

Identification of Peruvian timber species based on image processing and convolutional neural networks

Alejandro Blanco¹, José Sánchez¹, Christian del Carpio²,
Erwin Dianderas³, Guillermo Kemper^{1*}

¹ Faculty of Engineering, School of Electronic Engineering, Universidad Peruana de Ciencias Aplicadas, Lima 15023, Peru

² Laboratorio de Investigación en Inteligencia Artificial, Robótica y Procesamiento de Imágenes, Universidad Nacional de Ingeniería, Lima 15333, Peru

³ BOSQUES, Instituto de Investigaciones de la Amazonía Peruana, Iquitos 16001, Peru

ABSTRACT

The identification of timber species is highly important worldwide due to their global trade and the arising requirement of controlling, regulating and monitoring illegal logging. Peru has an important timber trade due to extensive forest resources in the Peruvian Amazon. This context makes it necessary to have technological tools for timber species identification using tree leaves, in an attempt to control deforestation in the Amazon forests. Current identification methods require manual labor, are subjective and inefficient regarding time, cost and precision. Moreover, the scientific literature does not include electronic devices nor algorithms which exclusively identify Peruvian timber species. With this in mind, this work proposes a portable electronic equipment that detects 11 timber species from the Peruvian jungle, based on image processing techniques and convolutional neural networks (CNN). The device consists of a carbon fiber structure, a keyboard, a 7-inch display, internal lighting and a tray for the leaf sample. The algorithm uses color filters, thresholding and other operations to segment the leaf and then input it to a neural network. After comparing the AlexNet, VGG-16 and MobileNet architectures, the last one shows the best performance, with an average precision of 98.64% when identifying the 11 timber species.

Keywords: Portable device, Timber species, Image processing, Convolutional neural networks (CNN), Species identification.

OPEN ACCESS

Received: July 24, 2023


Revised: September 6, 2023

Accepted: September 18, 2023

Corresponding Author:

Guillermo Kemper

guillermo.kemper@upc.pe

 **Copyright:** The Author(s).

This is an open access article distributed under the terms of the [Creative Commons Attribution License \(CC BY 4.0\)](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted distribution provided the original author and source are cited.

Publisher:

[Chaoyang University of Technology](https://www.chaoyang.edu.cn/)

ISSN: 1727-2394 (Print)

ISSN: 1727-7841 (Online)

1. INTRODUCTION

The forestry industry is highly relevant to modern global economics. According to the International Tropical Timber Organization (ITTO), in 2020, global trade exchanged more than 2.5 trillions of cubic meters of timber (Ferreira et al., 2021). On the other hand, illegal logging is a growing issue which harms producers and consumers worldwide such as: Australia, Cambodia, China, the European Union, Indonesia and Peru (Winkel et al., 2017). Both timber global trade as well as illegal logging show the need to regulate forests and inspect the origin of commercialized timber in order to protect species diversity. In the case of Peru, SERFOR (the National Forestry and Wildlife Service) compels owners of forest areas to register them in the national registry of forest plantations, indicating the number of trees and species in the owned area for commercial purposes. A botanist identifies the timber species, firstly through visual means and, if required, through laboratory analysis using a dendrological manual. This manual process is time-consuming, costly and has low precision, so an automatic identification method would significantly reduce these issues and allow timber companies and research

institutions to precisely monitor forests for an appropriate timber commercialization or for elaborating species protection plans. State-of-the-art solutions use algorithms with Support Vector Machines (SVM) or CNN which work with mobile applications, portable devices or graphical interfaces but do not solve the identified problems because of: high error rates, using public databases which do not have Peruvian timber species or comparing easily

differentiable plant species, instead of comparing timber species.

Table 1 shows these proposals, as well as their strengths and weaknesses. As Table 1 shows, none of the previous works detect Peruvian timber species, but in-stead use public datasets or build their own datasets with fruit, medicinal or non-Peruvian plants. This is a problem since SERFOR requires

Table 1. Proposed solutions in the state-of-the-art

Reference	Technique	Application	Strengths	Weaknesses
(Selda et al., 2017)	Uses SIFT (Scale-Invariant Feature Transform) to obtain the leaf vein structure and SVM to classify the SIFT features.	Portable equipment with a Raspberry Pi to identify plants	Integrates the SIFT and SVM algorithm in a portable device, adding a graphical user interface.	High error rate of 15.71%
(Singh et al., 2020)	Image pre-processing via noise reduction, border enhancing, binary thresholds and Fourier descriptors. Uses a VGG-16-based CNN for classification.	Method for leaf recognition.	High algorithm precision (95%). The algorithm can recognize leaves damaged up to a 30% of their shape.	The algorithm is trained and tested with the Flavia public database, which has plant leaves and no timber species.
(Monroy-de-Jesús et al., 2019)	Uses a CNN in Python trained with 20 epochs and 1000 iterations per epoch.	Algorithm which recognizes medicinal plants in Jocotitlan, Mexico.	The algorithm identifies 11 plant species.	Training images show notable differences in their flowers, structure and color. The achieved precision is 84.07%
(Song et al., 2019)	Uses an ABCNN (Attention-Based Convolutional Neural Network), trained with the Leafsnap database. Image pre-processing via gamma transform, geometric transforms, smoothing filters and noise for data augmentation	Method to recognize similar tree leaves.	High precision of 98.27%	Leafsnap only contains images from the Northwest of the United States.
(Srivastava and Khunteta, 2018)	Extracts 14 leaf image features: leaf area, eccentricity, orientation, circularity, perimeter, among others. Compares the performance of 3 classifiers: quadratic SVM, cubic SVM and medium Gaussian SVM.	Classification algorithm for 16 plant species in the Flavia database.	Method can identify 16 different species, which is plenty.	The achieved error rates are slightly high: 9.1% for the quadratic SVM, 10.6% for the cubic SVM, 10.2% for the medium Gaussian SVM.
(Zarrin and Islam, 2019)	Uses a Redmi 5A smartphone for leaf image acquisition and authors propose their own 9-layer CNN.	Leaf image classifier based on a CNN.	Algorithm achieves a precision of 99.4% for the 10 species.	All species are fruit trees and none are timber species. The algorithm has no electronic equipment.
(Kang and Oh, 2018)	Compares the following CNN models: AlexNet, VGG, GoogleNet; integrates the deep learning model with a mobile application which asks the uses to center the leaf with the phone camera.	Android application for plant classification in real time.	Achieves a 96.0% accuracy with VGG for 63 species.	The plant species in the dataset are from a field in a Korean university.
(Varghese et al., 2020)	Uses transfer learning with a MobileNet model connected to a mobile application through Firebase.	Android application for plant classification in real time.	Achieves a validation precision of 95%.	The training database consists of only 6 different medicinal plants from India.

registering Peruvian timber species mainly targeted by illegal loggers. Moreover, some works have high error rates, which could translate into financial penalties due to incorrectly registering misidentified species. Additionally, most proposals use mobile applications instead of their own portable devices for in situ image acquisition. Since the forest plantations are located deep in the Amazon rainforest, the use of an autonomous, portable device has two main advantages. Firstly, analyzing the samples in situ minimizes the need for taking the samples back to a stationary laboratory, which is time consuming. Secondly, performing the analysis in the device eliminates the requirement of a stable Internet connection, which is not available in the plantation locations. This work presents the design of a portable equipment which works jointly with the image pre-processing algorithms and the CNN model to classify 11 timber species, which were chosen because of their vulnerability to illegal logging and their ecological importance to the Peruvian amazon rainforest.

The main contributions of this work are:

1. A portable device for identifying 11 Peruvian timber species, consisting of a waterproof structure, a keyboard, an LCD screen, a single-board computer and image processing and artificial intelligence algorithms based on a CNN which achieve an average precision of 98.64%.
2. A database with images from the 11 Peruvian timber species acquired with con-trolled lighting using the portable device. The latter guarantees that the images were acquired with a good quality to maximize the performance of the proposed algorithm.

The following sections describe the portable device, algorithms and results.

2. MATERIALS AND METHODS

This section details the materials and methods applied in both the hardware (electronics and mechanical structure of the device) and software (image processing and artificial

intelligence algorithms) aspects. The chosen materials and methods are the result of a selection process which considered several experimental tests, application scenarios, ease of operation and transportation, energy autonomy, computational efficiency, species detection performance and engineering fundamentals.

The mechanical enclosure for the device is designed to be 3D printed in PETG filament. The structure is then reinforced with fiberglass and covered by a waterproof backpack to improve its portability and fall resistance. This design results in a structure which minimizes the impact of external light sources and shadow artifacts during the image acquisition. Thus, the device does not require additional image pre-processing such as the use of color format conversion (Ozturk and Akdemir, 2017) or thermal capture methods (Soetedjo and Hendrianti, 2022), which generate high use of computational load.

2.1 Electronics and Structure of the Proposed Device

The electronics of the proposed device are made up of 4 functional parts: image acquisition, data processing, data visualization and power supply. Fig. 1 shows the pictorial diagram of the indicated parts.

The image acquisition subsystem consists of a Raspberry Pi camera using a resolution of 12.3M pixels (4056×3040 pixels) for still images and 1920×1080 for video (Raspberry Pi, 2020). These parameters enable an adequate image quality for a correct performance of the species identification algorithm due to a high image acquisition resolution which captures most of the details in the leaves while using a camera model compatible with the Raspberry Pi. Additionally, this camera may use different lenses to enhance its field of view (FOV).

