kAHOY Mate!: Mobile Application for Wood Identification using Deep Learning to Identify Illegally Logged Wood

Jamie Nicole Castalla*, Isabel Austria†, Jhoanna Rhodette Pedrasa ‡
University of the Philippines Diliman, Quezon City, Philippines
{jamie.nicole.castalla}* {isabel.austria}† @eee.upd.edu.ph, jipedrasa@up.edu.ph‡

***Abstract*—Illegal logging continues to be a major threat to sustainable forestry in the Philippines. One effective way to combat this is by accurately and quickly identifying the different wood species of the confiscated wood. There are various ways to identify the wood species, such as traditional manual inspection, molecular-level research, and machine vision; however, not all of these methods are easily accessible to individuals in remote settings. With the lack of accessible wood identification techniques, kAHOY Mate! was created. This offline-functioning mobile application allows users to upload wood images, with or without an external macro lens, and classify them into four specific wood species and an 'Others' category. Deep learning architectures, specifically ShuffleNet V2 and MobileNet V2, were used to create a model for wood identification, achieving average validation accuracies of 95.57% and 76.03%, respectively. The MobileNet V2 model achieved a test accuracy of 66% and its TensorFlow Lite converted equivalent achieved an accuracy of 71%. The mobile application was developed using TensorFlow Lite and Flutter. This final mobile application achieved an accuracy of 50.50%, with an in-app camera accuracy of 29% and an image upload accuracy of 72%.**

***Index Terms*—ShuffleNet V2, MobileNet V2, wood identification, transfer learning**

I. Introduction

Illegal logging remains to be a significant problem in the Philippines, with the Department of Environment and Natural Resources (DENR) seizing PHP 4.5-million in illegally cut wood in 2021 [1]. Wood identification is crucial for anti-illegal logging efforts, providing a basis for charges against violators and supporting sustainable forestry practices [2]. Effective wood identification relies on accessibility, accuracy, and speed.

In 2024, the Department of Science and Technology-Forest Products Research and Development Institute (DOST-FPRDI) is actively working to improve wood identification in the Philippines [2]. While traditional manual inspection is common, advanced techniques like machine-aided methods and DNA sequencing are largely unavailable and costly in the country [2] [4]. The wood sector, including the DENR, WPPs, and SUCs, continues to seek a straightforward approach to identification [3].

Deep learning models are increasingly applied to wood identification [3], offering a promising solution for accessible and reliable identification, especially in remote, offline settings where field workers lack internet access [3]. Existing mobile applications for wood identification often rely on server-side processing [3]. To address the need for an offline solution, a lightweight, efficient, and accurate model is required. ShuffleNet V2 and MobileNetV2 are promising deep learning architectures for an offline mobile application due to their high accuracy, limited need for computational resources, and compatibility with mobile integration [5].

This work aims to develop an offline-functioning mobile application capable of identifying four distinct wood species (Acacia, Mahogany, Molave, and Narra), representative of illegally-sourced/smuggled wood, along with an 'Others' category for wood species not typically classified as smuggled (Coral Tree, Mulawin, Madre de Cacao, and Pine). The mobile application functions using either the smartphone's built-in camera alone or with an attached clip-on external lens. The wood identification functionality was powered by deep learning, employing a transfer learning approach on the pre-trained MobileNet V2 model utilizing pre-trained weights from ImageNet. This deep learning framework was integrated into the mobile application using TensorFlow Lite and Flutter. A model using ShuffleNet V2 was also made, but was not integrated into the mobile application due to errors in its converted TensorFlow Lite model.

The deep learning model was integrated directly into the mobile application, where all image uploading and processing took place offline. To determine the accuracy of the wood identification and the application's usability, the application underwent integrated and user testing. A qualitative evaluation was also conducted by having multiple individuals interact with the mobile application. This application was developed to identify the species of potentially illegal wood, creating a basis for combating illegal logging.

II. Review of Related Work

The manual inspection of wood requires examining anatomical features [2]. Deep learning, widely used for image recognition, extracts features and trains models to recognize patterns and objects [6]. This technique is applicable to wood identification.

