Evolution in Electrical and Electronic Engineering Vol. 4 No. 1 (2023) 572-578

EEEE

Homepage: http://publisher.uthm.edu.my/periodicals/index.php/eeee
e-ISSN : 2756-8458

Development of Web-based Chili Plant Diseases Identification System

Nurul Ainaa Juhari[1], Nuramin Fitri Aminuddin[1], Hafiz Haziq Osman[1], Munirah Ab Rahman[1,2], Zarina Tukiran[1,3]*

[1]Faculty of Electrical and Electronic Engineering,
Universiti Tun Hussein Onn Malaysia, Johor, 86400, MALAYSIA

[2]VLSI and Embedded System Technology (VEST) Focus Group,
Universiti Tun Hussein Onn Malaysia, Johor, 86400, MALAYSIA

[3]Internet of Things (IoT) Focus Group,
Universiti Tun Hussein Onn Malaysia, Johor, 86400, MALAYSIA

*Corresponding Author Designation

DOI: https://doi.org/10.30880/eeee.2023.04.01.069
Received 30 January 2023; Accepted 10 April 2023; Available online 30 April 2023

Abstract: Chili is one of high economic vegetables. However, chili diseases are considered one of the main factors influencing its production. To minimize losses in production, it is essential to develop a tool to assist the farmers identify the disease. Therefore, in this work, the development of web-based chili plant diseases identification system is developed. This system uses Convolutional Neural Network (CNN) model to identify healthy and diseased (Bacterial Spot and Whiteflies) chili based on its leaf image. All chili images datasets were obtained from Kaggle and it contains 1000 images per class. Besides that, this study also investigates the accuracy performance of the model based on number of images dataset (250 and 1000 per class) for epochs of 50. The findings show that the CNN model with 50 epochs and 1000 images per class gives better accuracy performance on the system. By using these findings, a local host server is developed using ReactJS user interface with CNN-based identification system as a core. The aim of the proposed solution is able to assist the farmers to monitor the health of their plant.

Keywords: Chili leaf, Chili Diseases, Identification System, User Interface

1. Introduction

The Solanaceae family includes chili, sometimes referred to as Capsicum annum L. and Capsicum Frustescens L. [1]. Chili originates in Mexico and South America but was long ago adapted to the tropical climate of Malaysia. Chili is susceptible to a variety of pathogens, particularly viruses. In comparison to other plants, the chili plant is reported to be more prone to disease [2], which can result in significant production losses. Infectious disease agents, seedling diseases, leaf diseases, fungi,

**Corresponding author: zarin@uthm.edu.my*

bacteria, abiotic disorders, and plant viruses are some of the possible causes of the illness. the plant's ability to produce food in both quantity and quality will be impacted. By administering early diagnosis and therapy, the damaged region may be diminished or returned to a healthier state. The knowledgeable plant pathology can make the diagnosis and prescribe a course of action. However, [3] It will take a long time to determine whether a plant is quality or not when compared to a manual method. With deep learning, it is possible to create a system for early detection of damaged chilies, protecting both growers and customers. For an understanding of the state of the chili, chili plants must be watched over. The chili plant's leaf is the primary subject of the study.

In identifying diseases of plant, previous works show the use of computer vision-based system [4]-[7] give promising result in assisting the farmer. Therefore, the aim of the paper is to apply the deep learning method for disease detection of chili plant. The deep learning that focused on is Convolutional Neural Network (CNN) and it is suitable to identify image and make a classification of image into few classes. Next, to analyze the accuracy performance of the identification system based on the learning curve of the deep learning system. The analysis will show the training and validation accuracy and it will interpret the performance of the system model. Last but not least, to design and develop a graphical user interface (GUI) for interaction between user and the deep learning system that has been successfully trained.

2. Methods

In this work, the process of proposed system is divided into two (2) main parts namely the development of identification system using CNN and the development of graphical user interface for web application.

2.1 The development of identification system using CNN

Figure 1 shows the block diagram mainly highlighted the input image, image processing, the Convolutional Neural Network that involve feature extraction, training, testing and validation and prediction.

