2018 International Conference on Information Technology Systems and Innovation (ICITSI)
Bandung - Padang. October 22-25, 2018
ISBN: 978-1-5386-5692-1

Identification of Tomato Plant Diseases by Leaf Image Using Squeezenet Model

Akbar Hidayatuloh[1], M. Nursalman[2], Eki Nugraha[3]
Program Studi Ilmu Komputer
Departemen Pendidikan Ilmu Komputer
Universitas Pendidikan Indonesia
Bandung, Indonesia

{mnursalman, ekinugraha}@upi.edu; akbar0102@student.upi.edu;

***Abstract*— One of the problems in the field of agriculture is regarding plant diseases. Plant diseases can cause a decrease in agricultural production. Therefore, early detection and diagnosis of plant diseases is very important. Plant diseases often appear on the leaves, and the characteristics of the affected leaves can be varied and difficult to distinguish. This makes it difficult to identify the disease automatically. Increased smartphone usage and advances in the field of computer vision through deep learning have made it possible to connect smartphones as a tool in diagnosing diseases. Plants used as case studies in this study were tomato plants. The method used is Convolutional Neural Network (CNN). This CNN method has many types based on the architecture it builds, one of which is squeezenet architecture. Squeezenet architecture can produce a model with a relatively small size, so that the model can be implemented on smartphone devices, server computing, and microcontroller devices. This research will focus on building a model based on squeezenet architecture to classify seven types of tomato plant diseases on the leaves including healthy leaves. The model is built using the help of Keras deep learning frameworks. The data used in this study is the image of tomato plant leaves obtained from the Vegetable Crops Research Institute (Balitsa) in Lembang, West Java. With 200 classes for each class. This study has successfully detected tomato plant disease through its leaf image automatically with an average accuracy of identification of 86.92%.**

Keywords*— *tomato plant diseases, convolutional neural network, squeezenet architecture, deep learning, computer vision, balitsa, keras

I. INTRODUCTION

Indonesia, which is an agricultural country and has abundant wealth in the fields of plantations, agriculture and vast land area. As a country with two seasons, Indonesia's potential as a producer of superior horticultural products is very high. The coverage of horticultural commodities used in the form of horticultural agriculture statistics includes 90 commodities, consisting of 26 seasonal vegetables and fruit plants, 25 annual types of fruit and vegetable plants, 15 types of biopharmaceutical plants and 24 types of ornamental plants [1].

Horticultural plants have been widely cultivated in Indonesia, but if seen from the results are still not satisfactory, this is caused by sharing factors, some of which are cultivation techniques, environmental conditions and disturbances of pests and diseases. Pests and diseases are a major obstacle that can reduce production by up to 40% [2].

To help overcome the problems regarding pests and diseases in plants, many methods have been used, one of which is the CNN method, which is included in the deep learning method. In recent years, deep learning methods won several contests in pattern recognition and machine learning. This deep learning method includes artificial neural networks. In the use of CNN, several architectures have been made such as AlexNet [3], ZFNet [4], VGG [5], etc. However, all of these architectures are more focused on improving their accuracy, it takes a little architecture with parameters to produce a model with a small size, to later be applied to a server to do computing, or planted in a embedded system that has limited resources.

Squeezenet architecture applies three main strategies in the formation of its structure so that it can provide good accuracy and minimize the number of parameters. The advantages of squeezenet architecture that is a small architecture requires a little bandwidth to export new models on the cloud, small architecture is also easier to deploy on FPGA devices and other hardware devices that have memory limitations [6]. Therefore, in this study the CNN method with squeezenet architecture will be used to identify plant diseases through their leaves.

II. RELATED WORKS

Image processing is widely used in the field of agriculture whether the image is taken through a remote sensor device or directly on the field. Image processing and artificial neural networks are used by [7] to detect symptoms in cucumber plants, use the SVM classification to identify two types of tomato plant diseases [8]. The disadvantage of using traditional methods is that you have to determine the features that will be used to detect the disease, the process of searching for features that match the data requires a long time [9]. Lately, deep learning methods are also used to detect plant diseases through images, [9] in his study comparing two CNN architectures to detect diseases from 14 plants using leaf images from an open repository resulting in an accuracy of 99.35% in a held-out test set, the drawbacks in this study were data used with lab conditions, so that when tried for field data produced accuracy is only 31.40%.