The chosen camera guarantees compatibility with the Raspberry Pi computer and with C or SC type lenses and connects to the Pi with a special flat cable. The lens compatibility enables the possibility of using Arducam models, which have a variety of focal lengths to modify the

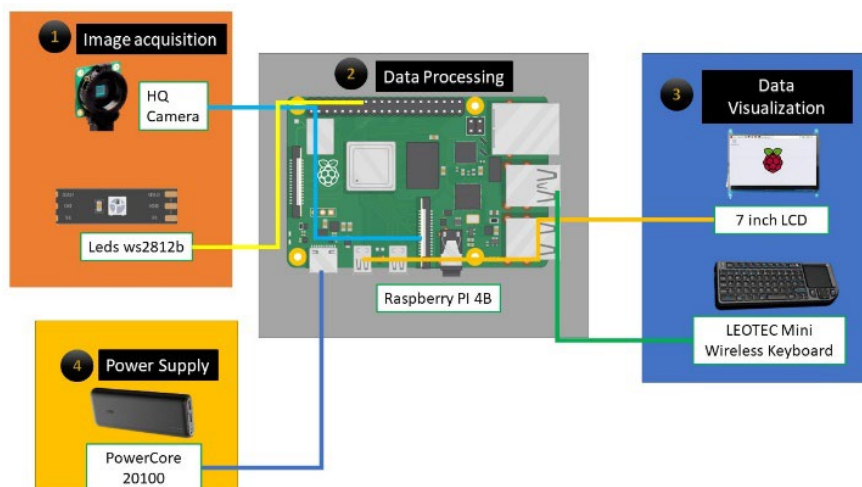


Fig. 1. Pictorial diagram of the device

desired FOV. The device also has two RGB LED strips with configurable intensity and color, useful for keeping the lighting constant throughout all captured images and allowing the segmentation algorithm to function properly.

A small single-board computer Raspberry Pi 4B performs the data processing and equipment control. This computer has enough resources to execute the image processing and artificial intelligence algorithms quickly, demonstrated by various applications such as gesture recognition, face detection, facial biometrics, etc. (Chen et al., 2021). Moreover, it has a lower cost and power consumption than other single-board computer alternatives. Some features of the 4B version are: 8GB RAM, 1.5 GHz CPU, 802.11ac and Bluetooth 5.0 wireless connectivity (electronicaplugandplay, n.d.). These features allow for a fast image processing pipeline and quick leaf classification. Additionally, all the device's peripherals are directly connected to the Pi since it has specific ports for each of them.

On the other hand, a 7-inch LCD screen shows the graphic user interface, through which users can operate the device as well as visualize information. The display has a resolution of 1024×600 pixels and connects to the Raspberry Pi via a micro-HDMI port. A Mini Leotec wireless keyboard serves to input data through the user interface, so that users have a comfortable way to interact with the device features, such as registering new plant species. The keyboard has a touch pad for users to navigate through the developed GUI and the device's operating system if needed.

The power supply consists of a 20100 mAh PowerCore. The energy consumption requirements are detailed in Fig. 2.

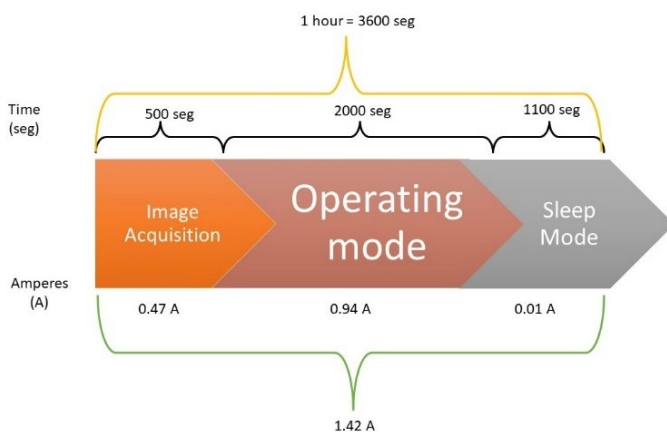


Fig. 2. Time diagram of estimated consumption for 1 h

The autonomy time is evaluated with Equation (1):

$$B_{cap} = t \times D \tag{1}$$

Where:

- t : autonomy time (h)
- B_{cap} : battery capacity(Ah)
- D : system consumption(A)

For 8 h of autonomy and a consumption per hour of 1.42 A, the required battery capacity is:

$$B_{cap} = t \times D = 8 \times 1.42 = 11.36 Ah$$

Then, the real capacity of the powerbank is calculated using Equation (2).

$$Cap_{real} = B_{cap} \times \frac{3.7}{5} \times E_{en} \tag{2}$$

Where:

B_{cap} : Battery capacity (Ah)

E_{en} : Energy efficiency (%)

A 90% power bank efficiency is a reasonable assumption, so replacing in the data we have:

$$Cap_{real} = 20.1 \times \frac{3.7}{5} \times 0.9 = 13.38Ah$$

Therefore, the PowerBank 20100 ensures the operational autonomy of 8 h of continuous execution of the species identification algorithm.

Regarding the structure of the equipment (see Fig. 3), this was manufactured with carbon fiber due to its resistance to elevated temperatures such as those in the Peruvian rainforest. Additionally, this material has a low density and high flexibility which enables improved ergonomics and aesthetics. The device has a total weight of 7.2 kg and is equipped with external waterproof plastic casing for adequate protection and transportation. The plastic casing allows the equipment to be carried out easily like a backpack, which is extremely helpful when identifying species deep inside the Peruvian Amazon, where walking is required to extract the leaf of said species.

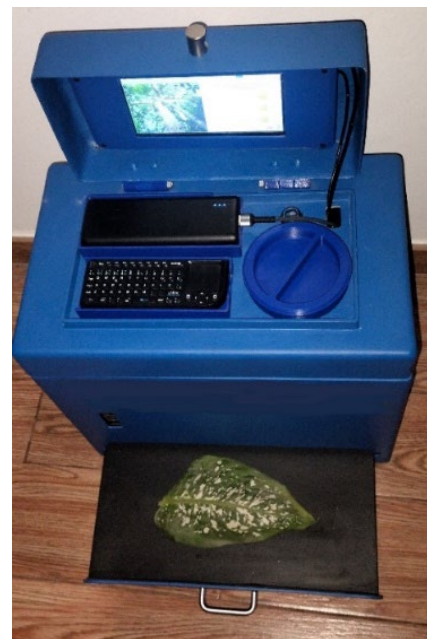


Fig. 3. Developed device

Fig. 4 shows some main parts of the equipment such as the power supply, keyboard, screen and leaves tray. Each measurement shown in Fig. 5 is justified. In the first place, the width, length and depth of the screen are designed to prevent liquids from entering and also to create a shadow so that the screen can be viewed even in bright sunlight. Secondly, the thickness of the electronics area generates greater ventilation for the electronic boards inside and prevents overheating. Third, the diameter of the hole in the upper part provides access to the components area and the image acquisition area in case an adjustment is needed, especially regarding the lens calibration. Lastly, the width of the image acquisition area ensures that leaves up to 30 cm long can be captured without losing image quality. Finally, Fig. 6 shows in detail the internal electronic components and the main areas of the developed device.

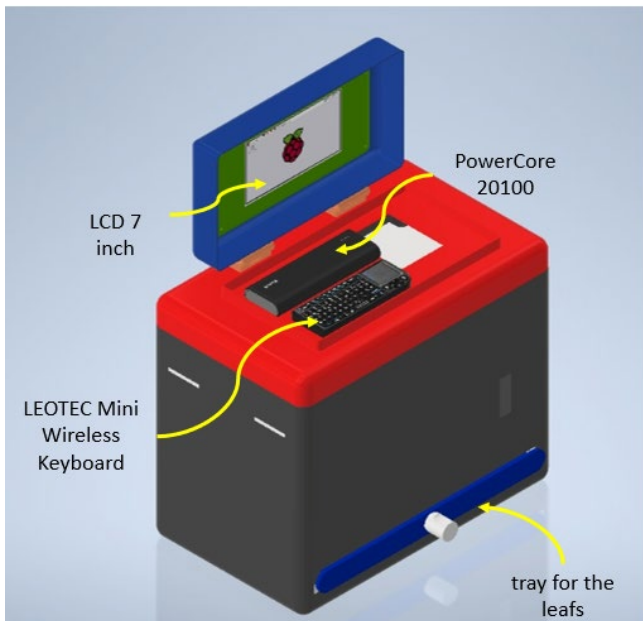


Fig. 4. External view of render of the device

2.2 Image Processing and Artificial Intelligence Algorithms

This section will detail the methods used for the image processing and artificial intelligence algorithms. Since the equipment has uniform acquisition conditions, the segmentation process is simplified and is performed with simple methods that reduce the computational load and allow to use greater resources for the leaf classification via artificial intelligence. Fig. 7 illustrates the block diagram of the proposed algorithm. As seen in Fig. 7, the input to the algorithm is the leaf image acquired under controlled lighting in the portable equipment. The first step consists of background elimination, using image processing techniques such as color filter, binary thresholds, morphological operations and border detection techniques. If the timber species has been previously registered, an artificial

intelligence identifies it. If not, the segmented leaf image is stored in a database such that, when enough images are available, the neural network model can be trained again. The following section describes each step in the algorithm.