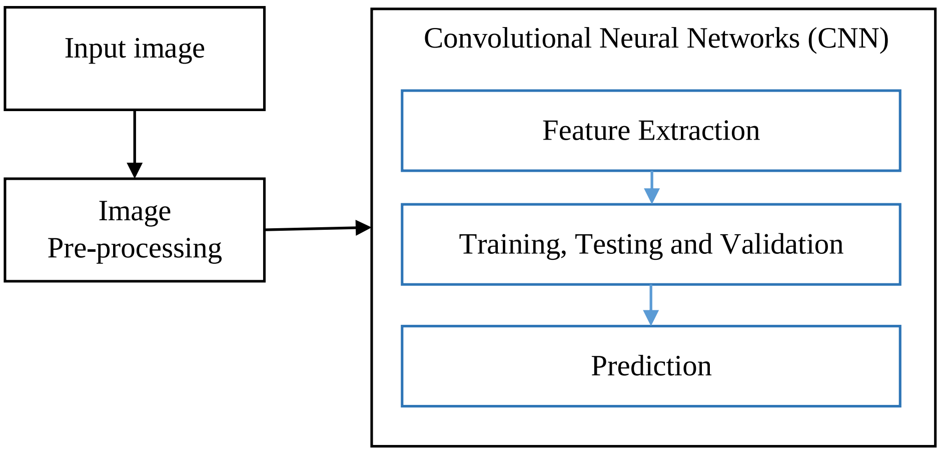


Figure 1: The block diagram of proposed identification system

The input image was taken from public dataset namely Kaggle. This work focuses on three (3) categories of chili leaf; healthy, whiteflies and bacterial spot as shown in Figure 2.

Figure 2: Images of chili leaf; (a) healthy, (b) whiteflies and (c) bacterial spot

Each category is provided with 1000 dataset which make 3000 images in total. Since the size of images varies for each category, pre-processing of the images were conducted by resizing the image into 256x256. Then, the CNN is designed with 12 layers that consist of convolution, and pooling layers to perform the feature extraction. To train, test and validate the CNN model, the dataset was feed to the model with distribution of 80%, 10%, 10% for training, testing and validation, respectively.

2.2 The development of graphical user interface for web application

The development of graphical user interface (GUI) for the web involves back-end server and front-end server. The back-end server is where the model was deployed using the Fast Application Programming Interfaces (FastAPIs). The FastAPIs are used to make applications accessible to users and other developers for use and integration. The API serves as the app's gateway to allows it to interact and consume the API through HTTP requests. In this work, the back-end is configured using PyCharm software.

The front-end server ran the frontend using Visual Studio Code and Node.js. React.js is an interface-building JavaScript library used for Web-based system development. The front-end framework of the web development stack is React.js. On the initial request for a single-page application, it is done by loading a single HTML document. After that, the necessary section, content, or body of the webpage is changed using JavaScript. This approach is known as client-side routing because it allows the client to obtain a new page without having to reload the entire website every time a user makes a new request. React intercepts the request and only fetches and updates the necessary regions rather than requiring a complete page reload. This method leads to improved performance and a more dynamic user experience. Based on Chrome's JavaScript runtime, Node. js is a framework for building rapid and scalable network applications. Node. js is a lightweight and effective platform that works well with distributed devices to run data-intensive real-time applications. It employs a non-blocking, event-driven I/O architecture. For VS Code, it is a desktop-based source code editor that is small but effective. Figure 3 shows the UI component that is available in the web based such as Avatar, Container, Card, Drop zone area and what not. All of the code for the front-end was made in Visual Studio code.

Figure 3 shows the working principle of the proposed GUI of the identification system. The GUI is designed to allow the user to insert their plant photo in two (2) methods either using drag and drop method or by selecting their photo in the folder. Once the photo is uploaded, the identification based on the selected photo is performed. Then the result window shows the identified categories either the plant photo is healthy or having bacteria spot or whiteflies disease. The output also shows the percentage confidence of the identification to show accurate probability of correctness for any of its prediction. In this work, the confidence percentage result is in between 0% to 100% which the higher confidence percentage indicates the likelihood correctness of prediction is also high.

