Identification of diseases in tomato leaves using convolutional neural network and transfer learning method

M M F Alim, Subiyanto*, and Sartini
Department of Electrical Engineering, Faculty of Engineering, Universitas Negeri Semarang, Indonesia

*Corresponding author: subiyanto@mail.unnes.ac.id

Abstract. The high market demand for tomatoes required high productivity in the agricultural sector. Plant disease is a threat that obstructs tomato production. Disease control is essential to prevent crop failure. Automatic identification is highly recommended for agriculture applications. Inspired by the recent successes research of deep learning for identification, this study applied a computer vision method for identifying tomato plant diseases. This paper adopted a Convolutional Neural Network (CNN) algorithm with the transfer learning approach to identify tomato plant disease. The CNN models such as VVG, ResNet, and DenseNet have been compared to identify and classify tomato plant diseases. The experiments were carried out using a PlantVillage dataset, with 22930 images of tomato leaves diseases and consists of 10 classes. The best model is achieved by ResNet-50 with accuracy, precision, recall, f1-score, and AUC 96.16%, 97%, 96%, 97%, and 97.92%, respectively. The experimental results proved that CNN models could be a useful tool in identifying tomato plant disease.

1. Introduction
The high market demand for fresh tomato and processed tomato require high tomato productivity [1]. Plant diseases of tomato is a threat that obstructs tomato production and impacts significant crop loss [2]. Bacteria, fungus, and insects cause tomato plant diseases. The traditional plant identification method is based on visual interpretation of leaf’s color and symptoms; these methods require laboratory analysis. This study aims to identify tomato plant diseases automatically with intelligent technology. The nine classes of tomato plant diseases used in this study were bacterial spot, early blight, late blight, leaf mold, septoria leaf spot, spider mites, two spotted-spider mites, target spot and tomato mosaic virus.

Intelligent systems based on computer vision should become part of agricultural production management to increase productivity [3]. Sabrol and Kumar in [5] conducted a study to identify diseases in tomato plants using the Decision Tree algorithm with a dataset of 598 images consisting of 6 classes. The algorithm used can achieve an accuracy of 78%. Hlaing and Maung Zaw in [6] identified tomato plant diseases using the Support Vector Machine (SVM) algorithm. The study used a dataset of 3474 consisting of 6 data classes. The accuracy obtained in this study was 84.7%. Sardogan et al. in [7] also identified tomato plant diseases using a CNN. This study used a dataset of 500 consisting of 4 data classes.

The results show the accuracy of the model is 86%. Hidayatuloh in [8] identifies tomato plant diseases use a CNN algorithm. The study used a dataset of 1400 tomato plant diseases consisting of 7 classes. The results showed an accuracy of 86.92%. De Luna in [9] identified tomato plant diseases using an artificial neural network. The study used a tomato plant disease dataset totaling 4923 consisting
of 4 data classes. The result of this research is that the accuracy value of the model reaches 95.75%. Agarwal in [10] identified tomato plant diseases using a CNN algorithm. This study uses the most amount of data compared to other studies, amounting to 10500 images consisting of 10 data classes. The results of this study are an accuracy value of 91.2%. Artificial intelligent technological developments always present newer methods with a better performance every year. From all existing research, the CNN algorithm can provide an excellent level of accuracy.

CNN with transfer learning needs to be investigated in its application for cases of tomato plant disease. The transfer learning approach allows applying a pre-trained CNN algorithm network and retraining for the new specific task [11]; this reduces network training time and increases performance. This study aims to get the best performance from the CNN model in identifying tomato diseases. The training and testing process used the PlantVillage dataset with 22930 images, consist of 9 diseases class and health. This study uses the CNN model, namely VGGnet [12], ResNet [13], and DenseNet [14], with the transfer learning approach.

2. Methods

This study used a tomato plant diseases named PlantVillage taken from research. This study used 10 classes of the data dataset, including 9 classes of diseases and 1 healthy class. The total number of images is 22930 with a resolution of 256-by-256 pixels. In this study, images were resized 224-by-224 pixels to be equated with each model’s input layer.

CNN was developed from a multi-layer perceptron of a neural network; it is designed to process grid-shaped data. CNN has three main points in its architecture, local connection, shared weight, and spatial subsampling [12]. Shared weights between units at different coordinates tend to look for the same pattern in different parts of the tomato leaves disease images [12]. There are three stages in CNN architecture. First, the convoluting image, apply function by the activation function then compress with pooling layer. Every one process inside convolution is named as a feature map, the way to get the output of each feature map is defined in equation (1).

\[ f(x) = \sum_{i,j=1}^{f} (a_{i,j}, b_{i,j}) \]  

Where \( x \) is the output of feature maps in each operation, and \( f \) is the kernel size. \( i, j = 1 \) represents \( x, y \) coordinates of the processed layer and the value will be increased with 1 value until achieving the f-number. \( a_{i,j} \) represented processed layer and its coordinates, and \( b_{i,j} \) represented kernel or filter.