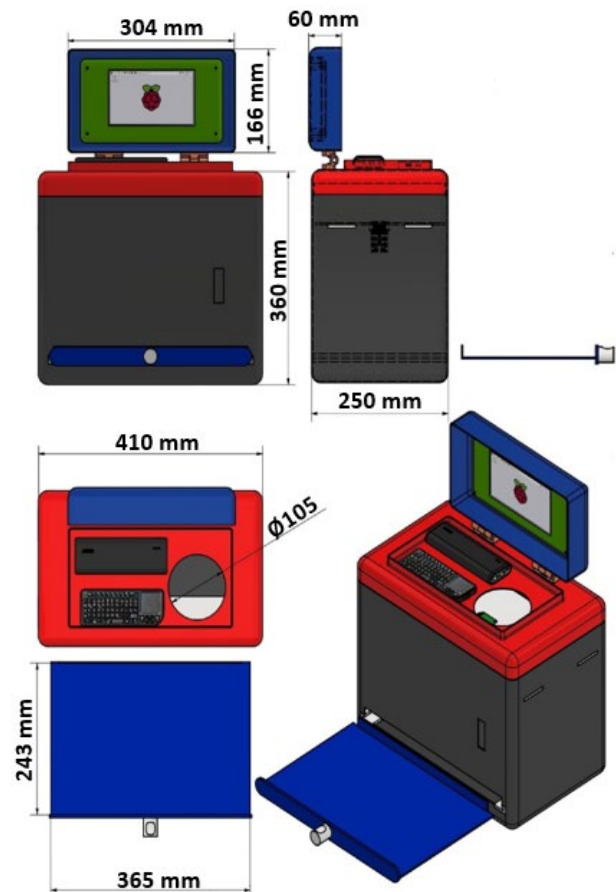


Fig. 5. Detailed views of the developed device, dimensions in millimeters

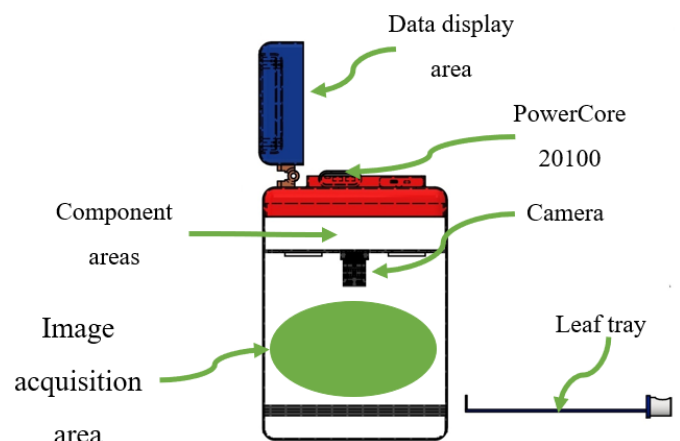


Fig. 6. Main areas of the developed device

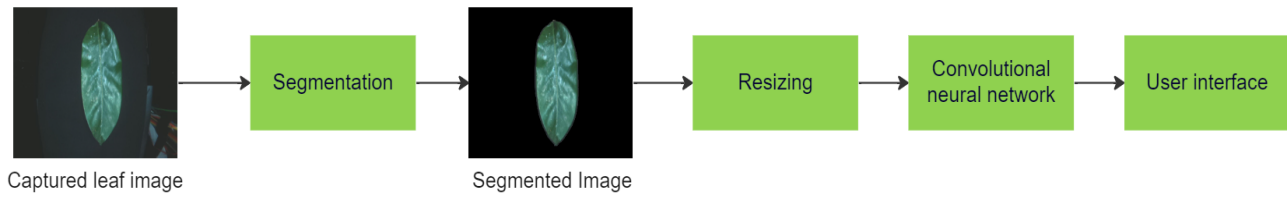


Fig. 7. Block diagram of the proposed algorithm

2.2.1 Image Acquisition

The input sample consists of the leaf of a tree from the Amazon rainforest, which must be in a good condition, without deformations and flattened out to avoid processing artifacts due to shadows. The device's dimensions are $410 \times 342 \times 250$ mm and the leaf tray has a size of 365×243 mm, which is the maximum leaf size. Acquired images are 24 bits RGB (true color) with primary components $I_R(x, y)$, $I_G(x, y)$ and $I_B(x, y)$. Here, (x, y) are the spatial coordinates of the image, where $x = 0, 1, \dots, M-1$ and $y = 0, 1, \dots, N-1$, for $M = 1080$ and $N = 1920$.

Considering the device dimensions, the following procedure determines the distance between the surface which holds the leaf and the acquisition camera.

Step 1: The chosen image resolution follows the criterion of achieving a performance of at least 90% in the identification of species, as well as a computational load where the algorithm execution time does not exceed 20 seconds. Empirical testing shows that a resolution of $N = 1920$ columns by $M = 1080$ rows is appropriate for image acquisition.

Step 2: Using the image resolution from step 1, the sensor dimensions are computed. The Raspberry Pi camera has a IMX477R sensor with square pixels of $L_p = 1.55 \mu\text{m}$, so Equations (3) and (4) show the sensor size:

$$L_{s_x} = L_p \times M = 1.55 \mu\text{m} \times 1080 = 1.674 \text{ mm} \quad (3)$$

$$L_{s_y} = L_p \times N = 1.55 \mu\text{m} \times 1920 = 2.976 \text{ mm} \quad (4)$$

Where:

L_{s_x} : Sensor's vertical dimension (mm)

L_{s_y} : Sensor's horizontal dimension (mm)

Step 3: The camera's focal distance serves to calculate the vertical α_x and horizontal α_y viewing angles, through Equations (5) and (6). This calculation uses the shortest focal distance in the multifocal camera lens, $d_f = 2.8$ mm.

$$\alpha_x = 2 \times \tan^{-1} \left(\frac{L_{s_x}}{2d_f} \right) = 2 \times \tan^{-1} \left(\frac{1.674 \text{ mm}}{2 \times 2.8 \text{ mm}} \right) = 33.29^\circ \quad (5)$$

$$\alpha_y = 2 \times \tan^{-1} \left(\frac{L_{s_y}}{2d_f} \right) = 2 \times \tan^{-1} \left(\frac{2.976 \text{ mm}}{2 \times 2.8 \text{ mm}} \right) = 55.97^\circ \quad (6)$$

Step 4: Finally, Equation (7) shows the calculation of the distance D_{cs} between the camera and the acquisition surface.

This calculation considers a rectangular image acquisition area with a length $L_{a_y} = 30$ cm.

$$D_{cs} = \frac{L_{a_y}}{2 \times \tan\left(\frac{\alpha_y}{2}\right)} = \frac{30 \text{ cm}}{2 \times \tan\left(\frac{55.97^\circ}{2}\right)} = 28.2 \text{ cm} \quad (7)$$

Then, Equation (8) computes the rectangle's width L_{a_x} :

$$L_{a_x} = 2 \times \tan\left(\frac{\alpha_x}{2}\right) \times D_{cs} = 2 \times \tan\left(\frac{33.29^\circ}{2}\right) \times 28.2 = 16.9 \text{ cm} \quad (8)$$

Fig. 8 illustrates the dimensions of the acquisition area (FOV) and the distance between the camera and the acquisition surface.

2.2.2 Dataset

This section describes the two datasets used for the development of the leaf classification algorithm. The first one is used to classify a leaf from 11 possible species, while the second one is used to determine whether a leaf belongs to one of the species in the first dataset. Details of each of the datasets are given below.

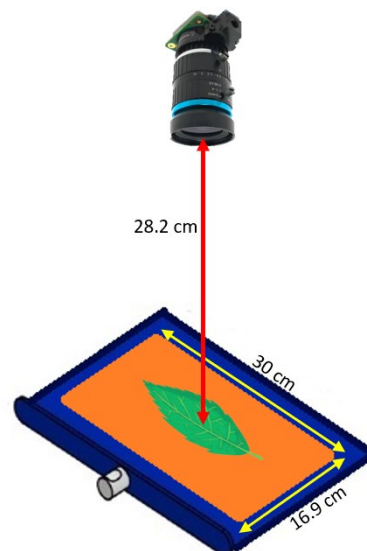


Fig. 8. Image acquisition area and working distance.

2.2.2.1 Species Classification

The species classification dataset consists of images of leaves from 11 timber species from the Peruvian amazon

rainforest, chosen due to their vulnerability to illegal logging and their economic and ecological importance. These species are among the most commercialized ones for timber production in Peru. The 11 selected species are accessible within a particular region of the Peruvian Amazon Rainforest. The developed device took pictures of the collected leaves, resulting in 100 images for each of 10 of the 11 species, and 120 images for the remaining one, since a collecting mission deep in the Peruvian Amazon was conducted and that was the number of available leaves. The application of data augmentation helps in building a more diverse dataset to avoid model overfitting. The augmentation transforms are random rotations in the range $[-20^\circ, 20^\circ]$ and random horizontal flipping. Rotations provide better results than other geometric transformations in diverse augmentation tasks (Maharana et al., 2022), while flipping augmentation is necessary to recognize leaves wrongly inserted into the device. Fig. 9 illustrates the block diagram of the data augmentation process.

The final dataset after data augmentation consists of 13308 images, where 69.70% are used for training and 30.3%, for testing. Table 2 indicates the species distribution in the training and testing datasets, and Table 3 shows examples of the 11 species in the dataset.

