Android Application of Leaf Identification System

https://doi.org/10.3991/ijim.v16i15.31881

Mohammad Hazwan Hakim Minhat[1], Sharifah Saon[2(✉)],
Abd Kadir Mahamad[2], Umi Fadlillah[3]
[1]Independent Researcher, Klang, Selangor, Malaysia
[2]Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia, Johor, Malaysia
[3]Teknik Elektro, Fakultas Teknik, Kampus 2, Universitas Muhammadiyah Surakarta (UMS), Jawa Tengah, Indonesia
sharifa@uthm.edu.my

Abstract—Leaf identification image is consistently a difficult task when using computer vision. The convolutional component extraction methods on images have their impediment and limitation, such as low accuracy, are not adaptable and less promising when converted to genuine application. The reasons are the lack of dataset needed to build a recognition model. Likewise, using the computer as a tool is bothering as it restricts the task in the research lab only. Convolutional Neural Network (CNN) shows a great solution for the computer version. Subsequently, this project utilizes the CNN's properties to solve the image classification task, and the CNN model chosen is run in Phyton coding in TensorFlow Lite. It is similar to TensorFlow's running code, but this project focused on developing Android application. It can perform faster and produce high accuracy results. This study involved four types of leaves that were considered the first Android application of the Leaf Application System for betik (Carica papaya), kari (Murraya Koenigii), pudina (Mentha) and cengal (Neobalanocarpus heimii). As a result, the model could reach around 99% accuracy with a 0.176 error rate. Hence, using CNN, an Android leaf image identifier, is proposed to solve the stated problem that can contribute to education and research.

Keywords—leaf identification, CNN, Android application, TensorFlow lite

1 Introduction

The plant is one of the main significant structures of life on the earth, with an enormous population. Plant identification or plant recognition is essential in agribusiness for the administration of plant species through botanists who can utilize this application for restorative purposes [1–5]. For people, especially kids who have little insight into knowing the type of plants, it can be challenging to become familiar with all the explicit plants. It can be easier if they take a picture of a leaf and the app uploads the photo to identify the plant [6–10]. Also, with parents and teachers' help, this is a very interactive learning process to do together.

Brisk walking is one of the physical activities that can burn more calories than walking at your usual pace. One suitable location for brisk walking is at Batu Pahat, Johor, Malaysia, Hutan Lipur Soga Perdana. Some may be very interested in flora and fauna around the hiking track during these physical activities. They can use the application installed on the phone by taking a photo of the plant leaf and the deep convolutional neural network (CNN), and automatically identifying leaf vein patterns [11–13]. However, plant recognition dependent on leaf image is utilized most widely because the leaves convey heaps of important data for recognizable proof of different plants, for example; shape, vein structure, and surface [14]. Developing a plant identification system for a personal computer with simple hardware is challenging. Building a system for Android is another different type of challenge [15].

This project aims to utilize TensorFlow, an open-source dataflow and machine learning data, to assemble an image classifying Convolutional Neural Network (CNN) [16]. TensorFlow can allow developers to assemble Neural Network layers and run on portable mobile platforms such as Android. It is the most used operating software for mobile phones in this world. Besides, it is user-friendly and not complicated in application development.

In particular, this paper considers classifying the whole surface of four types for different leave, which all the information about the trees would be available for people to learn. Hopefully, this proposed application can support the educational purposes that can attract more people to visit the forest, discover new knowledge, and recognize trees easily by identifying the leaves around them [17].

2 Leaf identification system

This paper discussed the features of the developed project. The features are the plant recognition by leaf identification system in the mobile application [18][19], related to the use of Convolution Neural Network (CNN), and the appropriate software to be used in this project.

Some digital image processing techniques are used for this task. The reviews work that has utilized digital image processing techniques to automate the identification of plant species by their leaf shapes. Image recognitions dependent on leaf images have a fixed structure framework [20]. A special gadget gains the object image (e.g., a camera). Some key highlights, which can recognize the object, are extracted from the picture through different algorithms [21]. A classifier is used for the course of action after component extraction. Most classifiers need to prepare the example before the conventional classification by utilizing the Keras library in the TensorFlow, and lastly, the outcome is obtained. The smartphone should automatically acknowledge the whole image recognition process [22].

Plant recognition is generally valuable, such as in agriculture, forestry, and medication. The plant leaves are more straightforward and open and contrast with other plant morphological structures, such as blossoms [23]. In practically all programmed leave plant identification, the state of the leaves is the most prominent feature used for identification as it is professed to be the most discriminative feature of a plant's leaf [24]. Although there is more method/algorithm available for leaf identification system, the two that are focused on in this paper are the deep learning model of CNN and the build of the interface of an application.

2.1 Deep learning of Convolutional Neural Networks (CNN)

A Convolutional Neural Network (CNN) is a Deep Learning algorithm that can take as much information from the image, transfer importance to many aspects or objects in the image and speciate one from the other. The process of preparing is essential in a CNN as it has a lot lower contrast with other order calculation techniques. CNN can gain this information through the filters or the characteristics of the image [25]. Furthermore, CNN effectively captures the spatial and temporal conditions in a picture through significant filters. As it were, the system can be prepared to comprehend the refinement for a better image.