| CNN Models   | Depth | Total Parameters | Trainable Parameters | Image input size |
|-------------|-------|-----------------|---------------------|-----------------|
| VGG-16      | 16    | 14,719,818      | 5130                | 224, 224        |
| VGG-19      | 19    | 20,029,514      | 5130                | 224, 224        |
| ResNet-50   | 50    | 23,608,202      | 20490               | 224, 224        |
| ResNet-101  | 101   | 42,678,666      | 20490               | 224, 224        |
| DenseNet-121| 121   | 7,047,754       | 10250               | 224, 224        |
| DenseNet-169| 169   | 12,659,530      | 16650               | 224, 224        |

The pooling layer collects and simplifies the image uses various statistical operations. The pooling layer takes maximum or average pixel values from the filtered image and processed them in every feature map. The fully-Connected layer is connected with every neuron in the previous layer. It is applied after transforming the data dimensions into a one-dimensional array. After all, layers have passed, the back-propagation algorithm used in training CNN to minimize the cost function concerning update the weights in every layer.
This study trains CNN models using the transfer learning approach. The ResNet model used a residual learning framework to train the CNN model. Model updates bring residual blocks and deep architecture. VGG networks are one of the widely used CNN models. It has convolution layers and fully-connected layers with ReLU activation. The model provides a significant improvement by pushing the depth to 16–19 layers [13]. Another notable deep CNN model is DenseNet. It exploits network potential by reusing features and fewer parameters compared to other equivalent CNN models. DenseNet consists of blocks named DenseBlocks, where the dimensions of the feature map remain constant within the block, but the number of filters changes between them. And between each block, there is a transition layer. Each model has two variations of depth. There are six CNN models to compare including VGG-16, VGG-19, ResNet-50, ResNet-101, DenseNet-121, and DenseNet-169. The detailed information about the number of trainable parameters of each CNN model is presented in Table 1, and the hyperparameter that is used to train the CNN model is shown in Table 2.

3. Result and Discussion

This research used the transfer learning approach with hyperparameters presented in Table 2. This research used three different CNN model architectures that were evaluated for detecting tomato diseases, such as VGGnet [12], ResNet [13], and DenseNet [14].

![Figure 1](image.png)

**Figure 1.** The relationship of training and validation by the epoch (a) training accuracy (b) training loss (c) validation accuracy (d) validation loss

There are two types of results obtained from experiments, training results and testing. The training results are a graph of the percentage growth in the learning ability of the model in each epoch. It can be observed in Figure 1 that each model has increased training accuracy and decreased training loss, it shows that the models can learn the dataset well. Furthermore, the ResNet-50 model has the best training performance, and the VGG-19 has poor training performance for the tomato plant disease dataset. But despite that ranking, all six models performed well in the training process. Validation accuracy and validation loss show the model's ability to recall some of the data used in training. The validation results show that the ResNet-50 model has the best validation, and VGG-19 is the worst validation model, but
regardless of the ranking, all models show good validation values. Besides, regardless of the training performance of the CNN model, the training time of each model is presented in Figure 2. The DenseNet-121 model requires the fastest training time while the ResNet-101 model requires the longest training time.

The tested model has been evaluated using the confusion matrix method. This method compared the number of true and false predictions, then the result was used to determine the accuracy, precision, recall, f1-score, and AUC of each model. The summary of the performance metrics of each model can be seen in Table 3. It could be observed that ResNet-50 has a better performance in detecting tomato diseases. the accuracy, precision, recall, f1-score, and AUC were 96.16%, 97%, 96%, 97%, and 97.92, respectively. And the VGG-19 got the last rank which the accuracy, precision, recall, f1-score, and AUC were 82.42%, 90%, 82%, 86%, and 90.63 respectively. The comparisons are visualized in the bar chart in Figure 2. Based on this diagram, a ranking of the CNN model can be made from the highest to the lowest. The ResNet-50 and ResNet-101 models have the best average on all the scoring criteria, followed by DenseNet-169, and DenseNet121, and finally VGG-16 and VGG-19.

Table 3. The performance of CNN models

| CNN Models  | Accuracy | Precision | Recall | F1-Score | AUC   |
|-------------|----------|-----------|--------|----------|-------|
| VGG-16      | 84.54    | 91.00     | 85.00  | 88.00    | 91.85 |
| VGG-19      | 82.42    | 90.00     | 82.00  | 86.00    | 90.63 |
| **ResNet-50** | **96.16** | **97.00** | **96.00** | **97.00** | **97.92** |
| ResNet-101  | 95.77    | 96.00     | 96.00  | 96.00    | 97.70 |
| DenseNet-121| 88.48    | 94.00     | 88.00  | 91.00    | 93.92 |
| DenseNet-169| 92.91    | 96.00     | 93.00  | 94.00    | 96.24 |