Ergun (2024) compared deep learning architectures ResNet18, GoogLeNet, VGG19, InceptionV3, MobileNetV2, DenseNet20, InceptionResNetV2, EfficientNet, and ShuffleNet V2 using transfer learning for wood identification [5]. Kirbas and Cifci (2022) also applied transfer learning with ResNet-50, Inception V3, Xception, and VGG19 [5]. Transfer learning reuses pre-trained CNN weights to extract features from new datasets, offering a more efficient alternative to training from scratch [5]. ShuffleNet V2 (top, Fig. 1), an improvement

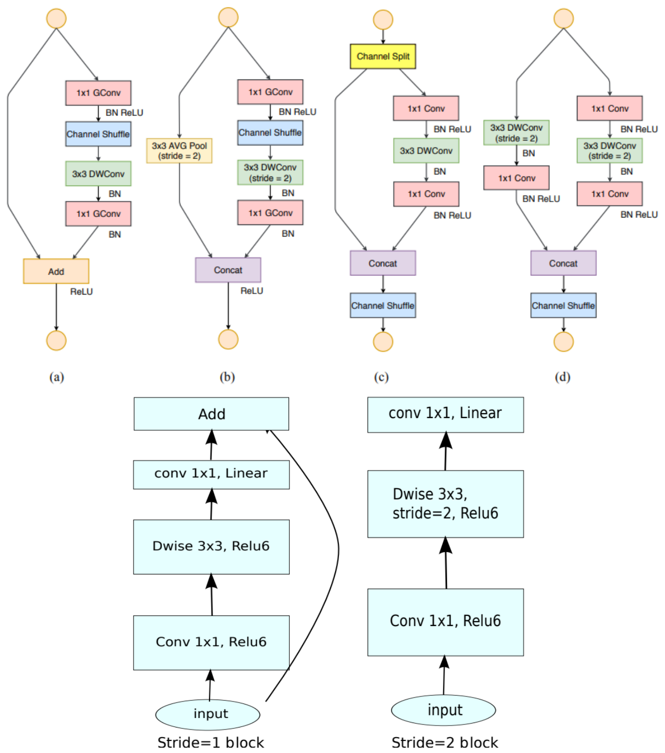

Fig. 1. ShuffleNet V2 and MobileNetV2 Convolution Blocks [6] [8]

on ShuffleNet V1, is accurate and efficient for mobile and embedded systems [6]. Its architecture includes a channel split with depthwise and group convolutions [6] [7]. MobileNet V2 (bottom, Fig. 1) is an efficient mobile architecture using inverted residuals with linear bottlenecks, useful for image classification, object detection, and semantic segmentation, leading to reduced computational cost and memory [8]. Both ShuffleNet V2 and MobileNet V2 are trained on the ImageNet dataset (224x224 input size), a large collection of 1.2 million diverse images well-suited for transfer learning [9].

Multiple applications have implemented the functionality of wood identification using deep learning. A web application capable of identifying 20 species through image uploads has been developed using image analysis and artificial neural networks [3]. For accessibility reasons, a mobile application that processes and identifies different wood species offline is being recommended for future researches [3]. Currently, a mobile application that takes the input image through the application and processes it in a separate server is already being implemented [10]. The two applications, web and mobile, provide more accessibility.

There are many pre-trained deep learning applications that are designed for mobile applications, such as MobileNet and ShuffleNet [5]. A project by Arifin et al. (2020) explored a mobile-based application for wood identification, but this mobile application would only take the input image in through the mobile application where it will be sent to an external server [10]. In the work by Bernante et al. (2024), they created an offline mobile application that uses image processing for edema stage identification; this work integrates edema severity identification using MobileNetV3 into an offline mobile application using TensorFlow Lite and Android Studio [11]. Bernante et al. (2024) demonstrated the feasibility of integrating deep learning models into offline mobile applications. While mobile applications for wood identification rely on server-side processing and offline mobile applications exist in other fields, this project focuses on an offline-functioning mobile application with wood identification capabilities.