In the making of CNN, convolutional layers are essential for detecting spatial features like the fully connected layers, as shown in Figure 1. No matter how many convolutional layers can learn, a fully connected layer could understand it too. The layer is usually utilized for picture classification, and there are restrictions in these applications [26]. A human can know the specific images more rapidly than a computer, but CNN has a 97.6% success rate in recognizing an image [27].

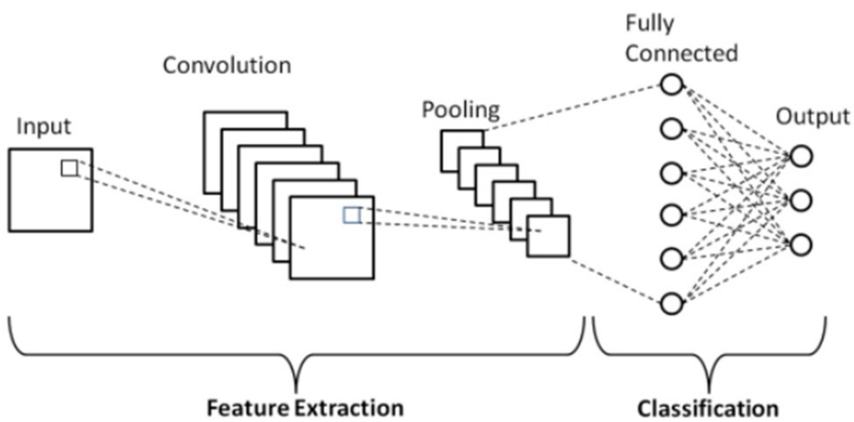

Fig. 1. CNN designed the header image of online-journals.org

2.2 Build the interface of a mobile application

To develop a leaf identification system were used image processing [28], the leaf identification system, the CNN model by Google Colab, TensorFlow library, and converting this algorithm output into an Android application for mobile phones by using Python language in Visual Studio Code [29].

3 Methodology of leaf identification system

Plant Leaf Identification is a system that can arrange types of plants based on their leaves utilizing advanced picture preparing procedures. The image is first preprocessed, and then its shape and features are extracted from the processed image.

Besides that, using the leaf identification system with suitable coding, the system is transferred into an Android application to develop a mobile software leaf identification system. There are a few plants ready to prevail for this leaf identification system which are the leaves of betik (Carica papaya), kari (Murraya Koenigii), pudina (Mentha) and cengal (Neobalanocarpus heimii), which selected leaves were done for proof of the concept of this system.

The procedure used for the leaf identification system is image processing with a specific algorithm. Figure 2 is the flowchart of the significant advances involving image processing for leaf identification systems with image capture, image pre-processing, feature extractions, and classification and training.

Fig. 2. Basic block diagram for leaf identification

3.1 Image capture

A leaf picture can be handily obtained using a smartphone camera. The image should ideally have a single shading color background with no petiole for a better result. Each image in the dataset is about 1600 × 1200 pixel resolution and has a white background and no leafstalk. The file names of all images and pictures are four-digit numbers, trailed by a "jpg" format.

3.2 Image pre-processing

The image pre-processing steps are done before the genuine investigation of the image information is proceeded to remove noise. Pre-processing alludes to the underlying preparation of leaf pictures to terminate the noise and stabilize the distorted or degraded data. Figure 3 outlines strategies like grayscale transformation, binarization, smoothing, separating, edge identification, and so on to improve the leaf image.

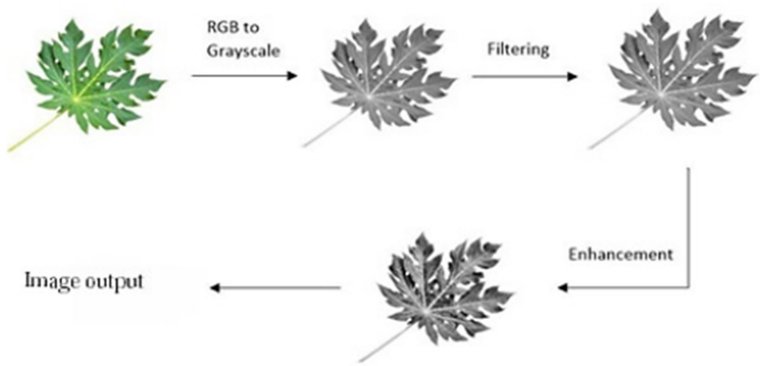

Fig. 3. Pre-processing of leaf images

3.3 Feature extraction

This strategy considers the shading and shapes highlights of the leaf. Leaves of different plants are never-ending close fit as a fiddle; accordingly, a solitary component alone may not convey expected results.