Figure 2. Bar chart of performance metrics for each CNN model

The CNN model with the best performance was then compared with the results of previous studies. The comparisons are presented in Table 4. In Table 4, nine other studies also examined the identification of tomato plant diseases using machine learning. Research by [15] obtained an accuracy value of 76.11% using the GA-FFNN method to identify tomato plant diseases. And the most recent study by [10] used the CNN algorithm to identify tomato plant diseases and got an accuracy of 91.2%. The data used is a dataset of 10500 images consisting of 10 classes. Meanwhile, in this research, the CNN algorithm is also used with a transfer learning approach to identify tomato plant diseases. From the most
recent research, this analysis provides additional data used to get better performance of the model, with an accuracy of 96.16%. That value is a significant improvement when compared to other works.

| No | Author(s) | Size of dataset | Number of Classes | Classification method | Accuracy (%) |
|----|-----------|-----------------|-------------------|-----------------------|--------------|
| 1  | (Muthukannan and Latha, 2018) [15] | 470 | 6 | GA FFNN | 76.11 |
| 2  | (Sabrol and Kumar, 2016) [16] | 598 | 6 | Decision Tree | 78.00 |
| 3  | (Hlaing and Zaw, 2018) [17] | 3474 | 6 | SVM | 84.70 |
| 4  | (Hlaing and Maung Zaw, 2018) [6] | 3535 | 7 | Multi-Class SVM | 85.10 |
| 5  | (Sardogan, Tuncer and Ozen, 2018) [7] | 500 | 4 | CNN | 86.00 |
| 6  | (Hidayatuloh, Nursalman and Nugraha, 2018) [8] | 1400 | 7 | CNN | 86.92 |
| 7  | (Agarwal et al., 2020) [10] | 10500 | 10 | CNN | 91.20 |
| 8  | (Aziz et al., 2019) [18] | 1882 | 5 | SVM | 94.00 |
| 9  | (De Luna, Dadios and Bandala, 2019) [9] | 4923 | 4 | ANN | 95.75 |
| 10 | This study | 22.973 | 10 | CNN | 96.16 |

4. Conclusion
The CNN models were applied in this paper to recognize tomato plant diseases automatically with the transfer learning approach. The transfer learning approach keeps and freezes the pre-trained network, and reconfigures the network with some additional layers to classify tomato plant disease. With the transfer learning approach, the best model achieved good results on the identification of tomato plant diseases. The best network of this study has 23.6 million parameters with 20.4 thousand trainable parameters. The experimental results indicate that the ResNet-50 was able to recognize the tomato diseases with an accuracy of 96.16%. Finally, it can be concluded that the CNN model has a good capability in its application to the identification of tomato plant diseases.

Acknowledgments
The authors would like to thank the UEESRG for the valuable suggestions and encouragement which have helped in improving the quality of this paper.

References
[1] Adhikari P, Oh Y and Panthee D R 2017 Int. J. Mol. Sci. 18
[2] Ali M M, Bachik N A, Muhadi N Atirah, Tuan Yusof T N and Gomes C 2019 Physiol. Mol. Plant Pathol. 108 101426
[3] Tian H, Wang T, Liu Y, Qiao X and Li Y 2019 Inf. Process. Agric. 7 1
[4] Hameed K, Chai D and Rassau A 2018 Image Vis. Comput. 80 24
[5] Sabrol H and Satish K 2016 Int. Conf. Commun. Signal Process. (ICCSP 2016) p 1242
[6] Hlaing C S and Maung Zaw S M 2018 Proc. - 17th IEEE/ACIS Int. Conf. Comput. Inf. Sci. (ICIS 2018) p 439
[7] Sardogan M, Tuncer A and Ozen Y 2018 UBMK 2018 - 3rd Int. Conf. Comput. Sci. Eng. p 382
[8] Hidayatuloh A, Nursalman M and Nugraha E 2018 Int. Conf. Inf. Technol. Syst. Innov. (ICITSI 2018) p 199
[9] De Luna R G, Dadios E P and Bandala A A 2019 IEEE Reg, 10 Annu, Int, Conf, Proc. (TENCON 2018) p 1414
[10] Agarwal M, Singh A, Arjaria S, Sinha A and Gupta S 2020 Procedia Comput. Sci. 167 293
[11] Lee S H, Goëau H, Bonnet P and Joly A 2020 Comput. Electron. Agric. 170 105220
[12] Simonyan K and Zisserman A 2015 3rd Int. Conf. Learn. Represent. (ICLR 2015) p 1
[13] He K, Zhang X, Ren S and Sun J 2016 Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. p 770
[14] Tong T, Li G, Liu X and Gao Q 2017 Proc. IEEE Int. Conf. Comput. Vis. p 4809
[15] Muthukannan K and Latha P 2018 Multimed. Tools Appl. 77 24387
[16] Sabrol H and Kumar S 2016 Int. J. Comput. Sci. Inf. Secur. 14 622
[17] Hlaing C S and Zaw S M M 2018 Parallel Distrib. Comput. Appl. Technol. PDCAT Proc. p 223
[18] Aziz S, Bashir M, Mughal O, Khan M U and Khan A 2019 2019 IEEE 10th Annu. Inf. Technol. Electron. Mob. Commun. Conf. (IEMCON 2019) p 973