When creating a dataset for wood identification, it is common to use macroscopic images of different wood samples. These images are captured using a lens to digitize the wood samples [2]. Additionally, these images are photographs that are magnified with the use of an external macro lens [5]. There are several datasets available that have these types of images. In the study by Ergun (2024), a dataset containing macroscopic tropical wood species images was used to retrain the models [5], this was sourced from Saenz et al. (2022) [12]. Similarly, Kirbas and Cifci (2022) retrained and evaluated the four models they tested using the WOOD-AUTH dataset, which is a collection of 12 Greek wood species that were taken using a Nikon D3300 digital camera with a resolution of 24 megapixels from a distance of 15-20 cm [13]. Another dataset, composed of macroscopic images of the top 20 smuggled wood species, was created using a smartphone and a Xylorix WIDK24X01 external macro-lens with a 24x magnification [3].

III. Methodology

A. Data Collection and Dataset Creation

A dataset of 800 square macroscopic wood images was created, 400 taken with an external clip-on macro lens and 400 with a mobile device's camera alone. An additional test dataset of 100 images with good lighting was compiled, with 20 images per class (10 using the lens, 10 without), except for the "Others" category, which had 12 and 8 images respectively. The images were captured using an iPhone 14 Pro with a 48MP camera, with macro control using a 12MP ultra wide lens on the images without an external lens attached [15]. While the external lens is attached, zoom was used until the circular rim of the lens is no longer in the frame.

Wood samples for Coral Tree and Mulawin (trunk sections) were sourced from El Jardin de Zaida, San Juan, Batangas. Samples of Acacia, Mahogany, Molave, Narra, Madre de Cacao, and Pine (planks) were collected from Sunday's Best, Kalibo, Aklan. Expert consultation from Mr. Christian Pansoy (College of Forestry and Natural Resources, University of the Philippines) was done to learn wood cross-section identifcation.

The model utilizes five classes: Acacia, Mahogany, Molave, Narra, and Others. The "Others" class includes Coral Tree, Mulawin, Madre de Cacao, and Pine. Each class contained 160 images, with forty images per condition, considering both external macro lens use and varied lighting:

- Good Lighting + External macro-lens
- Good Lighting + No External macro-lens (Smartphone camera only)
- Bad Lighting + External macro-lens
- Bad Lighting + No External macro-lens (Smartphone camera only)

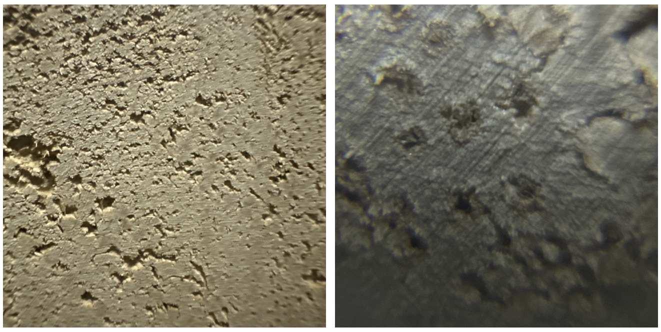

Fig. 2. Acacia Sample Images

Fig. 2 are sample images in the dataset. The image on the left was taken with just the camera of the mobile device with good lighting, and the image on the right was taken with the use of an attached external macro lens with good lighting.

B. ShuffleNet V2 and MobileNet V2 Model Training

The dataset utilized an 80/20 split for training and validation. PyTorch was used to retrain the ShuffleNet V2 architecture and TensorFlow for MobileNet V2 on the dataset. These deep learning frameworks were chosen based on the availability of the architecture within each framework. ShuffleNet V2 is not available on TensorFlow. Transfer learning was used on the ShuffleNet V2 and MobileNet V2 models; the pre-trained weights were utilized in retraining the model on the five classes. The models performed feature extraction by replacing the final fully connected layer to consider the five classes and maintaining the previous layers. Both models were trained for 20 epochs. The training loss, training accuracy, validation loss, validation accuracy, and the confusion matrix were displayed after every epoch. The MobileNet V2 model was also tested on the test dataset.

