A. (2018). Tomato crop disease classification using pre-trained deep learning algorithm. *Procedia Computer Science*, 133, 1040–1047. 10.1016/j.procs.2018.07.070
-

