Development of mobile-based application for practical wood identification

M R Arifin¹,², E P¹, B Sugiarto¹, R Wardoyo¹, and Y Rianto³

¹Computer Vision Research Group, Research Center for Informatics, Indonesian Institute of Sciences (LIPI), Bandung, Indonesia
²Department of Informatics, Faculty of Engineering Universitas Sangga Buana YPKP, Bandung, Indonesia
³Center for The Utilization and Innovation of Science and Technology, Indonesian Institute of Sciences (LIPI), Cibinong, Indonesia

*E-mail: muhammadrosyidarifin@gmail.com

Abstract. Wood Identification is an important task, especially for wood anatomist. The identification process is required in many fields, such as custom ports, forest survey, and wood industries. However, expertise-in wood identification is limited due to a few wood anatomists. To enable wood identification can be performed in a broader area, a mobile-based application is proposed to realise these purposes. The paper presents the development of a mobile-based application for wood identification. A dataset of Indonesian commercial wood images of cross-section surface was collected from Xylarium Bogoriense, The Ministry of Environment and Forestry of The Republic of Indonesia. The images were collected using a smartphone camera at optical magnification level around 200 times and a minimum resolution of 12 megapixels. The collected images were then used to develop a deep learning-based algorithm to classify wood species. These images are considered as the training datasets, whereas testing dataset is collected in another session. The final model obtained through the training process is then stored in a cloud-based server at Research Center for Informatics, Indonesian Institute of Sciences. A mobile application, namely AIKO stands of Indonesian words - Alat Identifikasi Kayu Otomatis (Automatic Tool for Wood Identification), is developed to enable image acquisition and images transferring to the server. AIKO application sends the acquired image to the trained identification model in the server. The output of the identification model is the species name of the observed images. Besides species/botanical name, AIKO also provides information on trade names, specific gravity and density, durability class, strength class, commercial timber classification, and recommended utilization.

1. Introduction
Indonesia has more than 4,000 wood species grown in its tropical rain forest. As one of the primary forest products, wood plays an important role to fulfill human needs. Therefore, it is essential to maintain its sustainability [1]. Wood identification is an important task, especially for wood anatomist. The
identification process is essential in several fields, such as custom ports, forest survey, and wood industries. However, wood identification capability is limited to a few wood anatomists.

In terms of wood identification process, there is still no assessment of the results of wood identification and availability of the identification process still challenging to fulfil. This condition occurs due to wood anatomist who can identify wood is limited in term of number and distribution. Condition and level of fatigue can influence the results of wood identification. Furthermore, if the researcher’s conditions are unwell, it might affect the identification accuracy. Besides, identification time could take longer. In order to enable identification work can be applied in broader area, a mobile-based application for wood identification is proposed. This application can be used using a mobile device which has an additional lens using a monocular lens as a replacement for a digital microscope. Monocular is a type of microscope that has only one ocular lens. The usefulness of this eyepiece is to increase the magnification in storing simple objects. Additionally, the mobile device is chosen because it is handy and can be used to capture an image anytime and anywhere.

In application development of an automatic wood identification application, namely AIKO, which is an abbreviation of Indonesian words – Alat Identifikasi Kayu Otomatis (Automatic Tool for Wood Identification). The app developed to enable image acquisition and data transfer to the server. In the process of wood identification, the images were collected by using a smartphone. The collected images are then used to find the weights of deep learning architecture. The optimum weights are considered as final coefficients of the trained model. The AIKO application sends the acquired image to the server, so the trained classification algorithm can provide the name of wood species and the identification accuracy.

2. Previous Works
Many researchers have conducted several works related to computer vision using a smartphone application. This paper proposes a mobile-based application for identifying Indonesian wood species. Development of a smartphone application has also conducted by Malaysian researchers. Tang et al. implemented deep learning algorithms to classify Malaysian wood types. The app can be used to identify about 100 wood species [2]. Tapu et al. [3] introduced a real-time obstacle detection and classification system designed to assist visually impaired people to navigate safely by using a smartphone. By utilising an adaptation HOG descriptor and k-means clustering algorithm, the detection of obstacle can be achieved with high accuracy rates. Khan et al. [4] showed the applicability and feasibility of different machine learning techniques on iris recognition from smartphone captured eye image. Through several steps and methods such as Daugman, canny edge detection and other methods, the result reached accuracy about 99.7% for training and 97% for testing. Sari et al [5], proposed a new framework to preprocessing tomato images using a combination of Linear Regression algorithm and V-Channel Otsu segmentation. This experiment used four types of smartphone devices and used a variance of Laplacian and produce a robust algorithm for blurred images. Finally, Shih et al. [6] proposed the maximally stable external region (MSER) algorithm to accurately detect all traffic sign candidates in real-time for an automatic driving assistance system (ADAS). By using HOG detector and LSVM classifier, the achieved accuracy was about 96% precision rate for recognition on the German Traffic Sign Recognition Benchmark (GTSRB).