2.2.2.2 Identification of Whether a Species belongs to the Built Dataset

This section describes an additional dataset used to train a neural network which distinguishes whether the leaf put in the device belongs to the species identification dataset. The additional dataset consists of some images from the 11 selected timber species, grouped in a “Known” class, and new images grouped in an “Unknown” class. The latter class consists of images from 8 fruit species, shown in Fig. 10, as well as images from the timber species with manual modifications such as shape deformations, cropped apex/bottom ends, margin undulations, etc. After building a balanced new dataset (50% of images in each class), it is

enhanced using data augmentation, resulting in 9865 images split in 69.34% for training and 30.66% for testing. Table 4 shows the final data distribution.

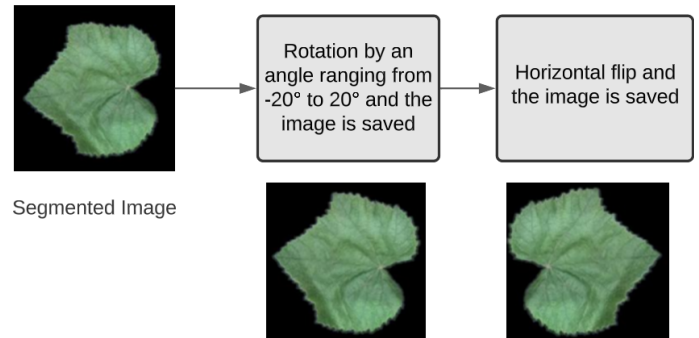


Fig. 9. Block diagram of the data augmentation process

2.2.3 Leaf Segmentation

The following computational procedure results in the leaf segmentation and elimination of unwanted objects and regions which could hamper the classification stage:












Step 1: A color filter extracts most of the leaf section from the background. The original image with primary components $I_R(x,y)$, $I_G(x,y)$ and $I_B(x,y)$ is converted to the HSV (Hue-Saturation-Value) color model (Gonzales and Woods, 2018), where $H(x,y)$ is the hue, $S(x,y)$ is the saturation and $V(x,y)$ is the value component. An extensive evaluation of the histograms from the 3 components in plenty images with only background resulted in the thresholding mask $Mas(x,y)$ in Equation (9) with thresholds illustrated in Fig. 11.

$$Mas(x,y) = \begin{cases} 0 & , 85 < H(x,y) < 100 \wedge \\ & S(x,y) < 52 \wedge \\ & V(x,y) < 50 \\ 1 & , \text{otherwise} \end{cases} \quad (9)$$

Table 2. Species distribution in the dataset, after data augmentation

Species	Common name	Training images	Testing images	Total images
Simarouba amara	Marupa	828	360	1188
Osteophoeum platypermum	Cumala Llorona	996	432	1428
Aniba rosaeodora	Palo Rosa	828	360	1188
Otoba glycyarpa	Aguanillo	828	360	1188
Cedrela odorata	Cedro	828	360	1188
Swietenia macrophylla	Caoba	828	360	1188
Cedrelinga cateniformis	Tornillo	828	360	1188
Parahancornia peruviana	Naranjo Podrido	828	360	1188
Caryocar glabrum	Almendra	828	360	1188
Guazuma crinita	Bolaina	828	360	1188
Dipteryx micrantha	Shihuahuaco	828	360	1188
	Total	9276	4032	13308

Table 3. Selected eleven species from the Peruvian Amazon rainforest

Species	Image	Leaf features
Simarouba amara		Pointy apex and bottom end, irregular margin, no visible vein structure.
Osteophocum platypermum		Blunt bottom end, circular apex, oval shape, slightly visible vein structure.
Aniba rosaeodora		Pointy apex and bottom end, glossy and rough surface.
Otoba glycyarpa		Pointy apex and bottom end, undulated margins, elliptical shape.
Cedrela odorata		Pointy bottom end, circular apex, blade is narrow near the stem.
Swietenia macrophylla		Visible stem in bottom end and apex, circular shape, visible vein structure.
Cedrelinga cateniformis		Pointy apex and bottom end, undulated margin, vertically oriented veins.
Parahancornia peruviana		Circular apex and bottom end, oval shape, only central vein is visible.
Caryocar glabrum		Pointy apex and bottom end, oval shape, blade is broader near the stem, diagonally oriented veins.
Guazuma crinita		Visible stem at the apex, vertically oriented veins, glossy and bright surface.
Dipteryx micrantha		Curved apex and bottom end, smallest size of the 11 chosen species.

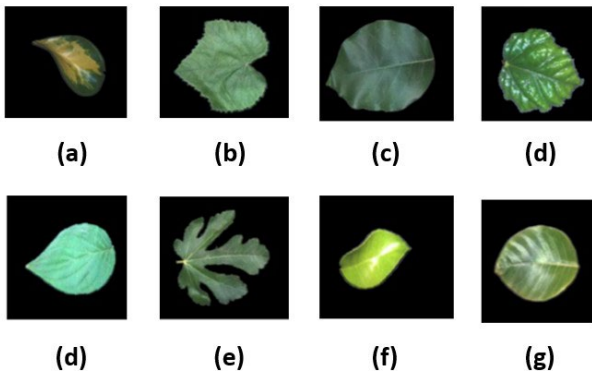


Fig. 10. Collected 8 fruit species: (a) Schefflera Arboricola, (b) Vitis Labrusca, (c) Persea Americana, (d) Bengonia Glabra, (e) Piper Aduncum, (f) Ficus Carica, (g) Ficus Benamina, (h) Artocarpus Heterophyllus lam

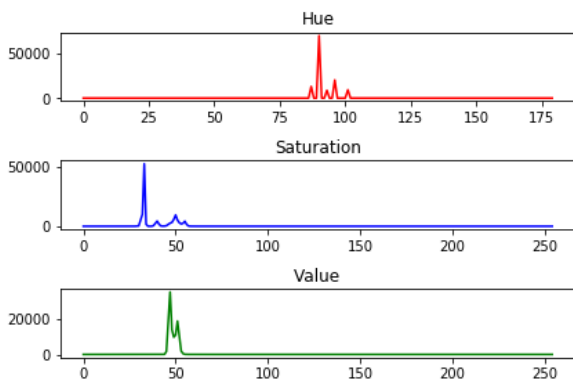


Fig. 11. Histograms of the Hue, Saturation and Value components for an image of the device background

$$I_{RR}(x,y) = I_R(x,y) \cdot Mas(x,y) \quad (10)$$

$$I_{GG}(x,y) = I_G(x,y) \cdot Mas(x,y) \quad (11)$$

$$I_{BB}(x,y) = I_B(x,y) \cdot Mas(x,y) \quad (12)$$

Step 2: The color filtered RGB image in Fig. 12 results from applying the mask in (7) to the original primary components, as per Equations (10)–(12):



Fig. 12. Image after color filtering

Step 3: Equation (13) calculates the conversion from the color filtered image to the gray-scale image in Fig. 13:

$$I(x,y) = 0.299 \cdot I_{RR}(x,y) + 0.587 \cdot I_{GG}(x,y) + 0.114 \cdot I_{BB}(x,y) \quad (13)$$



Fig. 13. Grayscale image

Step 4: The image background is easily distinguishable and has few artifacts after the color filter. Thus, the grayscale image $I(x,y)$ is thresholded to separate the background from the leaf, as per Equation (14):

$$F(x,y) = \begin{cases} 255 & , I(x,y) > u_a \\ 0 & , \text{otherwise} \end{cases} \quad (14)$$

The binary threshold u_a is computed using the Otsu method (Vijay and Patil, 2016, Hassanein et al., 2018), which employs the grayscale histogram to separate pixels into foreground and background classes. Then, it computes the interclass variance, and settles in a threshold value which minimizes this variance. This method achieves the best results and ensures that the pixels inside the image are assigned an intensity of 255 while the background ones get 0. The binary threshold computation and the thresholding operation are performed with OpenCV, using the `cv2.THRESH_OTSU` and the `cv2.threshold` functions respectively. The thresholding operation receives the grayscale image, the binary threshold and the intensity value assigned to the pixels with intensity greater than the computer threshold. Fig. 14 shows the resulting thresholded binary image.

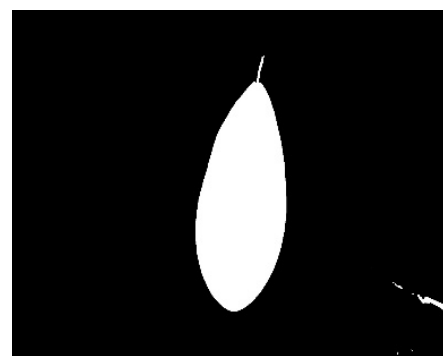


Fig. 14. Image after applying Otsu's method

Step 5: It is necessary to extract the edges of the binary image so that the largest contour (leaf shape) can be recognized and separated from the rest. To ensure an accurate edge detection, the Canny algorithm (Canny, 1986) is selected. This algorithm extracts the leaf edges in the thresholded binary image from the previous step. It is implemented using the cv2.Canny command from OpenCV library. It receives the input binary image from the previous step, the low gradient intensity threshold (0) and the high gradient intensity threshold (255). The result is stored in the binary image F_{Canny} . Then, F_{Canny} is morphologically dilated using Equation (15), to eliminate some artifacts after the Canny edge detector and connect any empty spaces in the leaf contour.

$$BC = F_{Canny} \oplus ES_{2 \times 5} \quad (15)$$

Where $ES_{2 \times 5}$ is an all-one rectangular structuring element with 2 rows and 5 columns. Fig. 15 shows the image after the Canny edge detector and the morphological dilation:

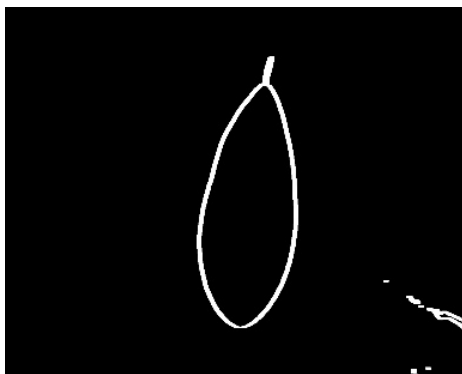


Fig. 15. Image after Canny edge detector and morphological dilation

Step 6: The final leaf mask $I_{seg}(x,y)$ results from choosing the largest contour in “BC” and filling the surrounded region with an intensity of 255. The cv2.fillPoly function from OpenCV performs the contour filling. It requires the destination image to draw the resulting filled polygon, the coordinates of the largest contour and the fill color (white). The result is shown in Fig. 16.

