INTERNATIONAL JOURNAL OF ADVANCED STUDIES IN COMPUTER SCIENCE AND ENGINEERING
IJASCSE VOLUME 9 ISSUE 6, 2020

06/30/2020

Implementation of Deep Learning Methods in Detecting Disease on Chili Leaf

Rosalina
Faculty of Computing, President University
Bekasi, Indonesia
rosalina@president.ac.id

Genta Sahuri
Faculty of Computing, President University
Bekasi, Indonesia
rosalina@president.ac.id

***Abstract*— Chili is one of the important horticultural plants in Indonesia. The price of chili has drastically increased in the market due to the scarcity of chili plants. The scarcity is caused by erratic weather changes, which resulted in many chili plantations experiencing crop failure, due to diseases that attack chili plants so that yields are reduced. This research will implement Deep Learning for image processing into disease detection systems. This disease detection system will be used to help users, especially chilli farmers to detect whether the leaves of their chili plants are infected by the disease or not. This system will take a picture of chili leaves using a Raspberry Pi camera and image processing on the chili leaf image to obtain important information about the image to find out whether the chili leaves are infected by disease or not. The purpose of this study is to create a desktop application for a disease detection system that has the ability to detect whether or not a chili leaf is infected by several diseases, display the condition of the chili leaves, display the type of disease that infects the chili leaves (if any), and provide a percentage probability of the system in detecting images of the chili leaves correctly (whether it is healthy chili leaves or diseased chili leaves).**

Keywords-component; Chili leaf disease; deep learning; image processing; raspberry

I. INTRODUCTION

Chili is one of the most valuable plant species in Indonesia. Chili is categorized as vegetables and fruits plant with protein to be produced and has great economic value [1]. The diseases that attacks the chili plants could cause crop failure, thus, proper control is needed to avoid it. Appropriate control is not only carried out when an attack has occurred, but the most important thing is to prevent them. The latency in the disease diagnosis is due to the lack of awareness among farmers about kinds of diseases which harm chili plants [2]. Currently, to detect disease in chili plants is done manually by specialists. In practice, a team of experts and continuous monitoring is needed, so it requires very high costs when it is done on extensive agriculture. At the same time, farmers in Indonesia generally do not have adequate facilities to be able to contact experts, but also limited expertise becomes an obstacle for farmers to consult. Based on the description described above, we need an expert system that can help chili farmers to detect whether their chili plants are infected by disease or not.

Several studies have been carried out to diagnose diseases in plants using leaf image data. These studies use leaf image data because the disease is evident in the leaves. A study published by Aduwo [3] presenting it's use computer vision to make a diagnosis cassava. They were using pictures of the leaf surface of the cassava plants captured in the laboratory with unified lighting and background. A two-class system has been proposed to identify whether leaves pictures come from infected or healthy plants. Three extracted features had been retrieved from leaf images, along with hue and image intensity (HSV) features, and attributes which detect keystrokes on pictures, which include Scale Invariant Feature Transformation (SIFT) and Speeded-Up Robust Features (SURF) features. Then the classification method were improved by [4 - 6] by implementing the prototype-based classification scheme. Several other studies extend the problem of two classes into multi-class problems with various diseases and different disease rates [7]. Some other features were also used in the extension of the early work by Aduwo et.al.

A newer image-based approach to chili disease detection is based on deep learning methods, for example as written in [8], a convolutional neural network model was developed to detect and diagnose plant diseases using simple leaf images of healthy and diseased plants, through a deep methodology learning. Several architectural models are trained, with the best performance achieving a 99.53% success rate in identifying the appropriate combination (plant, disease) (or healthy plant). The very high success rate makes this model a very useful advisory or early warning tool, and an approach that can be further developed to support integrated plant disease identification systems to operate in real cultivation conditions. Other research that uses deep learning method is [9,10] where this research uses the approach of drawing leaves captured by the user and then the images are processed to determine the health status of plant leaves. Detection of chili leaf disease through leaf drawing and data processing techniques are very useful and inexpensive systems especially to assist farmers in monitoring large crop areas.

This research will implement Deep Learning for image processing into disease detection systems. This disease detection system will be used to help users, especially chili farmers to detect whether the leaves of their chili plants are infected by the disease or not. This system will take a picture of chili leaves using a Raspberry Pi camera and perform image processing on the chili leaf image to obtain important information about the image to find out whether the chili leaves are infected by disease or not.