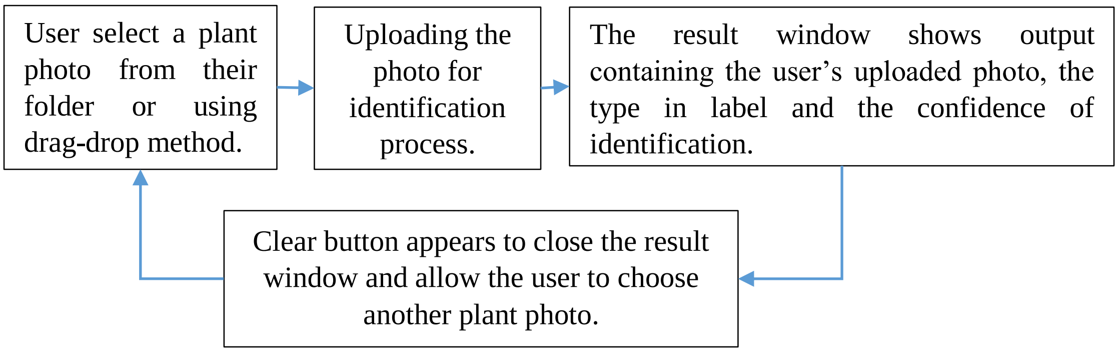


Figure 3: The working principle of the proposed GUI of the identification system

3. Results and Discussion

This section is divided into two (2) main findings; the CNN-based identification system and the GUI for web application.

3.1 The CNN-based identification result and discussion

The CNN-based model is trained using Jupyter Notebook. In this training, testing and validation, the model uses two (2) datasets (250 and 1000 dataset for each class) for the same epoch, 50. Figure 4(a) shows the graph of training and validation of 50 epochs for 250 datasets for each class in CNN model. This graph is compared with 1000 datasets for each class for the same epoch, 50 as shown in Figure 4(b).

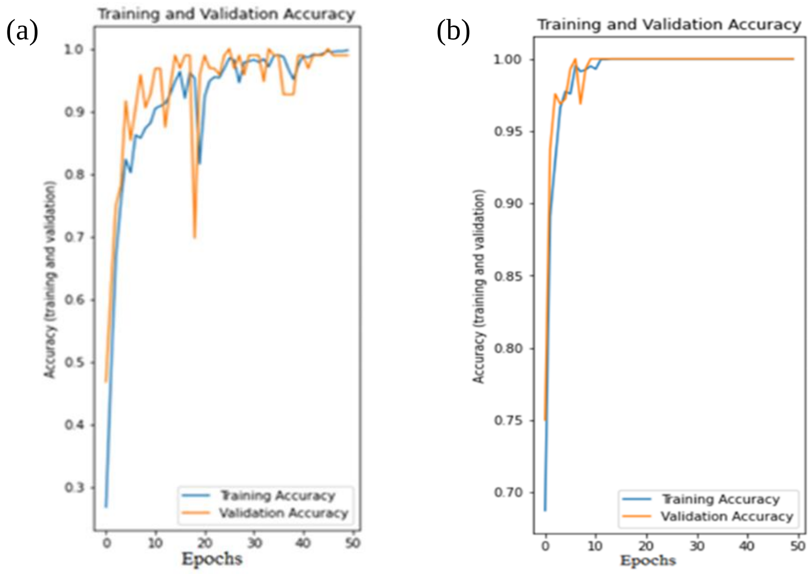


Figure 4: Findings for 50 epochs on (a) 250 and (b) 1000 datasets per class

The finding in Figure 4(a) shows unrepresentative validation data set where it does not provide sufficient information to evaluate the ability of the model to generalize. This may occur if the validation dataset has too few examples as compared to training data set. Also, it shows a validation is higher than the training will resulted in overfitting model [20]. In Figure 4(b), it shows a good fit learning curve as it increases to a point of stability with a minimal gap between training and validation data set accuracy. Based on the findings, the CNN-based identification for web application is designed to use 50 epochs and a total of 3000 images dataset.