C. MobileNet V2 Model Integration and Mobile Application Development

To integrate the model into the application, it was converted to TensorFlow Lite. ShuffleNet V2 integration was discontinued due to input shape errors during conversion from PyTorch, as it produced an NCHW format instead of the NHWC required by TensorFlow Lite. MobileNet V2 was trained with TensorFlow for compatibility with TensorFlow

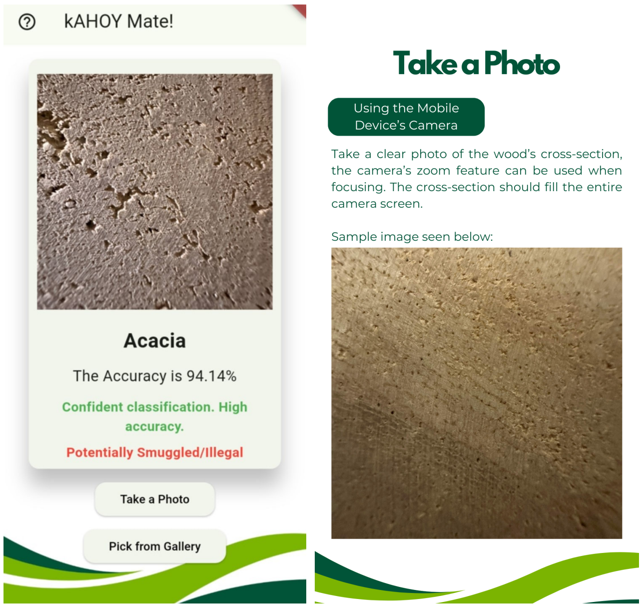

Fig. 3. Mobile Application Homepage and a "How to Use" Slide

Lite conversion; its converted model was then tested on the test dataset.

Using WinsWeb (2023) as a reference, Flutter was used to develop the mobile application, integrate the TensorFlow Lite model, an in-app camera, and an image upload feature [17]. The UI was enhanced using Canva for the "How to Use" image carousel and homepage background. Android Studio served as the IDE and emulator. The homepage (left, Fig. 3) displays the image uploaded, accuracy, confidence, and a warning on being potentially smuggled/illegal after processing.

The "How to Use" image carousel, accessible from the homepage's information icon, provided clear instructions on locating the wood cross-section, using the app's features (in-app camera and image uploading), and capturing images with or without a macro lens. A slide from the "How to Use" carousel is found on the right in Fig. 3.

D. Integrated Mobile Application and User Testing

The mobile application was loaded and ran on an OPPO A16 Android phone with a 13MP camera, and zoom was used in taking images [16].

The application's accuracy and inference time were tested forty times per class (20 with a macro lens and 20 without). This was done by first testing the in-app camera feature using images taken by the phone for 20 images per class (10 with a lens and 10 without) with good lighting, and the second was by using the image upload feature and uploading the images in the test dataset. The results (name of the species, accuracy, confidence, and smuggled warning if applicable) were displayed on the mobile application, while the inference time was displayed on the IDE terminal. The accuracy was computed using the formula below:

$$\text{Accuracy} = \frac{\text{no. of correct classifications}}{\text{total no. of classifications}} \tag{1}$$

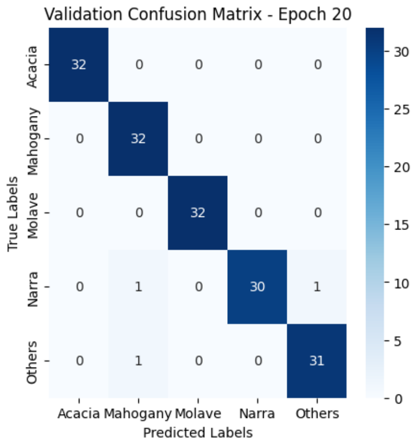

Fig. 4. ShuffleNet V2 Validation Confusion Matrix

Convenience sampling was used to gather responses from ten individuals on the functionality, speed, UI, and overall performance and impact of the mobile application. The respondents ranged in age from 19-57 years old, and they were a diverse group of students and working professionals. They had no prior knowledge of the mobile application or wood classification. They were asked about their demographics, and questions regarding the application's accuracy, usability, performance, UI, experience, and recommendations, as well as open-ended questions.