Research by [10] using transfer learning techniques with Inception v3 model to train a small amount of data in order to get good accuracy results to detect disease in cassava plants through its leaves with 93% accuracy, then implemented into an android application, the drawback is the large application size because of the Inception model the complex architecture has a model size of around 84MB.

Detection of tomato plant disease was carried out by [11] using the VGG16 architecture which obtained an accuracy of 89% and research by [12] by using the Alexnet and Squezetnet architecture whose purpose is to be implemented on a robot or through photos by sensors installed in a greenhouse, the accuracy obtained is 94%. The disadvantage of the previous research is that the data used is data with lab conditions, so that it can provide good results.

Although there have been many studies on the detection of plant diseases, but using data with lab conditions and the resulting model is large. This research focuses on using data directly from agricultural fields, producing small-sized models that can be used on android devices, server computing and FPGA devices as well as creating systems that can be used as initial diagnoses of disease-affected plants.

III. Convolution Neural Network (CNN)

Convolutional neural network is a method of deep learning designed to recognize visual patterns directly from image pixels by minimizing preprocessing. CNN can recognize patterns with a variety of variations contained in an image, resistant to distortion and simple geometry transformations [13].

Initially CNN has been widely used for object detection problems, but now it has been widely used in other domains such as object tracking, pose estimation, text recognition and detection, visual detection of saliency, detection of actions, labeling of sights, etc. [14].

Many CNN architectures have been put forward, the main purpose of the formation of various architectures is to improve the accuracy that can be obtained, while in the future deep learning will be implemented on mobile devices such as smartphones, cameras or FPGAs. In this study itself aims to detect plant diseases through a mobile device or computer, therefore it takes an architecture that can provide good accuracy as well as a relatively small model size. For this purpose squeezenet architecture is chosen.

Squeezenet is a form of convolutional neural network architecture, proposed by [6]. Squeezenet is a good form of network architecture engineering. To reduce the size of the model, squeezenet was built with three strategy designs namely, reducing filter size, reducing input channels and downsampling at the end of the network. Squeezenet uses a fire module, which consists of a squeeze layer (with 1x1 filter to decrease the input channel from 3x3) and expand the layer (a combination of 1x1 and 3x3 filters to reduce filter size), squeeze layer and expand layer followed by the ReLu activation layer [12].

Fire modules on squeezenet architecture consist of two layers, squeeze layer and expand layer, both of which are the main keys of squeezenet architecture.

1) Squeeze layer is a layer composed of three convolution layers with each size 1x1.

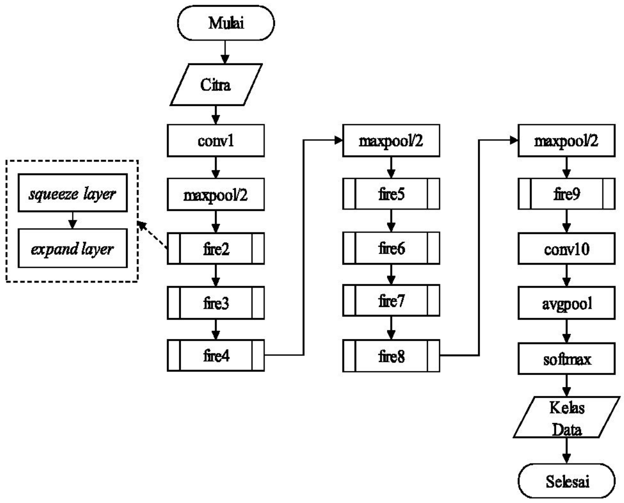

Figure 1 Squeezenet architecture

2) Expand layer is a layer composed of a combination of four 1x1 convolution layers and four 3x3 convolution layers.