3. Methodology
3.1. Data Preparation
Wood data is taken using a smartphone camera Samsung S9+ with 12 megapixels and an additional microscope for smartphone with magnification 60×. The application set a digital magnification at 3.5×.
By applying this magnification setting, the system will have a total magnification at 210× (60 times 3.5). The Samsung S9+ is equipped with several parts such as processor Snapdragon 845 Octa Core, CPU Kyro 385 Gold 2.8 GHz, GPU Andreno 630, RAM 6 Gigabyte, internal storage 64 Gigabyte, and primary camera with dual 12 Megapixel.

The wood samples are obtained from the collections of Xylarium of Bogoriense, Forest Products Research and Development Center (FORDA), The Indonesian Ministry of Environment and Forestry of The Republic Indonesia. The size of these wood images was 2160 x 2160 pixels. Figure 1 shows the examples of the cross section and texture image of the wood samples. In this paper, there are 158 species of wood used as datasets for training data into deep learning architecture. The research only used wood samples from the Xylarium. Therefore, samples from different locations were not yet collected in this paper. A total of 200 images were collected for each species. A variation on orientation and position points are applied during the image acquisitions.

![Figure 1. Example of wood images from several species](image)

3.2. Identification Process

In the process of wood identification, a data-set of Indonesian commercial wood surface images was collected from Xylarium Bogoriense, The Indonesian Ministry of Environment and Forestry of The Republic of Indonesia. The images were collected by using a smartphone camera at a total magnification of 210×. The collected images were then used to find the weights of deep learning architecture. The trained model is then stored in a cloud-based server at Research Center for Informatics, Indonesian Institute of Sciences. AIKO application sends the acquired image to the trained the identification model in the server. The image classification process is carried out on could-based server to produce of wood species and accuracy so that the output of identification model is the species name of the observed images.

3.3. Deep Learning

The artificial intelligence technology aims to provide a machine system that can mimic human intelligence. The system will learn by the expertise and acquire skills without human involvement. Deep learning is a subset of machine learning where the algorithm is processed like a human brain, learn from massive amounts of data or information. The similarity to how human learns from experience, the deep learning rule would perform a task repeatedly, each task that is repeated will improve the outcome. A part of deep learning is that neural networks have various (deep) layers that enable learning. Just about any problem that requires “thought” to figure out is a problem, deep learning can learn to solve [7].

The process of deep learning in image classification can be using the Convolutional Neural Network (CNN) method. Like the traditional neural network, CNN is composed of input layer, hidden layer and output layer. The difference is that the input of CNN is the image (the pixel matrix), and the output is
the image feature obtained by the convolution calculation [8]. The process of convolution can be seen in Figure 2.

![Figure 2. Explanation of the convolution process][1]

In the CNN, the convolution kernel is the most crucial part. The convolution kernel is a two-dimensional matrix with size n*n, in which each point has a corresponding weight. A convolution kernel corresponds to a neuron, and the size of the convolution kernel is called the receptive field of the neuron. Once an image is applied into the CNN algorithm, a calculation process starts k convolution kernels in the system to carry out convolution calculation on the image. The weight values in the convolution kernel and the pixel values at the corresponding position of the image are summed within the receptive field. Following this, the convolution kernels slide to the next position of the image according to the step size. The process is repeated until all the pixels in the image ultimately counted. The output of the pixel matrix is the feature map of the original image, and k convolution kernels result in k feature maps [8].

3.4. Flask Framework
Flask is a framework by most standards-small can be called a “micro-framework”, Flask aims to maintain the core of the framework remains small but highly extensible. The Flask provides ease, flexibility and power for developers to choose the configurations they needed for their applications, without imposing any restriction on the choice of database, template, and engine (so on is very informal writing) [9]. Flask has three main dependencies. The routing, debugging, and Web Server Gateway Interface (WSGI) subsystems; the template support; and the command-line integration that these dependencies are all authored by Armin Ronacher, the author of Flask [10].