Before allowing them to explore the mobile application on their own, a brief background of the study, an overview of its purpose, and knowledge of having a "How to Use" section were provided to the respondents. The questionnaire and a consent form, in the form of a Google Form, were given to them to answer after exploring the application and taking their own photos.

IV. Results and Discussion

A. ShuffleNet V2 and MobileNet V2 Model Training

After 20 epochs, the ShuffleNet V2 model achieved an average training accuracy of 94.96%, a validation accuracy of 95.57% after the 20th epoch, and a highest validation accuracy of 99.38% after the 10th and 12th epochs. Values stabilized after the 4th epoch. Fig. 4 shows true vs. predicted labels for the five classes. The model accurately classified wood species, with only three misclassifications, indicating its ability to distinguish between species.

Fig. 5 displays MobileNet V2's training accuracy, validation accuracy, training loss, and validation loss over 20 epochs. Its average training accuracy was 73.01%, validation accuracy after 20 epochs was 76.03%, and the highest validation accuracy was 83.13% at the 16th epoch. Values stabilized after the 8th epoch. The confusion matrix in Fig. 6 shows true vs. predicted labels. While classifying most species correctly, this model had more errors than ShuffleNet V2, misclassifying 31 of 160 validation images. Applying the model to the test dataset yielded 66% accuracy (Fig. 7).

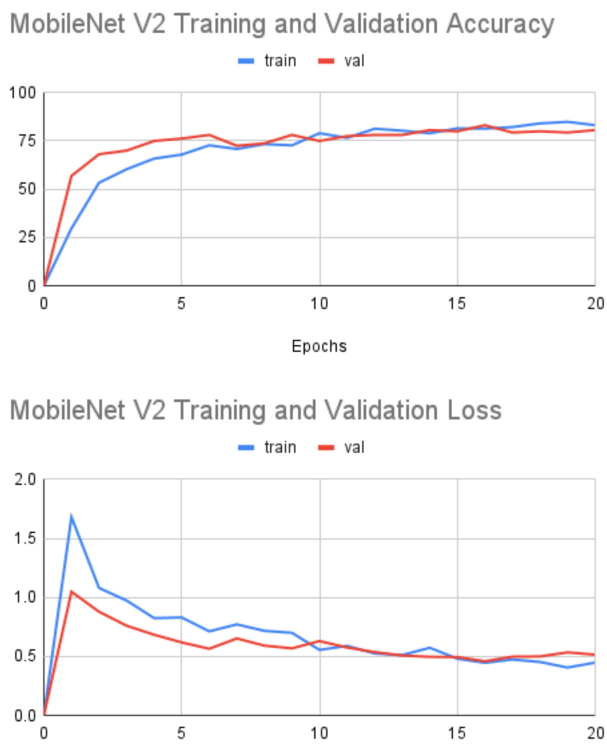

Fig. 5. MobileNet V2 Training and Validation Accuracy and Loss

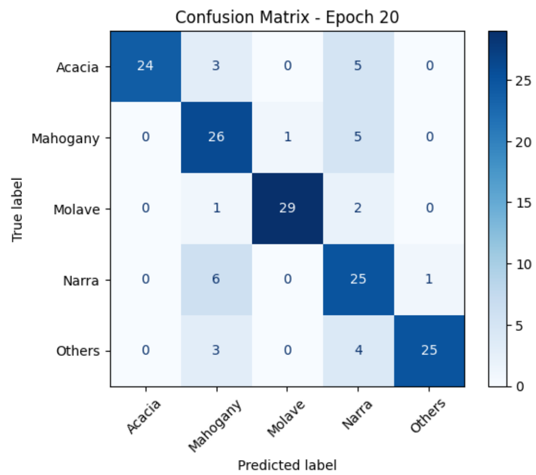

Fig. 6. MobileNet V2 Validation Confusion Matrix

The converted TensorFlow Lite model had an unexpected test dataset accuracy of 71%. This contradicts Bernante (2023) [11], who reported decreased accuracy post-conversion. Quantization, used in TensorFlow Lite conversion to reduce model size and latency, typically risks accuracy [18].