When through a fire module, the squeeze layer will help limit the number of input channels to the expand layer. So the number of parameters that are input to the expand layer decreases and makes the model formed by a small squeezenet architecture.

IV. Methods

The identification process of tomato plants began with the process of collecting tomato leaf datas obtained from the Vegetable Crop Research Institute (Balitsa). Then the entire dataset will go through preprocessing consisting of scaling and normalization. By using the CNN method which is squeezenet architecture, then the training dataset process is carried out to produce the model.

A. Dataset

The dataset used is image data of tomato leaves derived from plasma fields and greenhouses taken from the Vegetable Crops Research Institute (Balitsa) Lembang, Jawa Bara. Image data is photographed with the condition of one leaf for one image and exposed to direct sunlight. There are seven classes of image of tomato leaves including healthy leaf class, the number of image data that was obtained was 1400 images.

List of names of tomato plant diseases used as many as seven classes including healthy leaf class can be seen in Table 1.

TABLE 1 DISEASES LIST

Variable	Class Name
C0	Early Blight
C1	Late Blight
C2	Healthy
C3	Phosphorus Deficiency
C4	Calcium Deficiency
C5	Magnesium Deficiency
C6	Tomato Leaf Miner

Figure 2 Sample images

B. *Image Preprocessing*

Each image data will go through the preprocessing stage, this preprocessing phase includes scaling and normalization.

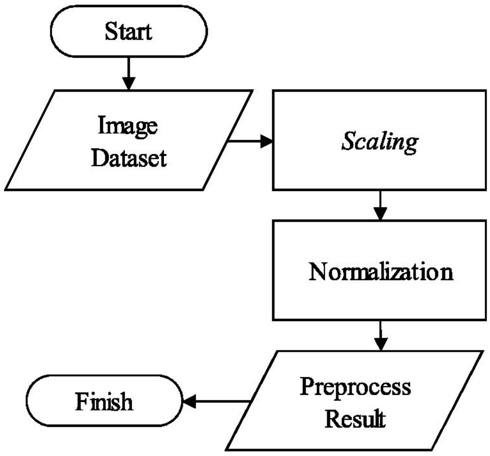

Figure 3 Preprocessing flow

In the preprocessing stage, a scaling process is performed to change the image size to 224 x 224 and the normalization process, which is the process of changing the image value whose origin has a value between [0.255] to a value of 0 to 1.

C. *Model Development*

After the image dataset goes through the preprocessing stage, the CNN model will be developed with squeezenet architecture. Model development is done to later be carried out the image dataset training process. The training process itself is a process to extract patterns or features in the image and characterize it using CNN so that it can identify tomato plant diseases.

1) *Training Process*

The training process begins by defining the initial weight of the network filter to be input into the hidden layer. This stage consists of two processes, namely feed-forward and backpropagation.

2) *Testing Process*

The testing process is a classification process using the weights and biases obtained in the training process. The testing process is not much different from the training process, but in testing it is not done backpropagation after the feedforward process is carried out. The result of this testing process is the accuracy of the classification process.

D. *Model Evaluation*

To measure the performance of the CNN model with squeezenet architecture in identifying tomato plant diseases from the leaves, cross validation process will be used using the k-fold method with a value of $k = 5$.

Cross validation will divide the data into two parts, the first part for training and the second part for validating the model. For the data used in the training process is k-1 fold and the remaining 1 fold will be used as validation data, this process will repeat until all folds become validation data [15]. To measure the effectiveness of the classification, confusion matrix will be used at each epoch.