3.5. Implementation of REST API
The process of wood identification is carried out on two different devices-mobile application and cloud-based server using flask framework. An intermediary system is needed to connect both devices. The system can act as a communication media between two different machines. The processes that often occur in this communication are requesting and responding. The device that requests data can be a mobile application, desktop application, or web application or most commonly known as the client. This client sends the request to the server or widely known as the web-server. The server will receive and respond that request data. The server will provide a response that requests data with sending the response data to the client in JSON format or XML format.
Application Programming Interface (API) is a set of functions, subroutine, communication protocol, or tools that can be used by any programs. This API can be applied to a programming language, library, framework, operation system, and web service. One part of the web service is the REST API. REST is a stand of Representation State Transfer which is an architectural style defined to help create and organize distributed system [11].

3.6. Android Operating System

Android is an operating system for smartphones, tablets, and various mobile devices. The operating system can be described as a hub between devices and users. The user can interact with the device and run installed applications on the device. Several operating systems that are widely known such as Windows, Mac OS and Linux [12]. Android can be categorised as a Linux-based operating system for mobile devices. This platform is provided free and numerous kinds of applications that have been developed to be run on Android-based devices [13].

Android provides various tools and frameworks for developers to enable fast software development [14]. The Android SDK (Software Development Kit) is considered a tool of API (Application Programming Interface) that required in the development of Android-based application. The Android operating system consists of various layers, whereas each layer has its own characteristics and purposes. The layers are not separated and often mixed. Figure 3 shows the parts of the android architecture.

- Linux kernel is responsible for managing and communicating with the hardware parts of the device.
- The library is a set of codes that can be used by the software components or programs. The program code is usually written using the C or C++ languages.
- Android runtime is a section where applications can be run. Android application is written using Java languages. Dalvik Virtual Machine (DVM) is demanded to execute the programs. The DVM is an optimized virtual machine for a device with small memory, limited power source and small processor capability. This DVM will be utilised to provide Java code translations for the operating system.
- Application Framework is a portion that can be used by a developer to create Android applications. The developer uses the same frameworks as utilised by the main applications of Android. This condition may allow developers to enrich their applications.
- Application is a part that can be accessed by users. Various applications are installed and maintained by themselves according to their objectives.
3.7. Utilisation of Flutter

Flutter is a Software Development Kit (SDK) delivered by Google that can be used to create mobile applications. The Flutter SDK adopts Dart programming language in their coding environment. Android application and iOS can be created using a code base and the same programming language. Google produced dart programming language in 2011 and Flutter was first published in May 2017. Previously, the native application for Android is made using Java or Kotlin programming languages, and native application for iOS is made using Objective-C or Swift programming languages. Flutter is indicated to make it easier and more effective in the process of developing a mobile application. The flutter can compile the codes either in Android or iOS operating systems. Therefore, to deliver an application for both operating systems, development processes in two different programming languages can be avoided [15]. The advantages of Flutter programming are given as follows [16]:

- The package module is automatically connected to Flutter. Thus, procedures to call the module manually using a terminal is not required.
- Dart programming language applies object-oriented programming (OOP) paradigms in software development.
- Manually setup can be performed easier. A new library might be added through pubspec.yaml file.
- Programming works are fast and more reliable. Data management uses state; therefore, it can minimise the programming complexities. Flutter provides a hot reload feature. This feature can speed up the debug process of the developed code.
- Flutter is compatible with widely known IDE platforms, such as Android Studio and Visual Code.

Figure 3. Diagram of the Android Architecture [13]
4. Development of AIKO
The AIKO is created by using Android Studio through the Flutter framework. There are four main menus in the AIKO application, namely, Identification, Wood List, History, and Information. Three steps are sequentially conducted in the Identification Menu. Those steps are image capturing, sending the image to the server, and displaying the identification results. The user might view the available wood types in the AIKO database through the Wood List menu. The History menu can be used to retrieve the previous identification results. The date of identification and captured wood images is also able to obtain using this menu. The last menu is Information. The users could explore manual instruction and development information of AIKO via this menu. The user interface that has been created can be seen in Figure 4.

![Figure 4. User Interface of the AIKO application.](image)

The process that happened in wood identification using mobile application consist of capturing an image by using a smartphone camera at an optical magnification of 210x, then send the acquired image to the server. This magnification level is defined to approximate the current magnification level used by the wood anatomists. The identification accuracy would suffer if an image with different resolution applied to the system. On the server, received image is processed into the trained identification model. The output of this process is the list of identified wood consisting of wood id and accuracy of wood identification and then send into the mobile application. If the accuracy of the identified wood is more than 90%, the received image will be store in the database. After wood identification processed, the server will send a list of identified wood to the mobile application with JSON format.