Table I summarizes the performance of ShuffleNet V2, MobileNet V2, and the converted MobileNet V2 TensorFlow Lite model during validation and testing. The average validation accuracy represents each model's performance across 20 epochs during training and validation. The converted MobileNet V2

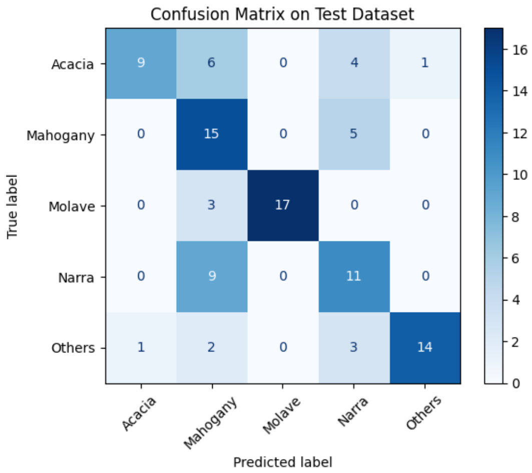

Fig. 7. MobileNet V2 Test Confusion Matrix

TABLE I
MODEL ACCURACY COMPARISON

	ShuffleNet V2	**MobileNet V2**	**MobileNet V2 TFLite**
Average Validation Accuracy	95.57%	76.03%	N/A
Test Accuracy	N/A	66%	71%

TensorFlow Lite model does not have an average validation accuracy, as it did not go through validation and training, it was converted from the MobileNet V2 model. ShuffleNet V2 achieved a higher average validation accuracy compared to MobileNet V2; this aligns with ShuffleNet V2's previously demonstrated accuracy over MobileNet V2 on the ImageNet dataset [7]. Test accuracy indicates the models' performance against the test dataset. This dataset was compiled after the development of the mobile application, which utilized the MobileNet V2 TFLite model, as a result, only the MobileNet V2 model and the MobileNet V2 TFLite model were evaluated against it. All models were run on a desktop environment using Google Colab.

B. *Integrated Testing of the Mobile Application*

Table II displays the mobile application's accuracy and average inference times (in parenthesis) through its in-app camera and image upload features. Using images from the test dataset, the image upload feature achieved 72% accuracy, while the in-app camera yielded 29% accuracy. The overall application accuracy was 50.5%.

It is noted that the test dataset images were captured with a high-quality camera (48MP) with macro control (12MP), while the application ran on a device with a lower-quality camera (13MP) using zoom without macro control, leading to a difference in image quality. Thus, image quality significantly impacted classification, a common issue in machine vision where models trained on high-quality images often fail with real-world, lower-quality inputs [19]. The work [19] studies the effect of different quality distortions; the distortion blur can be a result of improperly focused cameras and can simulate images captured with a low resolution, it removes the textures in images. The blur distortion in images reduced the classification accuracy of the neural networks tested; they are all susceptible to blur distortions [19]. This problem directly affected the mobile application, leading to incorrect classifications due to the model's inability to account for lower-quality images, when the wood images' textures are removed or not captured through the lower resolution camera. The dataset's limitation in image quality led to a decline in the application's classification performance.

During testing, the lowest inference accuracy displayed in the application was 32.8%, and the highest was 100%. In both scenarios (with and without a lens), processing an uploaded image was faster than processing an image from the in-app camera. The overall average inference time during testing was 398.55 ms, which is not a noticeable delay within the application. The average inference times for each class per condition were close in value.

C. *User Testing*

Most respondents (8 out of 10) found the application somewhat accurate, with one finding it accurate most of the time and one deeming it inaccurate. This aligns with the varying accuracy across wood species during integrated testing.