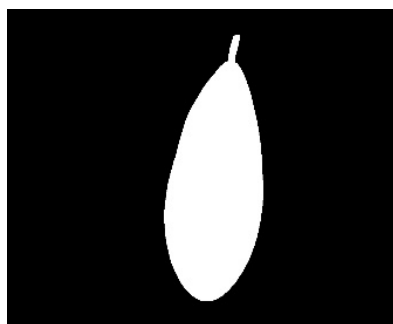


Fig. 16. Image of the final leaf mask, $I_{seg}(x,y)$

Step 7: The $I_{seg}(x,y)$ mask is applied to the original RGB image, resulting in the segmented leaf image at Fig. 17, with the following primary components:

$$I_{FR}(x,y) = I_R(x,y) \cdot I_{seg}(x,y) \quad (16)$$

$$I_{FG}(x,y) = I_G(x,y) \cdot I_{seg}(x,y) \quad (17)$$

$$I_{FB}(x,y) = I_B(x,y) \cdot I_{seg}(x,y) \quad (18)$$



Fig. 17. Segmented image

These methods achieve good leaf segmentation results, mostly due to the uniform acquisition conditions inside the developed equipment. These conditions allow the use of a straightforward image processing pipeline.

2.2.4 Re-scaling

The segmented images with primary components $I_{FR}(x,y)$, $I_{FG}(x,y)$ and $I_{FB}(x,y)$ are rescaled to sizes 224×224 and 227×227 pixels, since these are the input sizes for the CNN models in this work. Image scaling is done via bilinear interpolation (Kirkland, 2010), since it achieves adequate results and has a lower computational load than other methods such as cubic interpolation. The resulting image has primary components $I_{FFR}(x,y)$, $I_{FFG}(x,y)$ and $I_{FFB}(x,y)$, and is shown in Fig. 18.

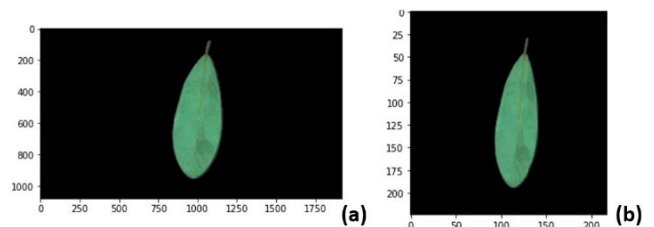


Fig. 18. Re-scaling example: (a) Image after segmentation. (b) Re-scaled image, 224×224 pixels

Overall, bilinear interpolation provides good results resizing the segmented images. The details observed in the leaf structure are preserved, which will be beneficial for the classification process.

2.2.5 Convolutional Neural Network

Transfer learning enables employing learnt features in pre-trained neural network models for a different application than it was originally trained for. The following 3 models are compared to choose the best CNN: AlexNet (Krizhevsky et al., 2017), MobileNet (Howard et al., 2017) and VGG-16 (Tammina, 2019). Said models were chosen due to the following reasons:

- The classification task is focused on a single type of object (tree leaves). Thus, a simple model like AlexNet may achieve good results.
- Since the project intends to add more species to the classification task in the future, new models will need to be trained. Therefore, a lightweight CNN model like MobileNet could help to store new trained models while keeping below the maximum storage capacity.
- One of the goals of the portable equipment is to achieve an accuracy greater than 95%. In turn, a model like VGG-16 known for its high performances in classification tasks could achieve this.

In the 3 models, the last layer is changed to a softmax layer with 11 outputs for each of the timber species. MobileNet and VGG-16 are pre-trained in the ImageNet dataset, consisting of around 14 million images with more than 20000 classes (Ghorui et al., 2023). Training lasts for 20 epochs with a batch size of 32, using the Adam (Kingma and Ba, 2014) optimizer with a learning rate of 0.001. Limited memory resources result in using a small batch size, which in turn results in a small learning rate. Training is done with early stopping, after 12 epochs of continuous reduction in accuracy. Fig. 19 shows the CNN architectures and training parameters used for this comparison. The results obtained for each of the CNN models and other characteristics are compared in Table 5. Evidently, MobileNet achieves the best accuracy with the smallest memory footprint next to AlexNet or VGG-16. The MobileNet model uses separable depth convolutions followed by point convolutions to achieve its low memory footprint. It has a depth of 88 layers and uses ReLU activations (Howard et al., 2017).

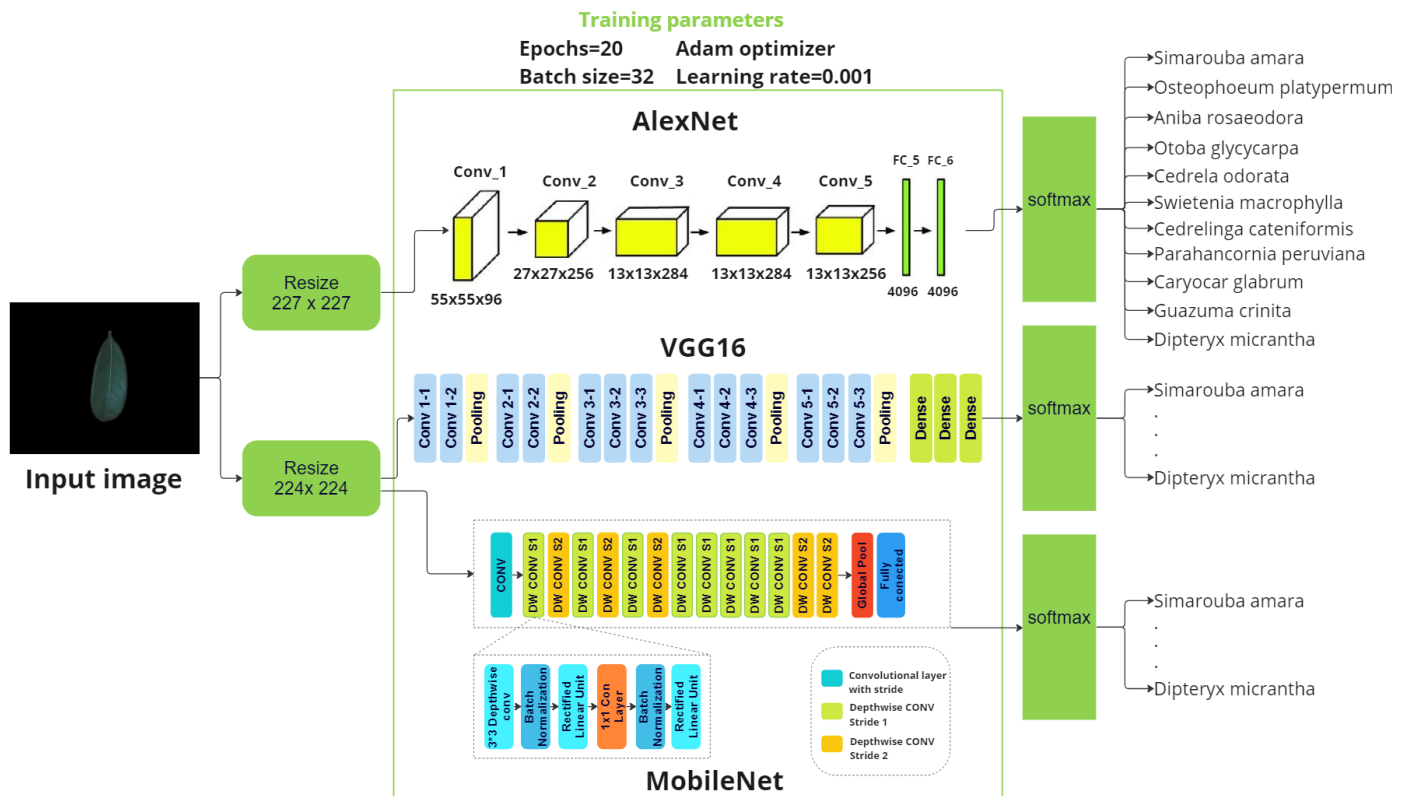


Fig. 19. CNN architectures

Table 5. Model comparison: AlexNet, MobileNet and VGG-16

Model	Depth	# of parameters (M)	Memory (MB)	Accuracy (%)
AlexNet	8	60	260	86.11
VGG-16	23	138	528	95.39
MobileNet	88	4	16	98.64

2.2.6 Graphical User Interface and Device Operation

A graphical user interface made in Tkinter enables users to easily operate the device. Fig. 20 shows the leaf classification screen after the software has analyzed a leaf. The operation steps are described below:

- Step 1: The user places the leaf in the sliding tray.
- Step 2: Using the mousepad, the user chooses the classification option in the graphical user interface menu.
- Step 3: The user presses the “Capture” button, after which the captured (left) and segmented (right) images appear on screen.
- Step 4: If user considers that the image is correctly taken, they press the “Identify” button; otherwise, they repeat the image acquisition with the “Repeat” button.
- Step 5: After the algorithm has identified the leaf, the user may press the “Save” button if they desire to store the segmented image.