TABLE 2 CROSS VALIDATION

1	*fold 1*	*fold 2*	*fold 3*	*fold 4*	*fold 5*
2	*fold 1*	*fold 2*	*fold 3*	*fold 4*	*fold 5*
3	*fold 1*	*fold 2*	*fold 3*	*fold 4*	*fold 5*
4	*fold 1*	*fold 2*	*fold 3*	*fold 4*	*fold 5*
5	*fold 1*	*fold 2*	*fold 3*	*fold 4*	*fold 5*

Then the calculation will be done to calculate the accuracy of each fold with the equation (1):

$$Akurasi = \frac{Jumlah\ data\ benar}{Total\ data} x\ 100\% \qquad (1)$$

After calculating the accuracy value of each fold, then the average is calculated from k = 5 for each epoch. From each epoch, confusion matrix will be calculated from each class to see the effectiveness of the CNN method used.

V. RESULTS

To evaluate the performance of the model's ability to predict new data that is not through the previous training process, experiments were conducted using the cross-validation method of k-fold with a value of k = 5. Data is divided into five parts (fold) for training and testing. Each fold is 280 images. With the amount of class data is 40 images.

The parameters used to build the model in the training process are as follows:

- Number of epoch: 100, 200, 300, 400, dan 500
- Optimizer (learning rate) : Adam (lr: 0.0001)
- Loss function: categorical crossentropy

The following is a table of model accuracy using cross validation method k-fold cross validation, with a value of k = 5. From each fold, the accuracy of each epoch will be calculated to determine the ability of the model built in classifying new data that has not been seen during the training process, after getting the accuracy value of each fold, the average accuracy for each epoch will be calculated, which will show the ability of the model when tried to classify new data that has not been seen or when testing through field data later.

TABLE 3 EVALUATION RESULTS CROSS VALIDATION

Fold	*Epoch*				
	100	200	300	400	500
1	64.29%	77,50%	82,14%	87,50%	85.71%
2	74.29%	79,29%	85%	84,64%	88.21%
3	56.43%	77,50%	81,43%	86,07%	87.86%
4	71.43%	77,86%	82,50%	86,43%	85.71%
5	68.21%	81,79%	82,50%	82,86%	87.14%
Average	66.93%	78,78%	82,71%	85,50%	86.92%

The results obtained in Table 3 have the lowest accuracy at 100 epoch at the 3rd fold. Meanwhile, the highest accuracy is at the 500th epoch at the 2nd fold. Confusion matrix for epoch 100 can be seen in Table 4 and confusion matrix for epoch 500 can be seen in Table 8.

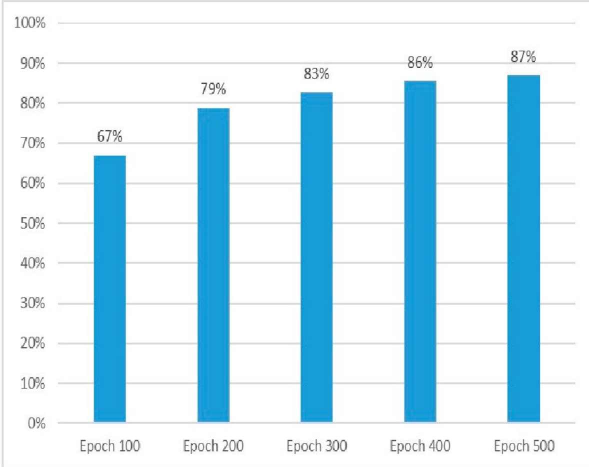

Figure 4 Average accuracy every epoch

The accuracy value obtained by the model in predicting validation data with the k-fold cross validation method using a configuration of epoch value ranges from 100-500 has increased in each of its epochs, with the highest accuracy value at the 500 epoch which is 87%.

Figure 5 Average accuracy every class

The average accuracy of each data class at each epoch in the range 100-500 has increased, with the highest accuracy values in each class at the epoch 500. As in early blight class the accuracy increases are directly proportional to the increase in the number of epochs. Likewise, the class of leaf miner has increased significantly from epoch 100 to 500. Compared to the class of calcium deficiencies in each epoch it looks stable. Models have good ability to get certain features for calcium deficiency classes.