The purpose of this study is to create a desktop application for a disease detection system that has the ability to detect whether or not a chili leaf is infected by several diseases, display the condition of the chili leaves, display the type of disease that infects the chili leaves (if any), and provide a percentage probability of the system in detecting images of the chili leaves correctly (whether it is healthy chili leaves or diseased chili leaves).

II. Related Works

A. Disease Detection System on Probolinggo Onion Leaves using Template Matching based Raspberry Pi

This system can diagnose the onion leaf disease experienced by the red onion plant. This system uses Logitech Camera C270 as a data collector to collect data on onion leaf disease and Raspberry Pi as a microcomputer. This system only uses the Template Matching method which is image processing algorithm instead of using object detection algorithm and image processing algorithm to recognize one of the diseases such as purple spot disease, feather dew disease, and moler disease [11].

B. Plant Leaf Disease Detection using SVM Classifier and K-Means Segmentation

This system can detect and classify the paddy leaf disease using SVM (Support Vector Machine) Classifier. K-Means Clustering is used for segmentation of images. This method provides a solution to the early recognition of diseases. By using this method, the farmers can automatically identify the leaf diseases at initial stages. There are three types of diseases that can be detected by this system which are Bacterial Blight, Myrothecium, and Alternaria [12]. Listed in Table 1 is the comparison overview between the features of this system and those of related work.

Table 1. Comparison Table

No.	Program Features	This System	Disease Detection System on Probolinggo Onion Leaves Using Template Matching Based Raspberry Pi	Plant Leaf Disease Detection using SVM Classifier and K-Means Segmentation
1	Capture an Image	Yes	Yes	No
2	Detect Healthy Leaf	Yes	No	No
3	Detect Sick Leaf	Yes	Yes	Yes
4	Display Type of Disease	Yes (1 type of disease)	Yes (3 types of diseases)	Yes (3 types of diseases)
5	Display Probability Percentage Value of The System in Detecting Chili Leaf Image Correctly	Yes	No	No

III. System Overview

This research implement Deep Learning Framework which is TensorFlow for image processing into disease detection system. This disease detection system will be used to help the users especially chili farmers to detect whether the leaf of their chili plants are infected by the disease or not. This system will take a picture of chili leaf by using Raspberry Pi camera and do the image processing on the chili leaf image to get the important information about that image in order to know whether the chili leaf is infected by the disease or not. The objective of this research is to create a desktop application for disease detection system that has the abilities to detect whether the chili leaf is infected by some diseases or not, display the condition of the chili leaf, display the type of disease that infect the chili leaf (if any), and give the probability percentage value of the system in detecting the image of chili leaf correctly (whether it is healthy chili leaf or sick chili leaf).

A. Data

The datasets for this system consists of 1000 images which are 500 images for healthy chili leaf and 500 images for sick chili leaf. From 500 images for healthy chili leaf and sick chili leaf, 100 images from healthy chili leaf and sick chili leaf will be used for testing data, while the remaining will be used for training data. The datasets for this system are shown in Fig.1 and Fig.2.