If the system detects that the leaf does not belong to any species in the dataset (see sections 2.3.2.), the interface shows a secondary screen as seen in Fig. 21. The operation steps are:

- Step 1: The user presses the “Back” button in the upper left corner and chooses the species record from the menu.
- Step 2: The user presses the “Capture” button and, if the image has been correctly taken, inputs a species name in the text box. Otherwise, the user can repeat the image acquisition.
- Step 3: The user presses the “Upload” button.

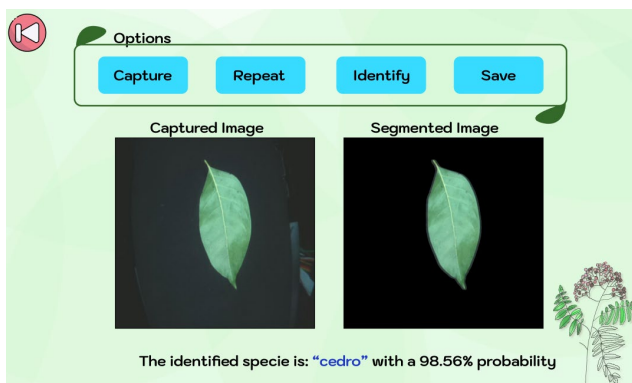


Fig. 20. Main screen in the graphical user interface

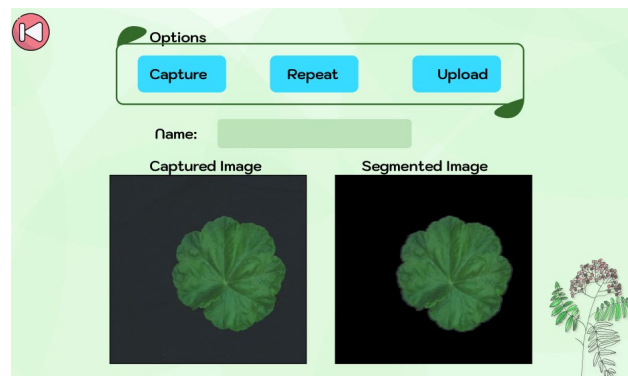


Fig. 21. Screen to record a new species

2.2.7 Performance Evaluation Metrics

The following metrics are used to evaluate the equipment’s performance: accuracy, precision, recall and F1-score. Each metric is described as follows:

- Accuracy: Is the percentage of correctly classified images with regards to the total number of predictions (Tharwat, 2021). It is computed according to (19):

$$\text{Accuracy} = \frac{\# \text{ of correct predictions}}{\text{Total \# of predictions}} \quad (19)$$

- Precision: Represents the ratio of true positive samples with respect to the total number of positively classified samples (TP+FP) (Tharwat, 2021). It is computed as per (20):

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (20)$$

Where:

TP: # of true positives (Images correctly classified as a particular class)

FP: # of false positives (Images classified as a class but belonging to a different)

- Recall: Ratio of true positive samples with respect to the total of predicted samples (TP+FN) (Tharwat, 2021). It is computed according to (21):

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (21)$$

Where:

FN: # of false negatives (Images belonging to a class which were not classified as said class)

- F1-score: It is the harmonic mean between precision and recall, and is computed by (22):

$$\text{F1-score} = 2 \times \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \quad (22)$$

3. RESULTS AND DISCUSSION

Model evaluation is performed using Google Collab. 30.30% of images are used to test the classification model while 30.66% of images are used to evaluate the model that determines whether an input belongs to the dataset.

3.1 Identification of whether the Image Belongs to the Classification Dataset

Figs. 22–24 show the obtained confusion matrices. Table 6 presents the metrics for each model, computed via the confusion matrices. The VGG-16 model achieves the lowest accuracy, at 91.57%. This might be explained due to VGG-16 having the largest number of parameters, which might result in the network overfitting during training. On the other hand, MobileNet achieves an accuracy of 99.34% and AlexNet, of 98.61%. Finally, the MobileNet model is chosen due to its higher accuracy and its lower memory

footprint.

Table 6. Model comparison, determination if an input belongs to the dataset

	Accuracy (%)	Precision (%)	F1-score (%)
AlexNet	98.61	98.61	98.61
VGG-16	91.57	92.03	91.52
MobileNet	99.34	99.34	99.34

3.2 Classification of the 11 Timber Species

Figs. 25–27 show the species classification confusion matrices. Table 7 shows the accuracy per timber species for the 3 evaluated models. Table 8 shows confusion matrix metrics (Precision, Accuracy, F1-score) for each model.

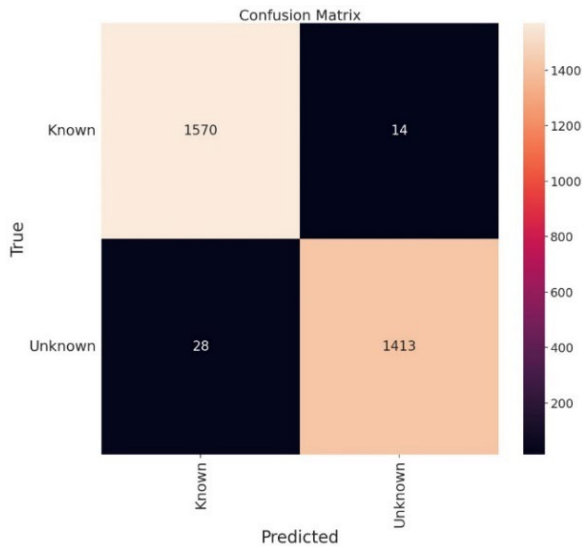


Fig. 22. Confusion matrix for AlexNet, determination if an input belongs to the dataset

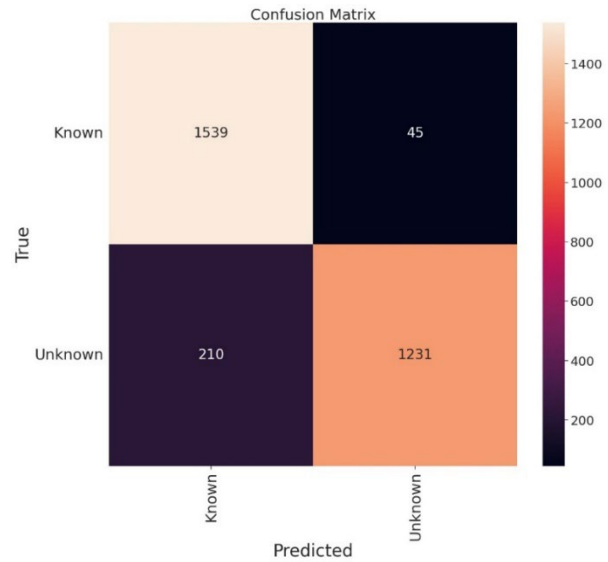


Fig. 23. Confusion matrix for VGG-16, determination if an input belongs to the dataset

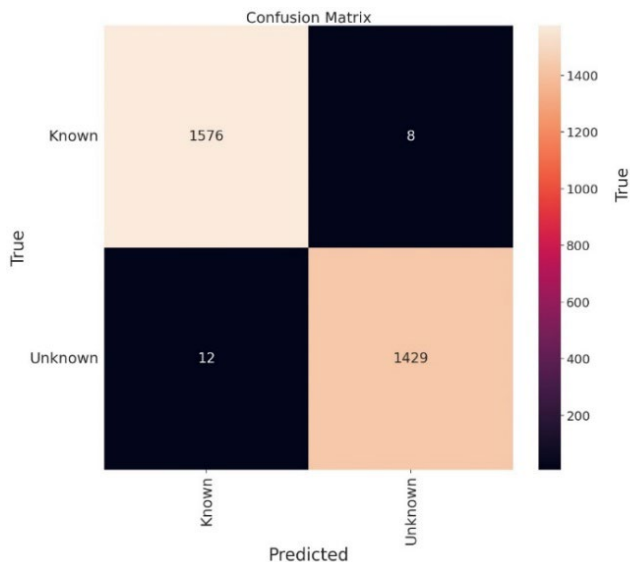


Fig. 24. Confusion matrix for MobileNet, determination if an input belongs to the dataset