3.2 The GUI for web application result and discussion

Figure 5 shows the GUI of the web application that runs in localhost. The GUI for web application is designed and developed in two (2) versions; the GUI for mobile phone.

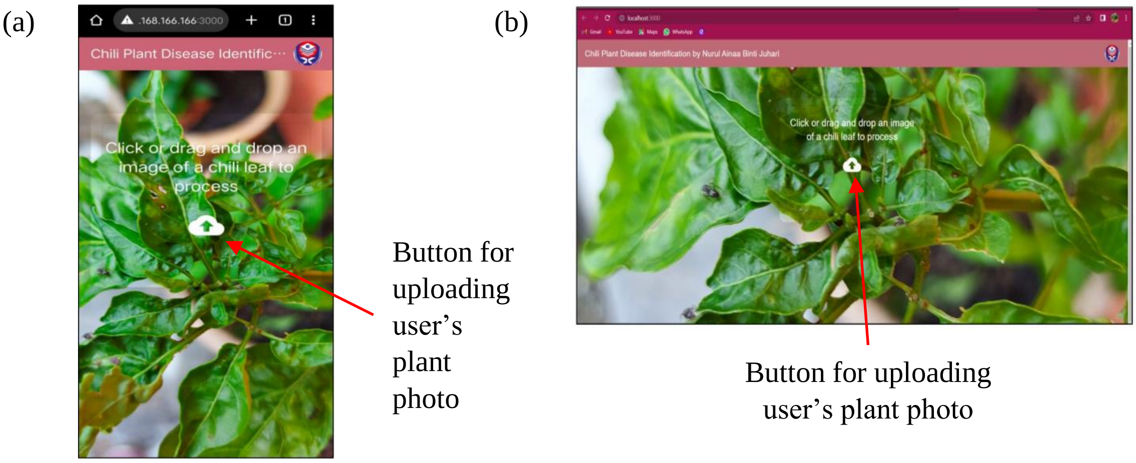


Figure 5: GUI on (a) the mobile phone and (b) desktop

Figure 6 shows the GUI with the result window and clear button. The result window shows the identification result with label of healthy/disease and confidence percentage. Whilst the clear button is programmed to close the result window and allow the user to choose another plant photo.

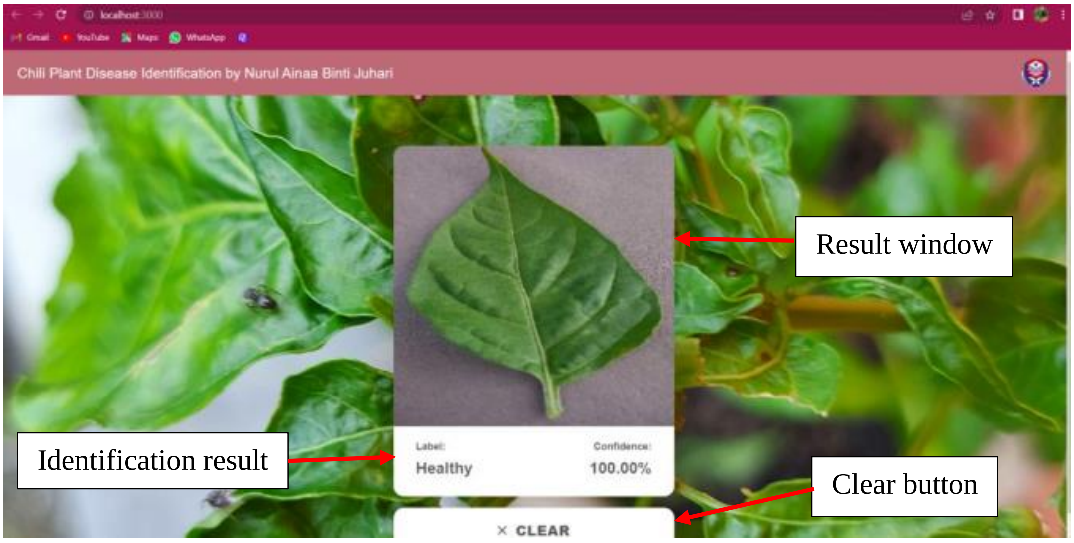
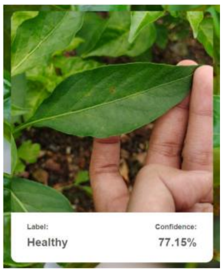