An image-looking and recovery strategy centers around the shading highlight vector by figuring the normal methods. The three-shading planes, specifically Red, Green and Blue, are isolated in the proposed calculation. For each plane line mean and segment mean of tones are determined. The normal of all line means and sections imply determined for each plane. The highlights of every one of the three planes are joined to shape a component vector. When the component vectors are generated for a picture, they are stored in an element database

3.4 Classification, training, and testing

General factual characterization is the way toward recognizing a bunch of classifications or classes. A novel perception has a place, based on earlier information, for example, a preparation dataset. The order in this work will be the cycle used to relegate a specific plant category to a picture in view of its list of capabilities. It is additionally a subset of the broader order issue in insights and AI, specifically administered learning.

4 Development of leaf identification system

Project development of this leaf identification system is summarised in Figure 4, divided into three phases to ensure successful of the project.

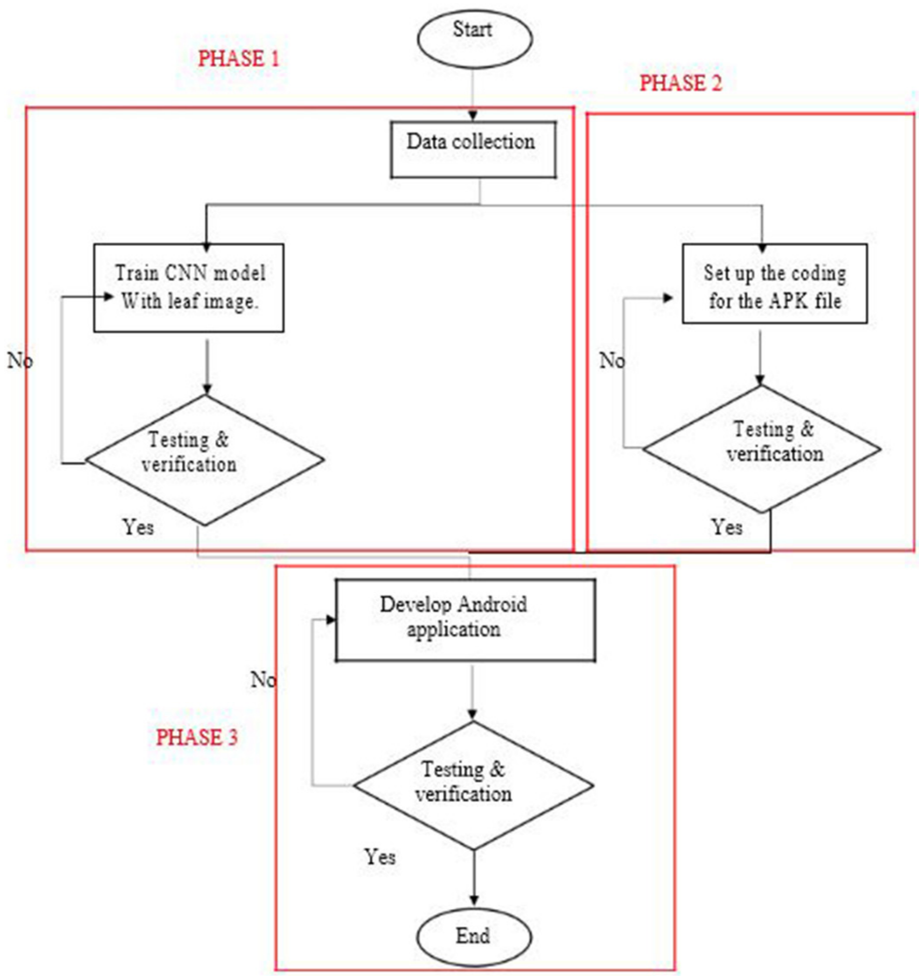

Fig. 4. Project development flowchart

4.1 Phase one

The first phase of this project is to complete the leaf identification system using image processing software. The coding must be set to matplotlib for visualizations cv2 in the Google Colab for some image pre-processing to train the model. The data collected has an imbalance and is scattered. To overcome these issues, image augmentation is used to get the most learning out of this data, especially by splitting out a validation set from the training. Next is to remove the leaf background since the raw image is taken from the camera. It removes the background image, and the training model has fewer distractions. Then, it needs to combine the leaves pixel image to the RGB basic colour. The images of leaves are collected and undergo the process of image processing, including pre-processing, feature extraction, classification and testing. Different nodes per layer have different activation functions, and the training image size loses some vital information may decrease the output accuracy. For testing and confirmation, if the system successfully identifies the leaves by using the proposed algorithm, thus phase one is achieved. Deep learning architecture selection was the critical issue for the implementation. In earlier stages, plant leaf was predicted using classic image processing techniques like threshold, contrast enhancement, and morphological contour operations. To detect at an advanced level, they have used data mining applications such as classification and clustering approaches to predict infected leaves.

Consequently, this project results in high accuracy of 99% (approximately) and effective prediction accuracy of the plant with minimum time. The trained dataset is saved as model_unquant tflite. CNN architecture basically has three main layers: Conventional layer, Pooling layer, and Fully Connected layer, that are stacked together to perform full CNN architecture. The proposed CNN model can simplify as in Figure 5. Capture image with 1600 × 1200 pixel resolution is resized to 224 × 224 pixels and a depth of three since an image is composed of RGB (Red, Green, Blue) colours. The batch size is set to 64 as a batch size that is too high requires more computational resources. A total number of prediction outputs for the classification task is the probability of 21 classes.