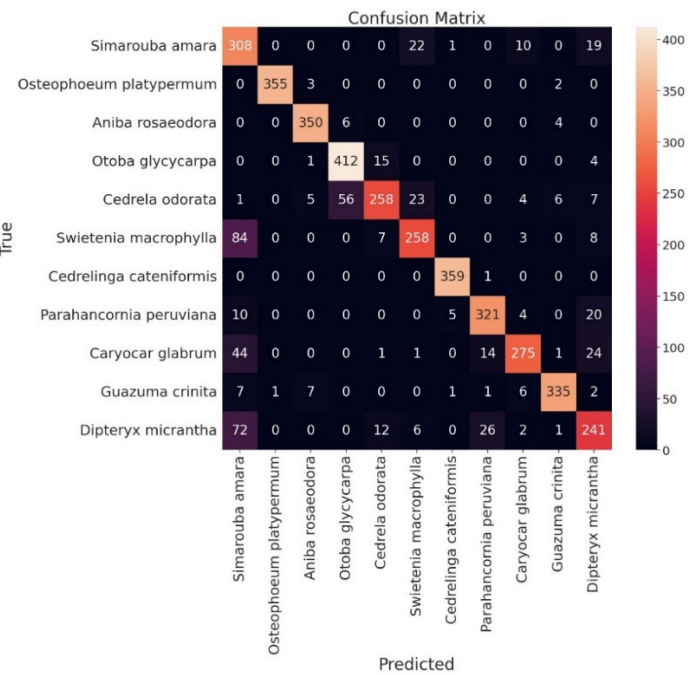


Fig. 25. Confusion matrix for AlexNet, 11 timber species classification

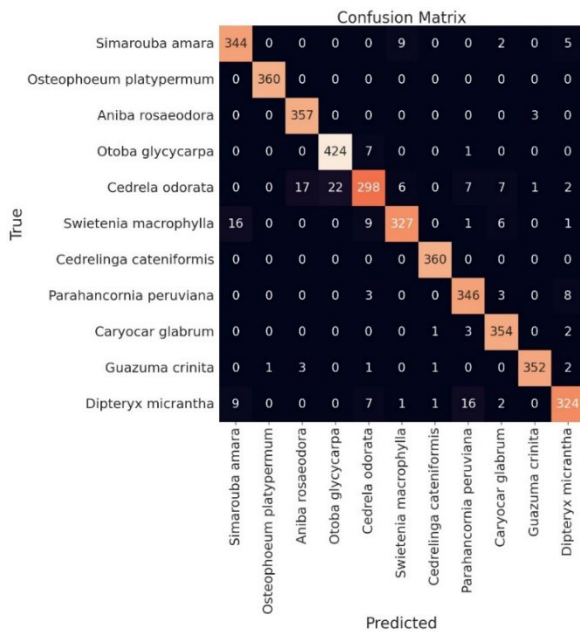


Fig. 26. Confusion matrix for VGG-16, 11 timber species classification

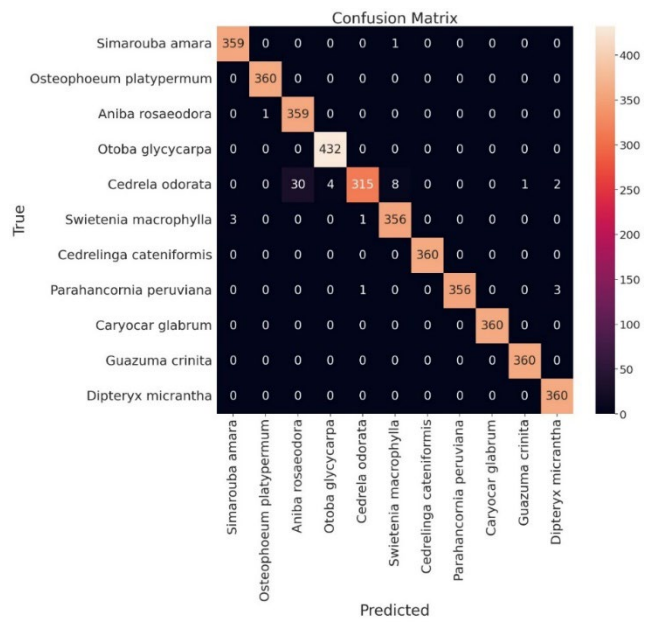


Fig. 27. Confusion matrix for MobileNet, 11 timber species classification

Table 7. Accuracy for each model (AlexNet, VGG-16, MobileNet) for each of the 11 timber species

Species	Accuracy (%)		
	AlexNet	VGG-16	MobileNet
Simarouba amara	85.56	95.56	99.72
Osteophoeum platypermum	98.61	100	100
Aniba rosaeodora	97.22	99.17	99.72
Otoba glycyarpa	95.37	98.15	100
Cedrela odorata	71.67	82.78	87.50
Swietenia macrophylla	71.67	90.83	98.89
Cedrelinga cateniformis	99.72	100	100
Parahancornia peruviana	89.17	96.11	98.89
Caryocar glabrum	76.39	98.33	100
Guazuma crinita	93.06	97.78	100
Dipteryx micrantha	66.94	90	100

Table 8. General comparison of the 3 models (AlexNet, MobileNet, VGG-16)

	Accuracy (%)	Precision (%)	F1-score (%)
AlexNet	86.11	87.19	86.22
VGG-16	95.39	95.37	95.33
MobileNet	98.64	98.69	98.62

Table 7 shows that AlexNet and VGG-16 perform poorly for some timber species, achieving an accuracy of less than 95% for Cedrela odorata, Swietenia macrophylla and Dipteryx micrantha. AlexNet has the worst performance, achieving a 95% accuracy for only 5 of the 11 species, while VGG-16 did so for 8 species and MobileNet, for 10 species. Nonetheless, MobileNet (87.5%) does not achieve the 95% accuracy threshold for Cedrela odorata, just as happens for AlexNet (71.67%) and VGG-16 (82.78%). This might be

explained by the leaves having many similar features as other species, such as diverse margins (undulated or straight), apices and bottom ends.

In general terms, AlexNet achieves the lowest accuracy (86.11%) which is lower than 90%, while MobileNet (98.64%) and VGG-16 (95.35%) surpass the 90% threshold. Additionally, MobileNet’s precision is 98.69% and its F1-score, of 98.62%, which indicates that the model can correctly predict classes as well as avoid assigning incorrect classes to samples.

Figs. 28–30 show the accuracy and loss curves for each model during training. Fig. 28 shows that, for AlexNet, the validation loss slightly increases and the validation accuracy slightly decreases in the last epochs, which triggered the early stopping criterion at epoch 16. Fig. 29 shows that VGG-16 achieved a stable loss and accuracy from epoch 5 onwards. This stagnated loss and accuracy curve, coupled with the fact that the validation curve is somewhat far from the training curve, might indicate overfitting. Finally, Fig. 30 shows that MobileNet has a validation loss and accuracy spike at epoch 5 but both curves quickly close the gap to the training loss and accuracy, achieving a value of 0.1 (loss) and 98.69% (accuracy) by the last epoch.

Based on the Figs. and Tables shown above, the MobileNet architecture was selected because it has the highest accuracy of the 3 architectures (98.64%) and requires the least storage space per trained model (16MB). Thus, a new model can be trained in the network with more species and then update the one stored inside the equipment without requiring a large storage space. Moreover, a more lightweight model tends to have a lower computational load and thus, a lower power consumption.

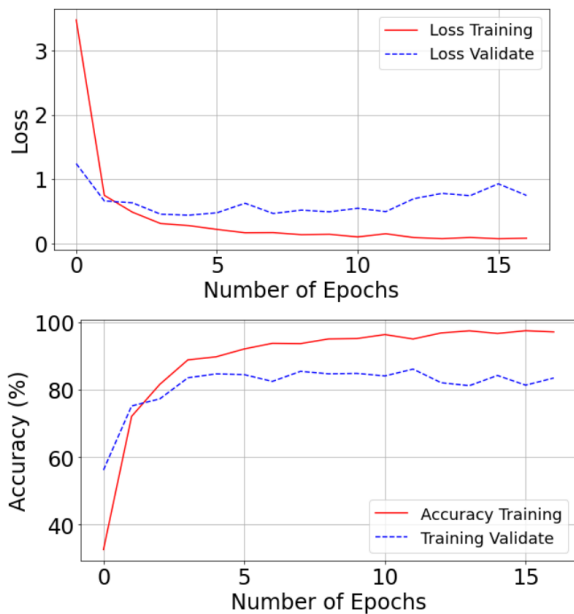


Fig. 28. Loss and accuracy for AlexNet during training

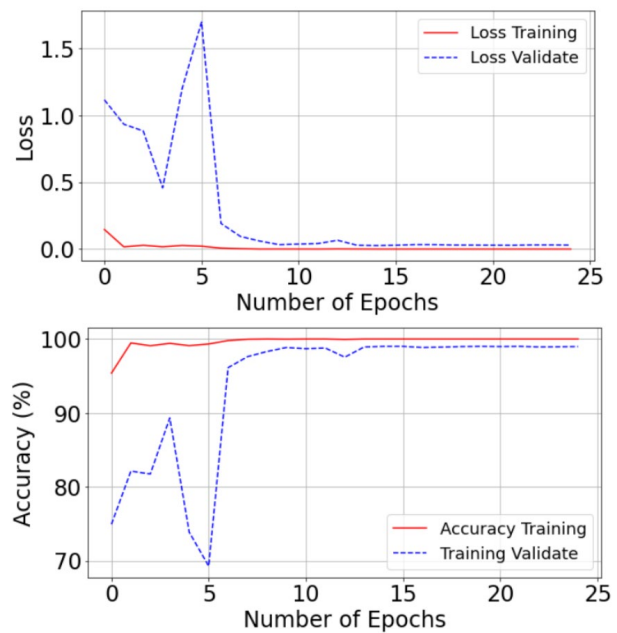


Fig. 30. Loss and accuracy for MobileNet during training

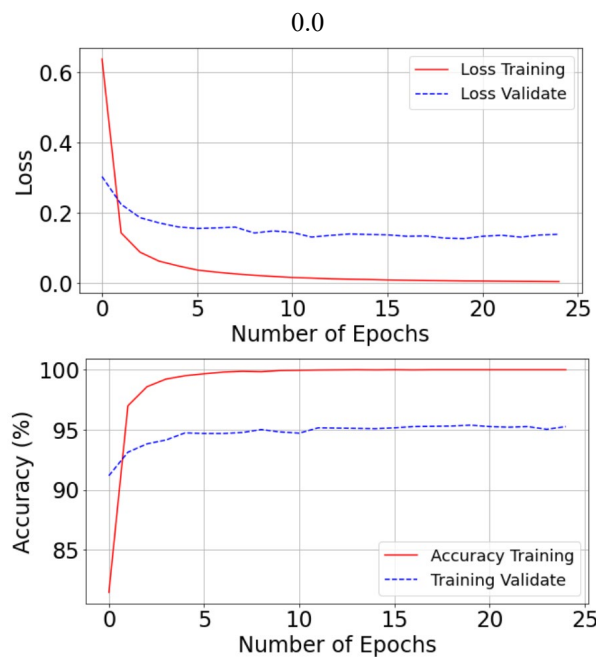


Fig. 29. Loss and accuracy for VGG-16 during training

3.3 Related Work Comparisons

Table 9 shows a comparison of the classification accuracy between the present work and related ones.