The user interface (UI/UX) was praised for its clarity, visual appeal, and understandability, attributed to its simple and direct functionalities. Instructions within the app were reported as very clear by half of the respondents, despite a learning curve associated with identifying wood cross-sections.

The in-app camera feature was largely considered very easy to use by most (6 out of 10), with difficulties likely stemming from the image-taking process itself rather than the app's functionality. Uploading images from the gallery was almost universally described as very easy (9 out of 10). The application also performed well in terms of speed, with results generated very fast according to the majority of users (8 out of 10), consistent with the inference time results.

Respondents generally rated their overall experience as good, highlighting the app's usability but noting that accuracy needs improvement. The potential for recommending the application was high, with respondents being very likely, moderately likely, and extremely likely to recommend it, indicating its perceived value once accuracy is enhanced.

Key feedback included the app's clarity, efficiency, speed, offline performance, and its recognized potential beyond wood identification. However, accuracy remains the primary concern, particularly with species like Molave. Improvement suggestions include recalibrating the application with more diverse devices and conditions, and enhancing user understanding by adding video instructions and Filipino language support for the target audience.

TABLE II
ACCURACY AND AVERAGE INFERENCE TIME (IN PARENTHESIS) OF EACH CLASS PREDICTION UNDER THE CONDITIONS OF USING THE IN-APP CAMERA OR IMAGE UPLOAD FEATURE, AND THE USE OF THE EXTERNAL LENS

Classes	**In-app camera, % (ms)**		**Image upload, %(ms)**	
	With lens	Without lens	With lens	Without lens
Acacia	0 (460.5)	0 (509)	30 (291.4)	50 (308.9)
Mahogany	0 (456.6)	0 (515.2)	50 (297.1)	100 (322.7)
Molave	90 (466.1)	70 (480.5)	80 (288.1)	100 (310.9)
Narra	0 (477.3)	40 (511.8)	100 (291.3)	60 (313.8)
Others	30 (511.2)	60 (536.63)	62.5 (292.88)	83.33 (303.85)

V. CONCLUSIONS AND RECOMMENDATIONS

This project aimed to create an offline mobile application for wood identification. A dataset of 800 images across 5 categories (Acacia, Mahogany, Molave, Narra, Others) and an additional 100-image test dataset were created. Two models were developed: ShuffleNet V2, with an average validation accuracy of 95.57%, and MobileNet V2, with an average validation accuracy of 76.03% and a test accuracy of 66%. Only MobileNet V2 was integrated into the mobile application and converted to TensorFlow Lite, achieving 71% accuracy. The mobile application's overall wood classification accuracy was 50.50%, with 29% using the in-app camera and 72% using the image upload feature. The accuracy difference between the in-app camera and image upload is attributed to the input image quality. Future work could involve additional pre-processing techniques and retraining the model with lower-resolution images to improve on this.

Due to ShuffleNet V2's high accuracy and its incompatibility as a TensorFlow Lite model due to input shape errors, the proponent recommends finding another method to incorporate this PyTorch model into a mobile application through different conversion tools or Lite models that can also be integrated into a mobile application. To improve user understanding, an instructional video can be added. Lastly, given its use in the Philippines, future works can consider language options like Filipino for easier application understandability. This mobile application through its wood identification capability, can be a crucial tool in strengthening the anti-illegal logging campaign.

DECLARATION OF AI USE

This work utilized generative artificial intelligence (AI), Gemini and ChatGPT, to create and edit the code used in the models, integrate the model into the mobile application, and create and edit the code used in developing the mobile application.

REFERENCES

[1] "DENR-led task force seizes P4.5-M worth of hot logs," Department of Environment and Natural Resources, Oct. 2019. [Online]. Available: https://denr.gov.ph/news-events/denr-led-task-force-seizes-p4-5-m-worth-of-hot-logs/.

[2] "DOST-FPRDI moves to boost wood identification in the Philippines," Department of Science, Technology-Forest Products Research, and Development Institute, Aug. 2024. [Online]. Available: https://fprdi.dost.gov.ph/index.php/563-dost-fprdi-moves-to-boost-wood-identification-in-the-philippines.