TABLE 4 CONFUSION MATRIX EPOCH 100

Actual	Predicted							Accuracy
	C0	C1	C2	C3	C4	C5	C6	
C0	134	2	5	5	7	22	25	67%
C1	3	118	49	11	0	1	18	59%
C2	4	17	162	0	1	0	16	81%
C3	24	10	1	114	13	33	5	57%
C4	2	0	0	6	188	3	1	94%
C5	28	2	0	12	2	150	6	75%
C6	69	13	37	4	3	3	71	35%

The results obtained in Table 4 have the highest accuracy in class C4, namely calcium deficiency of 94% while the lowest accuracy is in class C6, namely leafminer fly pests by 35%. Class C6 or leafminer fly pests get the smallest accuracy because the model cannot classify well, because when viewed from the data used in this study, the image of leaves affected by leaf slitting pests is also exposed to early blight disease or class C0 so that the confusion matrix is wrong predictions that should be identified as leaf slaughterflies, but early blight disease was identified.

TABLE 5 CONFUSION MATRIX EPOCH 200

Actual	Predicted							Accuracy
	C0	C1	C2	C3	C4	C5	C6	
C0	139	0	6	0	4	14	37	70%
C1	0	168	25	0	0	0	7	84%
C2	0	9	172	0	1	0	18	86%
C3	6	6	0	171	1	15	1	86%
C4	6	0	0	11	177	6	0	89%
C5	12	1	0	12	2	165	8	83%
C6	35	10	34	4	1	5	111	56%

The results obtained in Table 5 are the highest accuracy also found in class C4, namely the calcium deficit class with an accuracy value of 89%. While the smallest accuracy on epoch 200 is class C6 or the class of leaf miner fly with an accuracy value of 56%. This number is slightly increased from the previous epoch.

TABLE 6 CONFUSION MATRIX EPOCH 300

Actual	Predicted							Accuracy
	C0	C1	C2	C3	C4	C5	C6	
C0	142	0	4	0	4	18	32	71%
C1	0	172	14	3	0	0	11	86%
C2	0	8	171	1	0	0	20	86%
C3	1	3	0	187	1	7	1	94%
C4	4	0	0	5	183	7	1	92%
C5	9	1	0	5	3	174	8	87%
C6	36	3	21	3	2	6	129	65%

In Table 6 the results of accuracy with the highest value is in class C3 or class of phosphorus deficiency that is equal to 94%. While the smallest accuracy at epoch 300 is found in class C6 or leaf miner fly which is equal to 65%.

TABLE 7 CONFUSION MATRIX EPOCH 400

Actual	Predicted							Accuracy
	C0	C1	C2	C3	C4	C5	C6	
C0	159	1	5	1	2	10	22	80%
C1	0	178	15	1	0	0	6	89%
C2	0	7	183	0	1	0	9	92%
C3	4	7	0	181	2	6	0	91%
C4	8	0	0	5	180	5	2	90%
C5	13	1	0	4	2	173	7	87%
C6	22	4	25	3	1	2	143	72%

In Table 7 the results of the accuracy with the highest value is in class C2 or healthy leaf class that is equal to 92%. Whereas the smallest accuracy on epoch 400 is found in class C6 or leaf miner fly that is equal to 72%.

TABLE 8 CONFUSION MATRIX EPOCH 500

Actual	Predicted							Akurasi
	C0	C1	C2	C3	C4	C5	C6	
C0	156	0	5	1	5	7	26	78%
C1	1	175	13	1	0	0	5	87%
C2	2	3	179	1	1	0	14	89%
C3	3	2	0	187	1	6	1	93%
C4	6	0	1	4	183	3	3	91%
C5	8	1	0	7	2	168	14	84%
C6	16	4	10	2	1	3	164	82%

The results obtained in Table 5 are the highest accuracy in the C3 class, namely healthy leaves with accuray 93% while the lowest accuracy is in class C0 is early blight with accuray 78%.

There is an increase in the accuracy of each class at the epoch 500 of the previous epoch. By increasing the number of epochs, the accuracy of the model will also increase, but the computational time of the training model will be longer.