Two of the related works achieved an accuracy lower than 90%. One of these (Selda et al., 2017) relied on the leaf vein structure and an SVM, but did not take into account the leaf surface characteristic.

Likewise, authors at Monroy-de-Jesús et al. (2019) built a custom basic CNN with limited results. The proposed system achieved the third highest accuracy of the models. The proposal at Varguese et al. (2020) considered medicinal plants which could be easily distinguished and did not require a large dataset to train the neural network. Similarly, Zarrin and Islam (2019) evaluated fruit trees with somewhat distinguishable leaves and had a dataset with 10,000 images.

Besides the proposed image pre-processing and neural network-based algorithm, this work also details a hardware device which captures leaf images and shows the results through a graphical interface, which enables in-situ leaf identification and analysis.

Table 9. Results comparison with related work

Reference	Number of identifiable species	Dataset	Maximum achieved accuracy
Developed device with proposed algorithm	11	Custom dataset	98.64%
(Selda et al., 2017)	20	Custom dataset	84.29%
(Singh et al., 2020)	32	Flavia dataset	95%
(Monroy-de-Jesús et al., 2019)	11	Custom dataset	84.07%
(Song et al., 2019)	10	SVHN dataset	98.27%
(Srivastava and Khunteta, 2018)	16	Flavia dataset	90.90%
(Zarrin and Islam, 2019)	10	Custom dataset	99.40%
(Kang and Oh, 2018)	63	Custom dataset	96.08%
(Varghese et al., 2020)	6	Custom dataset	99%

4. CONCLUSIONS AND FUTURE WORKS

The large amounts of globally traded timber require regulations and control of existing species to avoid issues such as illegal logging, and current methods are not sufficiently appropriate. This work presents a portable leaf image acquisition equipment whose design ensures good image capturing conditions. It works jointly with a Python algorithm and a graphical user interface which enables easy user operation. This device serves to identify species in situ and to build new species datasets through imaging and recording of unregistered species.

Apart from ensuring good capture conditions, the equipment was manufactured with carbon fiber and considering requirements such as: rigidity, low weight and portability in order to prevent health problems in the botanical experts who will operate said equipment in the long term. Also, thanks to the components inside the equipment, the illumination control and uniform background, the image segmentation is facilitated, which allows the software to work effectively without the need for complex processing or the use of high computational resources.

In addition to cutting computational resources from the segmentation process, the algorithm uses the MobileNet architecture, which due to its low memory footprint does not need to consume high resources to perform a classification and leaves plenty of space to train new models with updated datasets with more species in the future. MobileNet achieved 98.64% accuracy, superior to the AlexNet and VGG-16 models. Future works can improve these results by building larger datasets with thousands of images per species to improve the training process.

Finally, the device aims to be operated by forestry institutions or botanical to increase the analysis throughput and precision when identifying timber species from the Peruvian amazon rainforest.

DECLARATIONS

FUNDING

This work was funded by “PROCIENCIA” (formerly FONDECYT), an initiative of CONCYTEC under contract number 052-2021-FONDECYT

DATA AVAILABILITY STATEMENT

Please contact the corresponding author (guillermo.kemper@upc.pe) to access the dataset generated in this work.

ACKNOWLEDGMENTS

To “PROCIENCIA” (formerly FONDECYT), an

initiative of CONCYTEC, for the funds allocated to the project under the contract N° 052-2021-FONDECYT.

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

REFERENCES

- Canny, J. 1986. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 679–698.
- Chen, R.C., Saravananarajan, V.S., Hung, H.T. 2021. Monitoring the behaviours of pet cat based on YOLO model and raspberry Pi. *International Journal of Applied Science and Engineering*, 18, 1–12.
- Ferreira, C.A., Inga, J.G., Vidal, O.D., Goytendia, W.E., Moya, S.M., Centeno, T.B., Vélez, A., Gamarra, D., Tomazello-Filho, M. 2021. Identification of tree species from the peruvian tropical amazon “selva central” forests according to wood anatomy. *BioResources*, 16, 7161–7179.
- Ghorui, A., Chatterjee, S., Makkar, R., Pachiyappan, A., Balamurugan, S. 2023. Deployment of CNN on colour fundus images for the automatic detection of glaucoma. *International Journal of Applied Science and Engineering*, 20, 1–9.
- Gonzales, R.C., Woods, R.E. 2018. Digital image processing. fourth edition. In Pearson Education, Inc.
- Hassanein, M., Lari, Z. El-Sheimy, N. 2018. A new vegetation segmentation approach for cropped fields based on threshold detection from hue histograms. *Sensors*, 18, 1–25.
- Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H. 2017. MobileNets: Efficient convolutional neural networks for mobile vision applications. Retrieved 2023-01-31 from <https://www.electronicaplugandplay.com/sistemas-embe-bidos/product/797-rpi4-4gb>
- Kang, E., Oh, I.S. 2018. Weak constraint leaf image recognition based on convolutional neural network. *International Conference on Electronics, Information and Communication, ICEIC 2018, 2018-Janua*, 1–4.
- Kingma, D.P., Ba, J.A. 2014. A method for stochastic optimization. *Computer Research Repository*.
- Kirkland, E.J., 2010. Bilinear interpolation. *Advanced Computing in Electron Microscopy*, 261–263. Springer US.
- Krizhevsky, A., Sutskever, I., Hinton, G.E. 2012. ImageNet classification with deep convolutional neural networks. *Neural Information Processing Systems*, 25.
- Maharana, K., Mondal, S., Nemade, B. 2022. A review: Data pre-processing and data augmentation techniques. *Global Transitions Proceedings*, 3, 91–99.
- Monroy-de-Jesús, J., Reyes-Nava, A., Olmos, F. 2019. Clasificador de plantas medicinales por medio de deep

- learning. *Research in Computing Science*, 148, 65–78.
- Ozturk, S. Akdemir, B. 2017. Automatic leaf segmentation using grey wolf optimizer based neural network. In 2017 Electronics, 1–6.
- Raspberry Pi. 2020. Raspberry pi high quality camera. Retrieved 2023-01-31 from https://static.raspberrypi.org/files/product-briefs/Raspberry_Pi_HQ_Camera_Product_Brief.pdf
- Selda, J.D.S., Ellera, R.M.R., Cajayon, L.C., Linsangan, N.B. 2017. Plant identification by image processing of leaf veins. *Proceedings of the International Conference on Imaging, Signal Processing and Communication*, 40–44.
- Singh, G., Aggarwal, N., Gupta, K., Misra, D.K. 2020. Plant identification using leaf specimen. 2020 11th International Conference on Computing, Communication and Networking Technologies, 1–7.
- Soetedjo, A. Hendrianti, E. 2022. A comparative study of vetiveria zizanioides leaf segmentation techniques using visible, infrared, and thermal camera sensors in an outdoor environment. *Applied System Innovation*, 6, 1.
- Song, Y., He, F., Zhang, X. 2019. To identify tree species with highly similar leaves based on a novel attention mechanism for CNN. *IEEE Access*, 7, 163277–163286.
- Srivastava, V., Khunteta, A. 2018. Comparative analysis of leaf classification and recognition by different SVM classifiers. 2018 International Conference on Inventive Research in Computing Applications, 626–631.
- Tamina, S. 2019. Transfer learning using VGG-16 with deep convolutional neural network for classifying images. *International Journal of Scientific and Research Publications*, 9.
- Tharwat, A. 2021. Classification assessment methods. *Applied Computing and Informatics*, 17, 168–192.
- Varghese, B.K., Augustine, A., Babu, J.M., Sunny, D., Cherian, S. 2020. INFOPLANT: Plant recognition using convolutional neural networks. 2020 Fourth International Conference on Computing Methodologies and Communication, 800–807.
- Vijay, P.P., Patil, N.C. 2016. Gray scale image segmentation using otsu thresholding optimal approach. *Journal for Research*, 2, 20–24.
- Winkel, G., Leipold, S., Buhmann, K., Cashore, B., De Jong, W., Nathan, I., Sotirov, M., Stone, M. 2017. Narrating illegal logging across the globe: Between green protectionism and sustainable resource use. *International Forestry Review*, 19, 81–97.
- Zarrin, I., Islam, S. 2019. Leaf based trees identification using convolutional neural network. 2019 IEEE 5th International Conference for Convergence in Technology, 1–4.