[3] R. Mendoza et al., "Automated classification of selected Philippine wood species using image analysis and artificial neural networks", The Philippine journal of science, vol. 151, no. 4, Jul. 2022. DOI: 10.56899/151.04.11. [Online]. Available: https://doi.org/10.56899/151.04.11.

[4] E. E. Dormontt et al., "Forensic timber identification: It's time to integrate disciplines to combat illegal logging," Biological Conservation, vol. 191, pp. 790–798, Nov. 2015, doi: https://doi.org/10.1016/j.biocon.2015.06.038.

[5] H. Ergun, "Wood identification based on macroscopic images using deep and transfer learning approaches", PeerJ, vol. 12, e17021, Feb. 2024. DOI: 10.7717/peerj.17021. [Online]. Available: https://doi.org/10.7717/peerj.17021.

[6] Y. Huang, "Deep learning in image recognition", Applied and Computational Engineering, vol. 8, pp. 61–67, Aug. 2023. DOI: 10.54254/2755-2721/8/20230082.

[7] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "ShuffleNet V2: Practical Guidelines for Efficient CNN Archi- tecture Design", arXiv (Cornell University), Jan. 2018. DOI: 10.48550/arxiv.1807.11164. [Online]. Available: https://arxiv.org/abs/1807.11164.

[8] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," Mar. 2019. Available: https://arxiv.org/pdf/1801.04381v4.

[9] M. Huh, P. Agrawal, and A. A. Efros, "What makes ImageNet good for transfer learning?" arXiv, Jan. 2016. [Online]. Available: https://arxiv.org/abs/1608.08614. DOI: 10.48550/arxiv.1608.08614.

[10] M. Arifin, B. Sugiarto, R. Wardoyo, and Y. Rianto, "Development of mobile-based application for practical wood identification," IOP Conf. Ser.: Earth Environ. Sci., vol. 572, no. 1, p. 012040, 2020, doi: 10.1088/1755-1315/572/1/012040.

[11] A. Bernante, K. Recto, and J. Pedrasa, "Mobile application for remote monitoring of peripheral edema", in Springer, Jun. 2024, pp. 259–273, ISBN: 978-3-031-59716-9. DOI: 10.1007/978-3-031-59717-6_18.

[12] D. A. C. Saenz, C. F. O. Urbano, H. R. G. Mesa, and R. Vargas-Canas, "Tropical Wood Species Recognition: A Dataset of Macroscopic images", Data, vol. 7, no. 8, p. 111, Aug. 2022. DOI: 10.3390/data7080111.[Online]. Available: https://doi.org/10.3390/data7080111.

[13] I. Kirbas and A. Çifci, "An effective and fast solution for classification of wood species: A deep transfer learning approach," Ecol. Inform., vol. 69, p. 101633, 2022, doi: 10.1016/j.ecoinf.2022.101633.

[14] F. Reboredo, "Socio-economic, environmental, and governance impacts of illegal logging", Environment Sys- tems Decisions, vol. 33, no. 2, pp. 295–304, May 2013. DOI: 10.1007/s10669-013-9444-7. [Online]. Available: https://doi.org/10.1007/s10669-013-9444-7.

[15] "iPhone 14 Pro - Technical Specifications - Apple Support (PH)," Apple Support, 2022. https://support.apple.com/en-ph/111849.

[16] "OPPO A16 Specs," OPPO. [Online]. Available: https://www.oppo.com/ph/smartphones/series-a/a16/specs/.

[17] WinsWeb, "Image classification TfLite mango disease detection flutter," YouTube, Dec. 25, 2023.

[18] "Post-training quantization," Google AI for Developers. https://ai.google.dev/edge/litert/models/post_training_quantization.

[19] S. Dodge and L. Karam, "Understanding how image quality affects deep neural networks," 2016 Eighth International Conference on Quality of Multimedia Experience (QoMEX), Lisbon, Portugal, 2016, pp. 1-6, doi: 10.1109/QoMEX.2016.7498955.