Figure 6: The GUI after the identification is performed

This GUI for web application is then tested with varies of chili leaf plant photo. These photos were taken by the author from a small chili farm located nearby Kluang, Johor. The results are shown in Table 1.

Table 1: Result of GUI on desktop version

<table>
<tr><th>Type of leaf</th><th></th><th>Photo taken by the author</th><th>Identification result in Result window</th></tr>
<tr><td>Healthy</td><td>(a)</td><td></td><td>
</td></tr>
<tr><td></td><td>(b)</td><td></td><td>
</td></tr>
<tr><td>Bacterial spot</td><td>(c)</td><td></td><td>
</td></tr>
<tr><td></td><td>(d)</td><td></td><td>
</td></tr>
<tr><td>Whiteflies</td><td>(e)</td><td></td><td>
</td></tr>
<tr><td></td><td>(f)</td><td></td><td>
</td></tr>
</table>

4. Conclusion

In conclusion, the proposed web-based chili plant disease identification system is designed and developed in two (2) version; mobile phone and desktop layout. In this system, the identification is performed by using CNN model. This proposed solution is hope able to assist the farmers to monitor the health of their plant through the use of technology.

Acknowledgement

This research was supported by Universiti Tun Hussein Onn Malaysia (UTHM) through Tier 1 (vot H916). Special appreciation to the IoT Laboratory, UTHM, for providing related facilities. The authors would also like to thank the Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia for its support.

References

[1] “Capsicum Annum – Wikipedia”, En. En.wikipedia.org, 2022. [Online]. Available: https://en.wikipedia.org/wiki/Capsicum_annuum.Strunk, W., Jr., & White, E. B. (1979). The elements of style (3rd ed.). New York: MacMillan.

[2] P. Lujan and N. Goldberg, Chile pepper diseases, 5th ed. New Mexico: College of Agricultural, Consumer and Environmental Sciences, 2004, p. 27.

[3] H. Jindal, N. Sardana and R. Mehta, "Analyzing Performance of Deep Learning Techniques for Web Navigation Prediction", Procedia Computer Science, vol. 167, pp. 1739-1748, 2020. Available: 10.1016/j.procs.2020.03.384.

[4] A. R. Bahtiar, Pranowo, A. J. Santoso and J. Juhariah, "Deep Learning Detected Nutrient Deficiency in Chili Plant," 2020 8th International Conference on Information and Communication Technology (ICoICT), 2020, pp. 1-4, doi: 10.1109/ICoICT49345.2020.9166224.

[5] T. Purwaningsih, I. A. Anjani and P. B. Utami, "Convolutional Neural Networks Implementation for Chili Classification," 2018 International Symposium on Advanced Intelligent Informatics (SAIN), 2018, pp. 190-194, doi: 10.1109/SAIN.2018.8673373

[6] Sudianto, Y. Herdiyeni, A. Haristu and M. Hardhienata, "Chilli Quality Classification using Deep Learning," 2020 International Conference on Computer Science and Its Application in Agriculture (ICOSICA), 2020, pp. 1-5, doi: 10.1109/ICOSICA49951.2020.9243176.

[7] Y. Sari, A. R. Baskara and R. Wahyuni, "Classification of Chili Leaf Disease Using the Gray Level Co-occurrence Matrix (GLCM) and the Support Vector Machine (SVM) Methods," 2021 Sixth International Conference on Informatics and Computing (ICIC), 2021, pp. 1-4, doi: 10.1109/ICIC54025.2021.9632920.