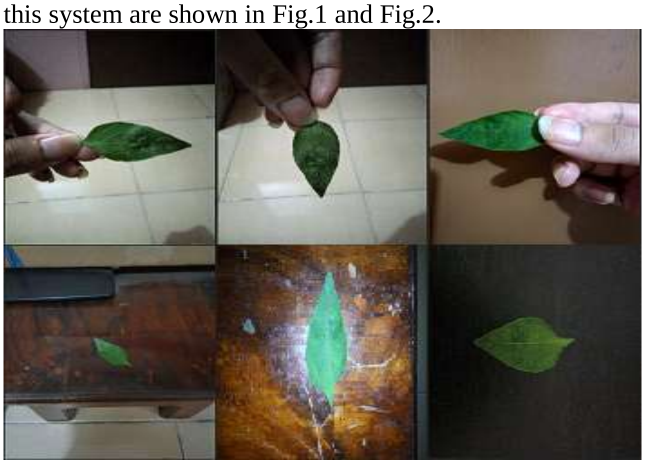

Figure 1. Datasets of Healthy Chili Leaf

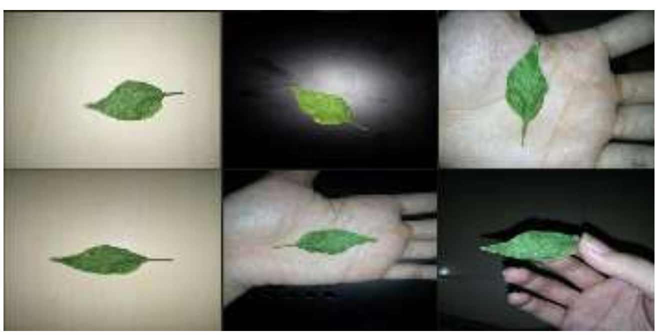

Figure 2. Datasets of Sick Chili Leaf

Figure 1. and Figure 2 shows the example datasets of healthy chili leaf image and sick chili leaf image (purple spot). This datasets were used for training data and testing data in training and evaluating the model.

B. Training Model

In training model, firstly, the system will load all the training images. Next, the system will convert all the training images into grayscale image and resize it. After that, the label will be added for each training image. There are two labels in this case which are the label to indicate the healthy chili leaf and the label to indicate the sick chili leaf (purple spot). Next, the training image data with its own label will be stored to the list. And then, the list will be shuffled. After that, the system will load all the testing images. Then, the system will convert all the testing images into grayscale, resizing the images to 128x128, converting the images into pixel array, and adding the list. Next, the label also will be added for each testing image. There are also two labels in this case which are the label to indicate the healthy chili leaf and the label to indicate sick chili leaf (purple spot). After that, the testing image data with its own label will be stored to the list.

Next, the training image data will be normalized, reshaped, and stored to a variable. Then, the label for training images also will be stored to the variable. After that, the testing image data will be normalized, reshaped, and stored to a variable. Next, the label for testing images also will be stored to a variable. And then, the system will initialize the model, add the layers (first layer, second layer, and fully connected layer), and add relu and sigmoid activation. The first layer and second layer consist of convolution 2D layer along with the activation function and max-pooling 2D layer. While fully connected layer consists of dropout layer, flatten layer, and dense layer along with the activation function. After that, the system will do the data augmentation process on the training images, compile the model by using the loss function “binary_crossentropy”, optimizer “Adam Optimizer” with the learning rate = 0.001, and metrics “accuracy”, and train the model based on the training datasets.

After that, the training data and the testing data will be normalized, reshaped into 1-D array with the size of 128 x 128 pixel, and stored in different variable. After that, the label of training data and the label of testing data will also be stored in different variable. After that, the system will create model by using that data.After the system prepared all the needed data for training and testing. Next, the system will initialize the model, add the layers on the model (consists of three layers which are first layer, second layer, and fully connected layer), do data augmentation process on the training images, compile the model, and train the model. After that, the system will count the number of file with extension .ckpt, since the model will be saved in the format .ckpt. Then, the system will generate the model path which is the location where the model will be saved and the model name. Next, the model is saved to generated model path with the model name based on the generated model name. Lastly, the system generates the text file and the model path is saved to the generated text file.

C. Neural Network Model

At first, the input image with the shape of 128 x 128 x 1 is fed to the first layer and second layer which consist of Convolution 2-D layer and Max-Pooling 2-D layer. In the first Convolution 2-D layer, the input image will be filtered by using the filter size of 3 x 3, number of filters = 16, padding = 1, and stride = 1. The output shape from this layer will be 128 x 128 x 16. Next, this output will be fed to a Max-Pooling 2-D layer in order to reduce the dimensional size, the computation, and overfitting with the filter size of 2 x 2, padding = 1, and stride = 2. The output shape from this layer will be 64 x 64 x 16. Then, this output will be fed to second Convolution 2-D layer with the filter size of 3 x 3, number of filters = 32, padding = 1, and stride = 1. And the output shape from this layer will be 64 x 64 x 32. After that, this output will be fed again to a Max-Pooling 2-D layer to reduce the dimensional size, the computation, and overfitting again with the filter size of 2 x 2, padding = 1, and stride = 2. The output shape from this layer will be 32 x 32 x 32. Next, 25% of this output will be dropped out and this output will be flatten from 2-D array to 1-D array. This process will produce 32768 neurons. Then, two fully connected layers are added in the end. The first fully connected layer contains 256 neurons, while the second fully connected layer only contains 2 neurons which represent the probability of an image being a healthy chili leaf or sick chili leaf. Figure 3 shows the neural network mode of chili leaf disease detection system.