In terms of mobile application, AIKO will decode the received JSON which there are a wood id and accuracy. If accuracy is more than 90%, AIKO will display the message that wood has been successfully identified and provides information on trade names, specific gravity/density, durability class, strength class, commercial timber classification, and recommended. Figure 5 show the Architecture wood identification.
Figure 5. Architecture of the wood identification system

In capture image section Figure 8, the user will take an image of wood with attaching a microscope lens to smartphone displayed on Figure 6. The wood sample has been sliced first using a cutter that the wood pore image looks bright and useful as in Figure 7. AIKO will show the preview of the captured image to give choices to the user whether to identify the wood image or re-take a new picture.

Figure 6. An additional microscope lens is attached to the smartphone camera.  
Figure 7. Make an incision at wood cross-section  
Figure 8. Capture wood image using the application

The user then sends the request of wood identification with the post method by sending the wood image that has been taken. Next, the user will receive the response from the server with JSON format contained wood id and percentage of wood identification. If accuracy is more than 90% AIKO will display the message that wood has been successfully identified and show information about the identify of wood.
Figure 9. The final description is given for successful identification.

Figure 10. Details of identified wood

4.1. Server-side
In this section, the flask framework is used to run a python program on the web. A place to store the trained model of deep learning, provide information about wood consisting of a species name, trade names, specific gravity/density, durability class, strength class, commercial timber classification, and recommended. This-section also provides a function of storing identified wood images which can be used to improve accuracy in the next wood image training models.

Figure 11. REST API in Flask Framework
Figure 11 shows REST API that available at Flask Framework that connects a mobile application with a cloud-based server. Url /upload has a function of wood identification by receiving wood images from a mobile application. Then the received images are processed into the trained identification model. The output of this function is the server that will respond or send the list of identified wood to mobile application (client) in JSON format. The.Url /getWoodList has a function to provide a list of woods. It also has a function to give an information species name, trade names, specific gravity/density, durability class, strength class, commercial timber classification, and recommended of a specific wood. Url /assessments have a function to save the history of wood identification to a database and store the identified wood to be used in further research.

This flask framework will load the deep learning model and classes to be used when mobile application sends a request through the URL /upload with a post the wood image. In this process, the received image will be processed through image pre-processing and then do the identification algorithm. The following is a flow of wood identification algorithm that can be seen in Figure 12.

![Figure 12. Wood identification algorithm.](image)

4.2. Testing Wood Identification

In the testing experiment, we were testing the wood identification using AIKO by identifying ten wood species from 158 wood species. Each species tested 18 times to process wood identification. Each testing of wood identification results in three probability. The first namely “Success” for correct wood identification. The second is “Wrong” for incorrect wood identification, and the last is “Fail” if the accuracy of identification from deep learning architecture is below 90%.

Meanwhile, for the first and the second probabilities can give a final accuracy higher than 90%, the second probability might not be used. Because the identification provides an incorrect species name. The testing results indicate that wood identification can be done using a mobile application that is connected with deep learning architecture through the REST API.

After all steps of the wood identification process are completed, identification testing is carried out to produce accuracy. The results of the wood identification test can be seen in Table 1. Successful identification can be found significantly for three species, namely Jati, Kapur, and Bipa. As shown in Figure 13, these woods have two notable features compared to the other species. These feature are
marked wood vessels and parallel lines of tree growth rings. The edges of these features can improve the algorithm capability in discriminating the wood species.

### Table 1. Testing Result of Wood Identification

| Species Name                        | Trade Name | Success | Wrong | Fail | Accuracy |
|-------------------------------------|------------|---------|-------|------|----------|
| *Tectona grandis* L                 | Jati       | 16      | 1     | 1    | 89%      |
| *Intsia palembanica*                | Merbau     | 14      | 2     | 2    | 78%      |
| *Falcataria moluccana*              | Sengon     | 14      | 3     | 1    | 78%      |
| *Dyrobalanops oocarpa V. SI.*       | Kapur      | 17      | 1     | 2    | 78%      |
| *Alstonia marcophylla* Wall         | Pulai      | 13      | 5     | 1    | 72%      |
| *Vatica rassak* B. L.               | Resak      | 15      | 2     | 2    | 83%      |
| *Octomeles sumatrana Miq*           | Benuang    | 14      | 3     | 1    | 78%      |
| *Pterygota forbesii* F. v. M.*      | Bipa       | 16      | 2     | 1    | 89%      |
| *Serianthes minahassa* Merr.*       | Bowoi      | 15      | 3     | 3    | 83%      |
| *Vitex cofassus* Reinw*             | Gopasa     | 13      | 2     | 3    | 72%      |

Misclassification cares are still significantly found in two species, namely Pulai and Gopasa. As depicted in Figure 14, both species have wood vessel at the small size. Their growth tree ring is also not distinct as found in the other species with reasonable accuracies. Downsizing process in the identification algorithm can degrade these essential features. This process is required to reduce the bandwidth cost of sending images from the application to the server. Hence, good identification accuracy is hard to achieve for these species. An optimum downsizing scale should be defined to minimise inaccurate cases.