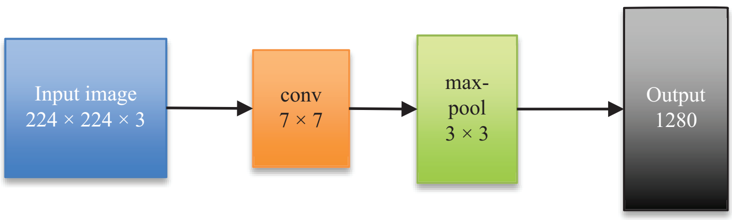

Fig. 5. Proposed CNN model

4.2 Phase two

The second stage is making the coding for the application. The methodology is to build up a programming language with translators and make graphical neighbourly UI accessible for a large portion of the cell phone. By giving these codes, all the plant data is consequently indicated at whatever point individuals snap the photo with their cell phone. The data that, as far as anyone knows to demonstrate is on the application that will be created. This project runs in Flutter, and these cross-platform instruments are expected to develop Android and iOS apps from a single codebase using a modern, reactive framework. Flutter apps are built using Dart, a basic object-oriented programming language. The main idea of Flutter revolves around widgets. The entire UI combines many types of devices, each of which defines a structural element like a button or menu, a stylistic element like a font or colour scheme, an aspect of layout (like padding), and so on. Flutter does not use OEM gadgets but provides ready-made widgets which look local either to Android or iOS.

4.3 Phase three

The last stage is to build up a completely working Android application joining both of the capacity expressed in stage one and stage two, a leaf identification proof framework on examining the leaves to perceive the sort of plants. After combining phase one, leaves datasets and phase two for the user interface for the android apps, phase three is crucial for testing and checks. The created application must satisfy the necessities. After the coding phase, the testing phase allows testing, troubleshooting, and debugging the developed system. This ensures that the system functions as expected and performs as expected and as bug-free. Hence, problem identification and debugging are essential activities during the testing phase. The testing is carried out to check and test the application of the built-in function. Test plans are made for each functional test case and documented. The application is then deployed and enters the user acceptance testing stage to ensure that the application can function as required and expected. Based on the testing results from functional and user acceptance could help valuate the requirements and the project's satisfaction. The application is open to the public, which means everyone can use the application. Application modules refer to Table 1, including the image recognition, database, and collection modules. The image recognition module focuses on an image recognition task that would output the detected leaf species to the user.

Table 1. Module in application

Module	Function	User
Image Recognition module	Detect images captured from the camera and recognize the leaf species. Display a description of the leaf.	Public
Database module	Provide a platform for a user to search flower images available	
User collection module	Allow users to know the type of leaf and view the image descriptions	

In contrast, the database module provides a platform for users to take in real life images available in the database. After capturing images, the user is able to know the plant on the spot. This project is also assumed to have the highest recognition accuracy within the range from 90% to 95%. The language used for this project includes Python. To ensure the developed CNN model works correctly and efficiently, it is first tested for the test set, which refers to the dataset that is never seen before.

5 Results

The result and analysis that have been done for this project are data collecting and running image processing technique on TensorFlow by using Keras method and get the output result of the testing sample data. System requirement is specified to fulfil the user requirement specification.

5.1 Data collection

Data collected are a significant part of this project to ensure the system runs all those features. For the first feature, the leaf identification system, data that needs to be collected is the image of leaves that the system needs to recognize. There are two conditions of leaves: white background and leaves captured on site of the plant. Figure 6 shows the sample of leaves image with a white background.

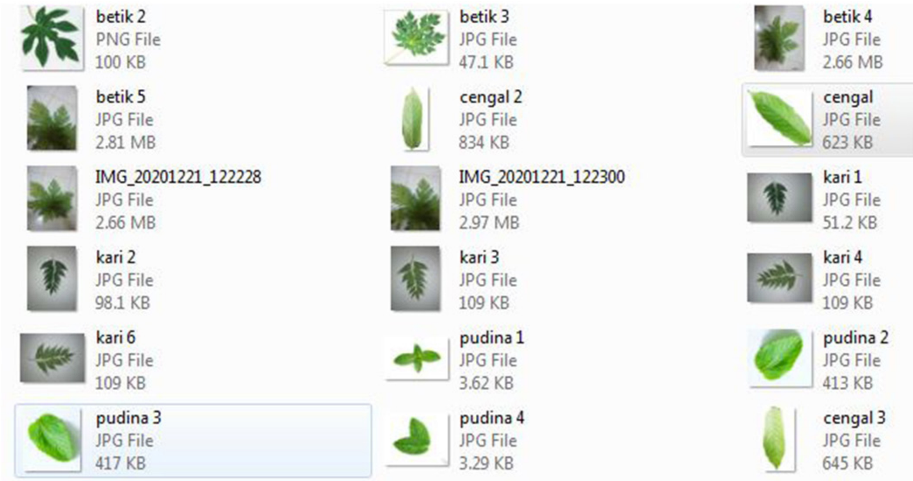

Fig. 6. Samples of leaves data

5.2 Image processing

For image processing, TensorFlow' is an appropriate software to be used. There are two sections for which the data is used for the image processing to be successfully done. The sections are leaves with white background and leave converted into pre-processing and feature extraction techniques by operating on a single image.