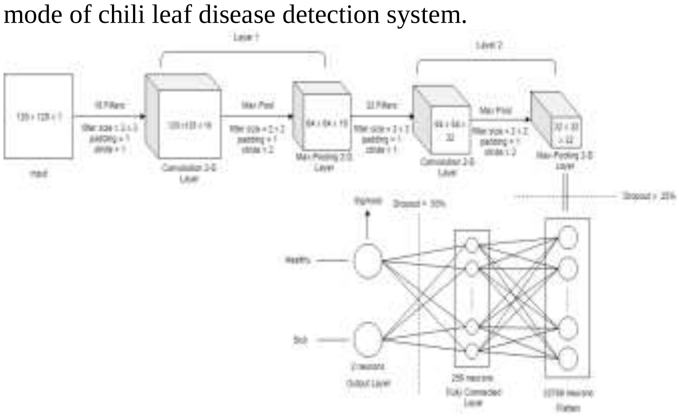

Figure 3. Neural Network Model of The System

IV. Chili Leaf Disease Detection System

The user interface of Disease Detection System on Chili Leaf only consists of single page. This page is quite simple and easily understandable for the user. The page contains two canvases and four buttons. The two canvases are canvas for input image and canvas for output image. While the four buttons are button for starting camera, capture image, train model, and process image. At first, "Capture Image" button will be disabled. This button will be enabled when the Raspberry Pi camera is active and accessible. The Raspberry Pi camera is said to be accessible and active if the live camera preview is displayed on the screen. To make the Raspberry Pi camera becomes accessible and active, the "Starting Camera" button must be clicked. The next button is "Train Model". This button can be clicked anytime without depending on the other button. The last button is "Process Image" button. The action on this button will be triggered if the user already captured an image and the trained model is available. If this condition is not fulfill, the system will give a notification to the user which inform that the user must capture an image and train the model first. The interface of this system is shown in Figure 4.

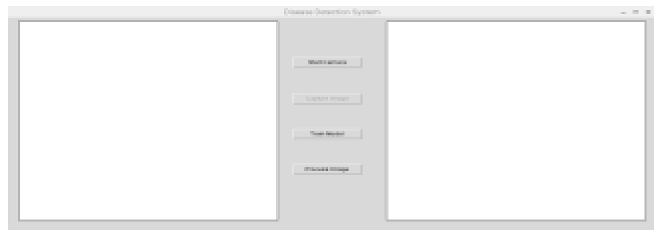

Figure 4. User Interface

Figure 3. shows the user interface of the system when running the system for the first time. At first, there are two canvases and four buttons. One of the buttons is disabled. That button is "Capture Image" button. This button will enabled again when Raspberry Pi camera is accessible and active.

Figure 5. Displaying Image

Figure 5 shows the captured image is displayed on the canvas for input image. When the user clicked on "Capture Image" button, first the captured image will be resized into 575 x 450 pixel. After that, the captured image is displayed on the canvas for input image.

Figure 6. Training Model

Figure 6. shows the training process of a model. When the user clicked on "Train Model" button, the system will automatically train the model. The training process will be

displayed on the python shell along with estimation time, accuracy, loss, validation loss, and validation accuracy.

Figure 7. Displaying Result (Healthy Chili Leaf)

Figure 8. Displaying Result (Sick Chili Leaf)

Figure 7 and Figure 8 show the result of processing an image. When the user clicked on "Process Image" button, the system will start to process the image by predicting the captured image whether it is healthy chili leaf or sick chili leaf based on the trained model. When the process is done, the result image with the descriptions about the condition of the chili leaf, type of disease that infect the chili leaf (if any), and probability percentage value of the system in detecting correctly the captured image (whether it is healthy chili leaf or sick chili leaf) will be displayed on the canvas for output image.

A. *System Accuracy*

The training data consists of 400 images of healthy chili leaf and sick chili leaf and 100 images of healthy chili leaf and sick chili leaf were used for testing data. The precision of the data gathered upon training a model utilizing TensorFlow shown in Table 2.