![Figure 13](image1.jpg) Three species with good accuracies, Jati, Kapur, and Bipa (from left to right).
5. Conclusion
The wood identification process can be performed by combining the process at mobile-based applications and cloud-based applications at the server side. Both applications are connected through internet facilities. Flask framework is beneficial to facilitate some main capabilities such as the implementation of wood identification based on deep learning algorithms, connecting the device with REST APIs, and providing wood information. An Android-based application, namely AIKO, can offer advantages in the wood identification process compared to manual assessment. The benefits of AIKO, including of practicality of wood identification, speed up the identification process, improvement in terms of identification accuracy, and integration of the collected wood images. In the future, the collected data could be considered as valuable resources for improving identification performance and studying species diversities.

6. Acknowledgments
High-Performance Computing facilities at LIPI is used to develop the classification algorithm and serve identification request from the application users. This research was funded by The Ministry of Research, Technology and Higher Education, The Republic of Indonesia, under Research Grant INSINAS, from 2017 to 2018. The research was a collaborative work between Research Center for Informatics, Indonesian Institute of Sciences (LIPI) and Forest Products Research and Development Center, Ministry of Environment and Forestry, Republic of Indonesia. The authors would like to thank Septian Dwi Indradi, Andria Arisol, Dr Ratih Damayanti, and Listya Mustika Dewi M.Sc. for their contributions in the system development and data collection.

References
[1] F.P.A.D.C. 1992 4000 Jenis Pohon di Indonesia dan Index 4000 Jenis Kayu Indonesia the Ministry of Forest Republic of Indonesia
[2] Tang X J, Tay Y H, Siam N A, Lim S C 2018 Proc. on the 2018 International Conference on Computational Intelligence and Intelligent System pp. 37-43.
[3] Tapu R, Mocanu B, Bursuc A, Zaharia T 2013 Conf. on The IEEE International Conference on Computer Vision (ICCV) Workshop (Sydney) pp. 444-451
[4] Khan Md F F, Akif A, Haque M A 2017 *Conf. on IEEE International Conference on Telecommunications and Photonics (ICTP)* (Dhaka) pp. 33-37

[5] Sari Y A, Adinugroho S 2018 *Conf. on 5th International Conference on Electrical and Electronic Engineering (ICEEE)* (Istanbul) pp. 399-403

[6] Shih Po-Cheng, Tsai Chi-Yi, Hsu Chun-Fei 2017 *Conf. on 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)* (Shanghai) pp.1-5

[7] Marr B 2018 *A Smartphone-Based Obstacle Detection and Classification System for Assisting Visually Impaired People* (online) https://www.forbes.com/sites/bernardmarr/2018/10/01/what-is-deep-learning-ai-a-simple-guide-with-8-practical-examples/#5d3db81d8d4b. [Accessed 1 October 2019]

[8] Lei X, Pan H; Huang X 2019 *A Dilated CNN Model for Image Classification* *IEEE Access* 7 pp. 124087-124095

[9] Aggarwal S 2014 *Flask Framework Cookbook* (Birmingham: Packt Publishing)

[10] Grinberg M 2014 *Learning Android* (Sebastopol: O'Reilly Media)

[11] Doglio F 2015 *Pro REST API Development with Node.js* (La Paz, Canelones: Apress)

[12] Satyaputra A, Aritonang E M 2016 *Let's Build Your Android Apps with Android Studio* (Jakarta: Elex Media Komputindo)

[13] Masruri M H, Creativity J 2015 *Buku Pintar Android* (Jakarta: Elex Media Komputindo)

[14] Gargenta M 2011 *Learning Android* (Sebastopol: O'Reilly)

[15] Raharjo B 2019 *Pemrograman Android Dengan FLUTTER* (Bandung: Informatika Bandung)

[16] Syaputra R Wiraganda Y P 2019 *Happy Flutter: Membuat Aplikasi Android dan iOS Dengan Mudah Menggunakan Flutter* (UDACODING)