Converting image leaves with a white background can be done using TensorFlow code by first converting each RGB original image into a grayscale image, as shown in Figure 7.

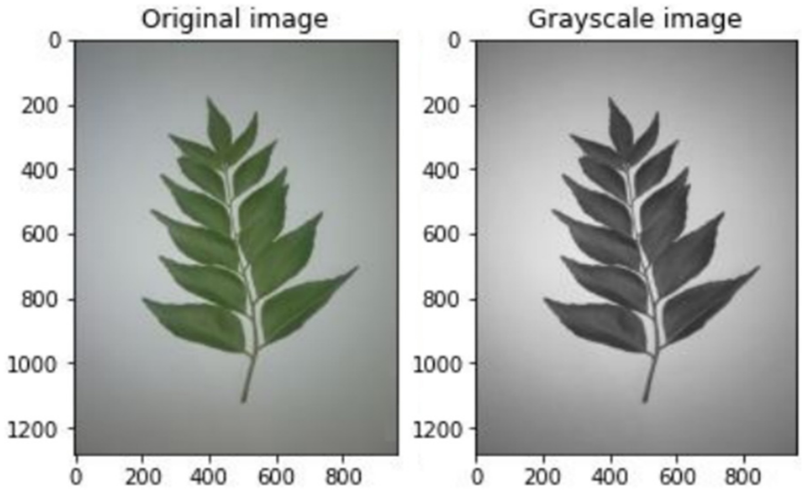

Fig. 7. Original image and grayscale image of kari leaf

Once the image is converted to grayscale, the new image is filtered to reduce noise on the image for smoothing the image. The suitable filter used in this process is a Gaussian filter, and its result is shown in Figure 8.

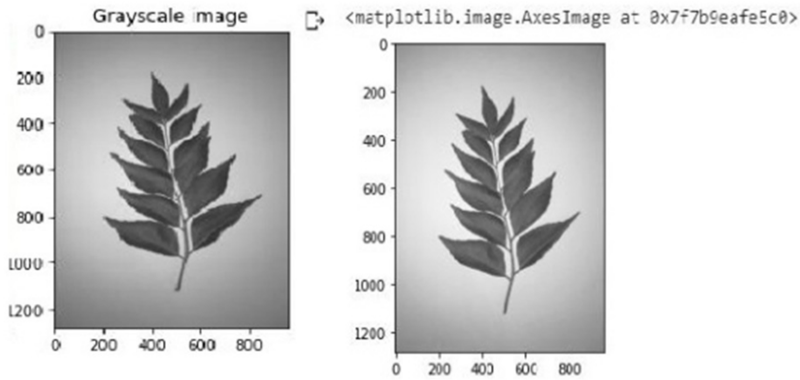

Fig. 8. Grayscale image and filtered image using Gaussian filter

5.3 Model training

The learning rate is set lower as too high of a learning rate may cause the pre-trained network to overfit faster. The metrics and loss function used are accuracies and categorical cross-entropy. An epoch is a term utilized in machine learning and specifies the number of passes of the entire training dataset the machine learning algorithm has completed. The batch size is the entire training dataset, and then the number of epochs is the number of iterations. The whole model building and training are developed using the TensorFlow Python library. The result of the test set is tabulated in Figure 9. It shows a great improvement of 95% and 99% in test accuracy. The loss in this project is because of the number of datasets that have been obtained. The bigger the dataset, the better the expected result. As the size of data increases, many new difficulties will emerge.

```
validation_steps=val_steps_per_epoch).history

    Epoch 1/10
    4/4 [==============================] - 72s 22s/step - loss: 1.7319 - acc: 0.2364 - val_
    Epoch 2/10
    4/4 [==============================] - 19s 4s/step - loss: 1.1447 - acc: 0.4892 - val_l
    Epoch 3/10
    4/4 [==============================] - 19s 4s/step - loss: 0.8365 - acc: 0.6820 - val_l
    Epoch 4/10
    4/4 [==============================] - 19s 4s/step - loss: 0.4008 - acc: 0.8826 - val_l
    Epoch 5/10
    4/4 [==============================] - 19s 4s/step - loss: 0.2638 - acc: 0.9755 - val_l
    Epoch 6/10
    4/4 [==============================] - 19s 4s/step - loss: 0.1763 - acc: 0.9930 - val_l
    Epoch 7/10
    4/4 [==============================] - 19s 4s/step - loss: 0.1756 - acc: 0.9580 - val_l
    Epoch 8/10
    4/4 [==============================] - 19s 4s/step - loss: 0.1440 - acc: 0.9654 - val_l
    Epoch 9/10
    4/4 [==============================] - 19s 6s/step - loss: 0.0968 - acc: 0.9960 - val_l
    Epoch 10/10
    4/4 [==============================] - 19s 4s/step - loss: 0.0717 - acc: 0.9939 - val_l
```

Fig. 9. Second stage training model summary

5.4 Interface design

Interface design is a crucial component before application development using code. It shows a roadmap of application navigations as well as how the application could integrate the functions with user interactions. The first interface design is the application landing page, also known as the startup page. Figure 10 is the icon of the application when the installation is completed.