Based on the testing result, it is found out that in order to make the system works properly, there are three limitations in the testing process. The first limitation is regarding to the lighting. In order to get the decent and clear picture, the Raspberry Pi camera should be placed in the environment with sufficient lighting. Since this condition will make the system to be able to predict the captured image accurately. The second limitation is regarding to the distance. The distance between the Raspberry Pi camera and chili leaf should not be more than 1 meter in order to make the system to be able to predict the captured image accurately. Because if the distance exceeds 1 meter, the size of the chili leaf on the image will become smaller and the system will not be able to predict the captured image accurately. The last limitation is regarding to the object. The object that is used for testing must be chili leaf. Since the system can only detect the object of chili leaf in order to get the result accurately. If the object is not chili leaf, the system must not be able to get the result accurately.

Tabel 2. Accuracy Result

	Accuracy
Main Functionality (Sufficient Lighting and Distance < 1 meter)	100%
Main Functionality (Sufficient Lighting and Distance >= 1 meter)	72,0%
Main Functionality (Less Lighting and Distance < 1 meter)	85,2%
Main Functionality (Less Lighting and Distance >= 1 meter)	68,8%

V. CONCLUSION

The chili leaf disease detection system is successfully to start Raspberry Pi camera, capture an image using Raspberry Pi camera, create the model, train the model on the training datasets, evaluate the model on the testing datasets, and process as well as recognize the chili leaf image in order to detect whether the chili leaf is infected by purple spot disease or not. In addition, the system also able to detect the chili leaf image in order to know whether the chili leaf is infected by purple spot disease or not but with some conditions which are the environment should have a sufficient lighting and the distance between chili leaf and Raspberry Pi camera should not be more than 1 meter.

REFERENCES

[1] Ralahalu, M. S., Hehanusa, M. L., and Oszaer, L.L, "Respon Tanaman Cabai Besar (Capsicum annum L) Terhadap Pemberian Pupuk Organik HormonTanaman Ungul", Jurnal Ilmu Budaya Tanaman, vol. 2, no.2, Oktober 2013

[2] Muslim, 2015. Sistem Pakar Diagnosa Hama Dan Penyakit Cabai Berbasis Teorema Bayes. Vol. 4, No. 3, Desember 2015 : 797 –876.

[3] J. R. Aduwo, E. Mwebaze, and J. A. Quinn, "Automated vision-based diagnosis of cassava mosaic disease," Industrial Conference on Data Mining - Workshops, pp. 114–122, 2010

[4] E. Mwebaze and M. Biehl, "Prototype-based classification for image analysis and its application to crop disease diagnosis," Advances in Self-Organizing Maps and Learning Vector Quantization - Proceed-ings of the 11th International Workshop WSOM 2016, pp. 329–339,January 2016

[5] E. Mwebaze, P. Schneider, F.-M. Schleif, J. Aduwo, J. Quinn,S. Haase, T. Villmann, and M. Biehl, "Divergence-based

classificationin learning vector quantization,"Neurocomputing, vol. 74, no. 9, pp.1429 – 1435, 2011

[6] E. Mwebaze, M. Biehl, G. Bearda, and D. Zuehlke, "Combining dissimilarity measures for prototype based classification,"European Symposium on Artificial Neural Networks (ESANN), vol. 23, pp. 31–36, 2015

[7] G. Owomugisha and E. Mwebaze, "Machine learning for plant disease incidence and severity measurements from leaf images,"15th IEEE International Conference on Machine Learning and Applications(ICMLA), pp. 158–163, 2016.

[8] Konstantinos P. Ferentinos, "Deep learning models for plant disease detection and diagnosis", Computers and Electronics in Agriculture 145, 311–318, 2018

[9] M. Karuna, B. S Varsha, Sneha. R. M, G. K. Meghana, "Early Detection of Chili Plant Leaf Diseases using Machine Learning", IJESC, Vol.9 No.5, 2019

[10] Mohammed Brahimi, Marko Arsenovic, Sohaib Laraba,Srdjan Sladojevic, Kamel Boukhalfa and Abdelouhab Moussaoui, "Deep Learning for Plant Diseases: Detection and Saliency Map Visualization" , J. Zhou and F. Chen (eds.),Human and Machine Learning, Human–Computer Interaction Series, Springer International Publishing AG, 2018

[11] Zamroni, M., Fitriyah, H., & Maulana, R. Sistem Pendeteksi Penyakit Daun Bawang Merah Probolinggo. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer , vol. 2(12), pp. 6026-6031, 2018

[12] Devi, T. G., Neelamegam, P., & Srinivasan, A. Plant Leaf Disease Detection using K means Segmentation. International Journal of Pure and Applied Mathematics, vol. 119(15), pp. 3477-3483, 2018.