The neural network convolution model using squeezenet architecture that is built can distinguish each class. By detecting every particular feature that these classes have. One feature that is obtained is the color feature, so the model can detect either class C4 or calcium deficiency and class C3 or healthy leaves. Because both classes have more dominant color characteristics than other classes and other characteristics. For classes C0 and C6 or early blight or leaf miner, both are classes with little accuracy among other classes. This can be caused by the condition of the image data used. Some data on C6 are also affected by C0 disease so that when the model predicts there is often a wrong prediction between the two classes.

The size of the model that is produced using squeezenet architecture is relatively small at 8.844KB, which is smaller compared to other models, so that it can be implemented for mobile devices, computational prediction through FPGA servers or embedded systems.

VI. Conclusion

It can be concluded that the convolutional neural networks model with squeezenet architecture can be used to identify six types of diseases on the leaves of tomato plants and their healthy leaves. With an average level of accuracy using the k-fold cross validation method of 86.92% and it can be seen that the squeezenet model that is built is a good choice to be implemented on a mobile device or to make computational predictions via the server to be used as REST API because of its relatively small size and resource requirements to run low model computing.

For further research, it can be done by increasing the number of epochs or adjusting the configuration parameters used and by increasing the amount of data used to improve data quality better, trying to identify other horticultural plant diseases, either through their leaves or other organs, and by designing software that can detect if there are new diseases that have never been through the previous model training process.

References

[1] Kementerian Pertanian Direktorat Jenderal Hortikultura, "Statistik Produksi Hortikultura Tahun 2014," *Stat. Produksi Hortik. Tahun 2014*, p. 286, 2014.

[2] G. W. Sasmito, "Sistem Pakar Diagnosis Hama dan Penyakit Tanaman Hortikultura dengan Teknik Inferensi Forward dan Backward Chaining," *J. Teknol. dan Sist. Komput.*, vol. 5, no. April, pp. 69–74, 2017.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Adv. Neural Inf. Process. Syst.*, pp. 1–9, 2012.

[4] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8689 LNCS, no. PART 1, pp. 818–833, 2014.

[5] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," pp. 1–14, 2014.

[6] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size," vol. 9349, pp. 1–13, Feb.2016. [7] M. P. Pawar, D. V. Turkar, and P. P. Patil, "Cucumber Disease Detection using Artificial Neural Network," *Int. Conf. Inven. Comput. Technol.*, pp. 391–395, 2016.

[8] U. Mokhtar, M. A. S. Alit, A. E. Hassenian, and H. Hefny, "Tomato leaves diseases detection approach based on support vector machines," *2015 11th Int. Comput. Eng. Conf.*, pp. 246–250, 2015.

[9] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using Deep Learning for Image-Based Plant Disease Detection," *Front. Plant Sci.*, vol. 7, Sep. 2016.

[10] A. Ramcharan, K. Baranowski, P. McCloskey, B. Ahmed, J. Legg, and D. P. Hughes, "Deep Learning for Image-Based Cassava Disease Detection," *Front. Plant Sci.*, vol. 8, no. October, pp. 1–7, Oct. 2017.

[11] J. Shijie, J. Peiyi, H. Siping, and Sl. Haibo, "Automatic detection of tomato diseases and pests based on leaf images," in *2017 Chinese Automation Congress (CAC)*, 2017, pp. 2537–2510.

[12] H. Durmus, E. O. Gunes, and M. Kirci, "Disease detection on the leaves of the tomato plants by using deep learning," in *2017 6th International Conference on Agro-Geoinformatics*, 2017, pp. 1–5.

[13] M. Setiawan, "Klasifikasi Penyakit pada Citra Daun Menggunakan Convolutional Neural Networks," 2018.

[14] N. Aloysius and G. M, "A Review on Deep Convolutional Neural Networks," *Int. Conf. Commun. Signal Process.*, pp. 588–592, 2017.

[15] P. Refaeilzadeh, L. Tang, and H. Liu, "Cross-Validation," in *Encyclopedia of Database Systems*, L. Liu and M. T. Özsu, Eds. New York, NY: Springer New York, 2016, pp. 1–7.