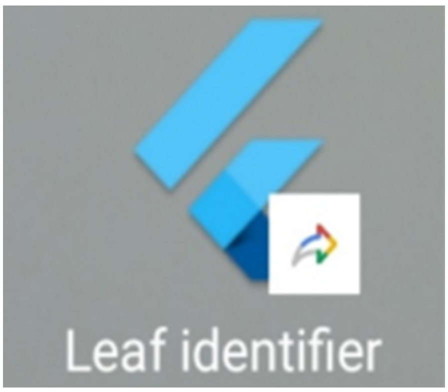

Fig. 10. The icon of the application

Figure 11 (a) shows the landing page interface in Leaf identification. Users could use the smartphone camera to capture the image, and the app processes and recognizes the leaf in the image. Navigation to the collection or explore page could be done through the tabs located at the bottom of the screen. The Leaf Identification will instantly trigger the CNN model to recognize the plant through the leaf images. Figure 11 (b) shows the result after the app identified the leaf species. The identified result page displays the identified species name as well as other similar species. While Figures 11 (c) and (d) show the leaf image is measured as the percentage of the CNN is calculated. The information and the percentage result are based on the accuracy of the data trained. As shown in Figure 11, all of the results are in the range of 97% to 100% accurate.

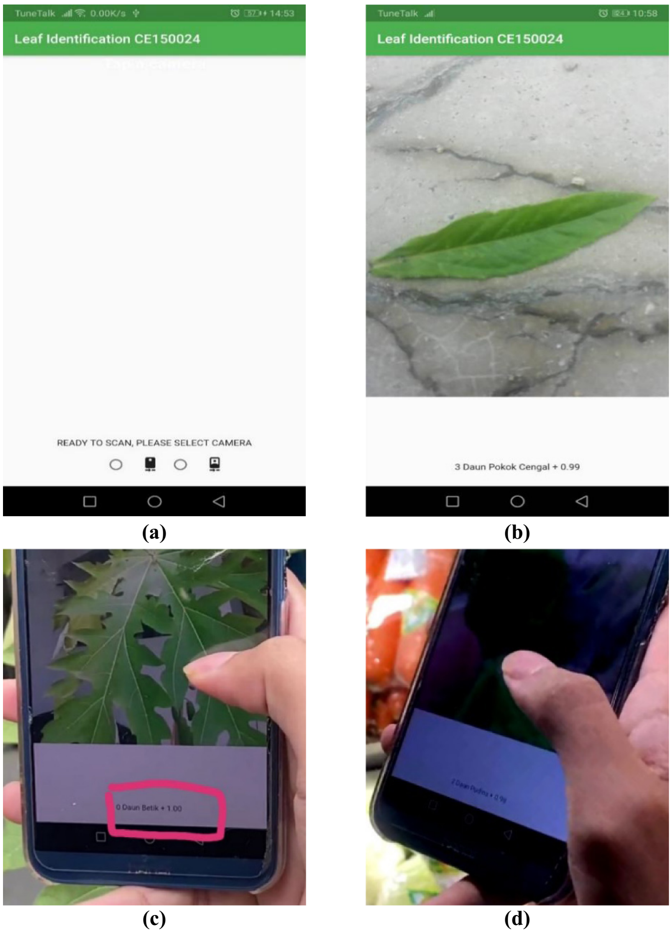

Fig. 11. (a) Main interface page, (b) Leaf information result shown at the bottom view; cengal with 99% accurate, (c) Betik leaf information with 100% accurate, (d) Pudina leaf information with 97% accurate

6 Conclusion

According to early results, all data collected were gathered on a computer as the main database. The images are used on a workspace in TensorFlow for image processing procedures, including pre-processing, grayscale, filtering, and enhancing. Conversion of RGB to Grayscale image method applied to the processed images to calculate the parameters of the leaves. Values of each parameter are recorded to obtain the specific results of the leaf identification system. This proposed CNN model of Android application of Leaf Application System with four types of leaves betik (Carica papaya), kari (Murraya Koenigii), pudina (Mentha) and cengal (Neobalanocarpus heimii), are successfully developed and can be considered as an additional tool for the education purposes. The accuracy of this Android application leaf identification system is 99%.

7 Acknowledgment

Communication of this research is made possible through monetary assistance by Universiti Tun Hussein Onn Malaysia and the UTHM Publisher's Office via Publication Fund E15216.

8 References

[1] Satti, V., Satya, A., & Sharma, S. (2013). An automatic leaf recognition system for plant identification using machine vision technology. International Journal of Engineering Science and Technology, 5(4), 874.
[2] Adedoja, A. O., Owolawi, P. A., Mapayi, T., & Tu, C. (2022). Intelligent mobile plant disease diagnostic system using NASNet-mobile deep learning. IAENG International Journal of Computer Science, 49(1), 216–231.
[3] Gong, L., Yu, C., Lin, K., & Liu, C. (2022). A lightweight powdery mildew disease evaluation model for its in-field detection with portable instrumentation. Agronomy, 12(1). https://doi.org/10.3390/agronomy12010097
[4] Cham, M. F. X., Tanone, R., & Riadi, H. A. T. (2021). Identification of rice leaf disease using convolutional neural network based on Android mobile platform. Paper presented at the 2021 2nd International Conference on Innovative and Creative Information Technology, ICITech 2021, 140–144. https://doi.org/10.1109/ICITech50181.2021.9590188
[5] Mohandas, A., Anjali, M. S., & Rahul Varma, U. (2021). Real-time detection and identification of plant leaf diseases using YOLOv4-tiny. Paper presented at the 2021 12th International Conference on Computing Communication and Networking Technologies, ICCCNT 2021. https://doi.org/10.1109/ICCCNT51525.2021.9579783
[6] Shanmugamani, R. (2018). Deep learning for computer vision: Expert techniques to train advanced neural networks using TensorFlow and Keras. Packt Publishing Ltd.
[7] Liu, Y., Su, J., Shen, L., Lu, N., Fang, Y., Liu, F., Su, B. (2021). Development of a mobile application for identification of grapevine (vitis vinifera l.) cultivars via deep learning. International Journal of Agricultural and Biological Engineering, 14(5), 172–179. https://doi.org/10.25165/j.ijabe.20211405.6593

[8] Hui, Z., Tianpeng, Z., Hong, L., Maohua, X., Zijian, W., & Juanjuan, W. (2021). Rice disease identification and classification based on matlab and android platform. International Journal of Mechatronics and Applied Mechanics, 1(9), 108–118. https://doi.org/10.17683/ijomam/issue9.16
[9] Anbarasi, M., Garg, K., Garg, S. D., Yokesh Babu, S., & Saleem Durai, M. A. (2019). An android application for plant leaf disease detection using convolution neural network. International Journal of Innovative Technology and Exploring Engineering, 8(12), 4133–4137. https://doi.org/10.35940/ijitee.L3649.1081219
[10] Deepa, N. R., Nagarajan, N., Palanivelrajan, D., & Balamurali, R. (2019). Image processing system for detecting and recovering from cotton leaf disease using the android application. International Journal of Innovative Technology and Exploring Engineering, 8(9), 3366–3381. https://doi.org/10.35940/ijitee.I8977.078919
[11] Cheng, Q., Zhao, H., Wang, C., & Du, H. (2018). An android application for plant identification. Paper presented at the Proceedings of 2018 IEEE 4th Information Technology and Mechatronics Engineering Conference, ITOEC 2018, 60–64. https://doi.org/10.1109/ITOEC.2018.8740771
[12] Xie, X., Zhang, X., He, B., Liang, D., Zhang, D., & Huang, L. (2016). A system for diagnosis of wheat leaf diseases based on android smartphone. Paper presented at the Proceedings of SPIE—the International Society for Optical Engineering, 10155. https://doi.org/10.1117/12.2246919
[13] Culman, M. A., Gomez, J. A., Talavera, J., Quiroz, L. A., Tobon, L. E., Aranda, J. M., Bayona, C. J. (2017). A novel application for identification of nutrient deficiencies in oil palm using the internet of things. Paper presented at the Proceedings—5th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering, Mobile Cloud 2017, 169–172. https://doi.org/10.1109/MobileCloud.2017.32
[14] Dangtongdee, K. G. (2018). Plant Identification Using Tensorflow. Final Report, California Polytechnic State University, San Luis Obispo.
[15] Gulli, A., & Pal, S. (2017). Deep learning with Keras. Packt Publishing Ltd.
[16] Chandra, A. Y., Prasetyaningrum, P. T., Suria, O., Santosa, P. I., & Nugroho, L. E. (2021). Virtual Reality Mobile Application Development with Scrum Framework as a New Media in Learning English. International Journal of Interactive Mobile Technologies (iJIM), 15(08), 31–49. https://doi.org/10.3991/ijim.v15i08.19923
[17] Zhu, H., Liu, Q., Qi, Y., Huang, X., Jiang, F., & Zhang, S. (2018). Plant identification based on very deep convolutional neural networks. Multimedia Tools and Applications, 77(22), 29779–29797. https://doi.org/10.1007/s11042-017-5578-9
[18] Lux, M., & Bertini, M. (2019). Open source column: Deep learning with Keras. ACM SIGMultimedia Records, 10(4), 7–7. https://doi.org/10.1145/3310195.3310202
[19] Bharali, P., Bhuyan, C., & Boruah, A. (2019, May). Plant Disease Detection by Leaf Image Classification Using Convolutional Neural Network. In International Conference on Information, Communication and Computing Technology, Springer, Singapore, 194–205. https://doi.org/10.1007/978-981-15-1384-8_16
[20] Sudharshan, D. P., & Raj, S. (2018, January). Object recognition in images using convolutional neural network. In 2018 2nd International Conference on Inventive Systems and Control (ICISC), IEEE, 718–722. https://doi.org/10.1109/ICISC.2018.8398912
[21] Latifi Oskouei, Seyyed Salar, et al. (2016). CNNdroid: GPU-accelerated execution of trained deep convolutional neural networks on android. Proceedings of the 24th ACM international conference on Multimedia. https://doi.org/10.1145/2964284.2973801

[22] Nguyen, Quang-Khue, Thi-Lan Le, and Ngoc-Hai Pham. (2013). Leaf based plant identification system for android using surf features in combination with bag of words model and supervised learning. International Conference on Advanced Technologies for Communications (ATC 2013). IEEE. https://doi.org/10.1109/ATC.2013.6698145
[23] Wäldchen, Jana, et al. (2018). Automated plant species identification—Trends and future directions. PLoS computational biology, 14(4). https://doi.org/10.1371/journal.pcbi.1005993
[24] Ibrahim, Zaidah, Nurbaity Sabri, and Nur Nabilah Abu Mangshor. (2018). Leaf recognition using texture features for herbal plant identification. Indonesian Journal of Electrical Engineering and Computer Science, 9(1), 152–156. https://doi.org/10.11591/ijeecs.v9.i1.pp152–156
[25] Mareta, Affix, Indah Soesanti, and Oyas Wahyunggoro. (2018). Herbal leaf classification using images in natural background. 2018 International Conference on Information and Communications Technology (ICOIACT). IEEE. https://doi.org/10.1109/ICOIACT.2018.8350775
[26] Geetharamani, G., & Arun Pandian. (2019). Identification of plant leaf diseases using a nine-layer deep convolutional neural network. Computers & Electrical Engineering, 76, 323–338. https://doi.org/10.1016/j.compeleceng.2019.04.011
[27] Zhu, Xiaolong, Meng Zhu, and Honge Ren. (2018). Method of plant leaf recognition based on improved deep convolutional neural network. Cognitive Systems Research, 52, 223–233. https://doi.org/10.1016/j.cogsys.2018.06.008
[28] Hu, Jing, et al. (2018). A multiscale fusion convolutional neural network for plant leaf recognition." IEEE Signal Processing Letters, 25(6): 853–857. https://doi.org/10.1109/LSP.2018.2809688
[29] Zhu, Xiaolong, Meng Zhu, and Honge Ren. (2018). Method of plant leaf recognition based on improved deep convolutional neural network. Cognitive Systems Research, 52, 223–233. https://doi.org/10.1016/j.cogsys.2018.06.008
[30] Nalawade, Rohit, et al. (2020). Agriculture field monitoring and plant leaf disease detection. 2020 3rd International Conference on Communication System, Computing and IT Applications (CSCITA). IEEE. https://doi.org/10.1109/CSCITA47329.2020.9137805

9 Authors

Mohammad Hazwan Hakim Minhat is an independent researcher. He is a graduate of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia, with comprehensive experience in software development. (email: mhazwanhakim09@gmail.com).

Sharifah Saon is Senior Lecturer in the Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia, Malaysia and registered Professional Technologists. Her research interest is in the area of theoretical digital signal processing, visible light communication and digital & data communication. Including the application to Internet of Things (IoT) and bigdata analysis. She is a member of IEEE, Institute of Engineering Malaysia (IEM), Board of Engineering Malaysia (BEM), and Professional Technologist of Malaysia Board of Technologists (MBOT). (email: sharifa@uthm.edu.my).

Abd Kadir Mahamad is Associate Professor at Faculty of Electrical and Electronic Engineering UTHM and registered as Professional Engineer. During the period of May 2015 through May 2016, he was doing industrial attachment at Melaka ICT

Holdings Sdn Bhd, as Executive Assistant Manager and was involved in Smart City project in Melaka. He currently leads a research team in Video Analytic and Internet of Things (IoT). His research interests include Deep Learning, Smart City, Intelligent System applications and embedded system. He is also a Senior Member of IEEE, Institute of Engineering Malaysia (IEM) and Board of Engineering Malaysia (BEM). (email: kadir@uthm.edu.my).

Umi Fadlillah is currently pursuing a PhD at the Faculty of Electrical and Electronic Engineering at Universiti Tun Hussein Onn Malaysia. She was also assigned by the Higher Education Service Institute, Region VI, Central Java, to teach at the Electrical Engineering Study Program, Faculty of Engineering in Universitas Muhammadiyah Surakarta, Indonesia. Her research interests include telecommunications technology, antennas and wave propagation, and electronics engineering. She is a member of FORTEI, IAOE, IAENG, and IEEE. (email: uf138@ums.ac.id)

Article submitted 2022-04-22. Resubmitted 2022-06-13. Final acceptance 2022-06-22. Final version published as submitted by the authors.