

Tree Species Identification and Geo-Mapping using Mobile Technology

Sergio R. Peruda Jr.¹, Roben A. Juanatas², Ryujen B. Minoru³, John David V. Verterra⁴, and Carlos A. Zacarias⁵

¹²Faculty of School of Engineering,
Asia Pacific College, 3 Humabon Place, Magallanes
Makati City, Philippines 1232

²Student of School of Engineering
Asia Pacific College, 3 Humabon Place, Magallanes
Makati City, Philippines 1232

Corresponding Author's Email: ¹sergiop@apc.edu.ph

Article History: Received xxxxx; Revised xxxxx; Accepted xxxx

ABSTRACT: The Department of Environment and Natural Resources (DENR) of the Philippines helps preserve native and production trees through strict supervision operated under their active laws and policies. A tree inventory form is provided by the office to register known trees around a forest land or community upon the request to cut down trees or tree preservation. Some of the form's contents include identification of tree species with its geographic location. Using the mobile application Pic-A-Tree, the manual process of identifying tree species and its location has been assisted using Android smartphone. The image of the tree leaf captured by the phone undergoes image processing and allows the user to send tree identification request to the server and return with an information confirming its identified tree specimen and is plotted graphically through Google map. The results of this study show that the system achieved 91% accuracy from the 100 samples conducted, certainly proving to have a good score. Thus, the system proves to be a reliable and beneficial utility for the identification and plotting of trees in the environment as a potential starting point for our government foresters and the development of this technology.

KEYWORDS: *Tree Inventory, Geographic location, Mobile Application, Image Processing, Google Map*

1.0 INTRODUCTION

Trees are all around us. Even within our urban landscape trees can still be found. Beyond the cities, there are plantations of trees which supplies us with wood. The

wood gathered from these plantations are used for many different things. The amount of wood collected is tallied through tree inventory [1].

Recently, Philippines is one of the tropical countries facing forest distinction due to technologically advanced logging, agricultural and urban expansion. The problem of denudation rises even with the idea of private tree farming and forest ownership that could hopefully solve the issue. The government could not protect forests even with well-armed guards.

Previous projects and studies have shown that there are different ways to be able to identify the type of plant, but end-up using the leaf of plants as basis for its species [2, 3, 4, 5, 6, 7, 8, 9, 10]. These projects focused on delivering the type of plant towards the user and were also able to show more information about the plant by analyzing the image of the leaf. A study shows the process in which it analyzes the leaf to identify its species. The process includes the application grayscale conversion towards the image then edge detection and dilation. After which, they select a part of the leaf, and apply “Inside gray-level co-occurrence matrix” to classify its tree specimen and determine its credibility [8].

Apparently, most research studies were focused more on plant families, but fewer studies focus on tree families due to its constraints of the height and the difficulty of obtaining fresh leaf samples. Since both tree and plant belonged on similar family or branch in science and its leaf characteristics are almost interchangeable, the identification of tree species based on leaf structure may be possible in the near future. The fact that the idea impacts the environment and government’s management for tree preservation proves that the development can be meaningful.

1.1 Background of the Study

According to DENR chief forester, the responsibility of the office is to produce a tree inventory form if a request to cut down a tree is issued. The manual process of identifying the species of a tree is implemented since there is no instrument provided to the foresters when identifying tree species. Foresters must rely on their own knowledge, memory, and landmarks to evoke the aspects of trees to identify, which sometime causes inconsistencies when identifying the type of tree and its location. The actual method of identifying tree species perceived to be time consuming and inconsistent especially on plantation and forest reserves. The fact that there are few mobile specie identification applications dedicated on smartphones, a need for an Android application using digital image processing and computer processing, must be introduced or redeveloped. The idea is innovative since Android have adaptable and supportive technologies that can integrate certain functions and applications. Therefore, developing an interactive and responsive mobile automated application would hopefully solve the problem exerted on the conventional method.

In this case, the proponents aim to develop a tree classifying and locating Android application connected to a server-based system to assist the DENR in identifying the species and the location of trees that will be displayed on a tree mapping module. They only focus on five common and naturally growing trees in CALABARZON region in

the Philippines. The application will be Android compatible only to help support other platforms, features, and applications during the development process. In this way, the system will make the current process more comprehensive, accurate, reliable in recognition, consistent, and response efficient.

1.2. General and Specific Objectives

The proponents aim to develop a system that identifies tree species and map it geographically using mobile application. Given the primary object of the study, they must also satisfy the system requirements of the specific objective that consists of the following:

- To create an accurate system for tree species identification and geo-mapping
- To develop a reliable system in leaf image identification of tree species
- To design a user-friendly graphical user interface with a mapping module
- To determine the consistency of tree species at varying distance
- To develop a responsive system throughout the process of leaf identification

1.3. Constraints

- The mobile application must be able to:
 - identify the species of tree
 - provide the longitude and latitude of tree location
- The mobile application must operate under the laws and policies of the Department of Natural and Environmental Resources.
- Google Maps application program interface is written in JavaScript.

1.4. Conceptual Framework

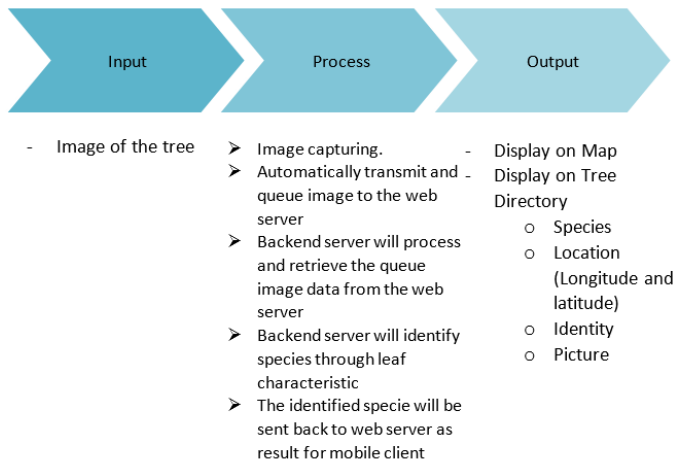


Figure 1. Conceptual framework of the study.

Figure 1 illustrates the input, process, and output of the proposed system. The system

identifies tree species based on leaf image and the location of its area. The exact location will be determined once the user captured the image of the tree leaf proximately close to the tree itself. The user’s confirmation of the taken photo will be automatically sent to the server for processing. After the server receives the queried image, leaf features will be extracted. Then it will compare the extracted features with the basis feature from the training dataset. The result will transmit back to the mobile client who sent the request for leaf identification. It will then display the tree specimen and its geographic location through a map.

2.0. METHODS AND PROCEDURES

2.1. Functional Analysis

Figure 2 shows the different internal functions of the Pic-A-Tree system expected to perform. The system acquires data using the front-end application installed in the Android device. It uploads the data to the server for preprocessing and feature extraction by image processing. The image features extracted will then be compared with the pre-defined features datasets from the training dataset for classification of tree species.

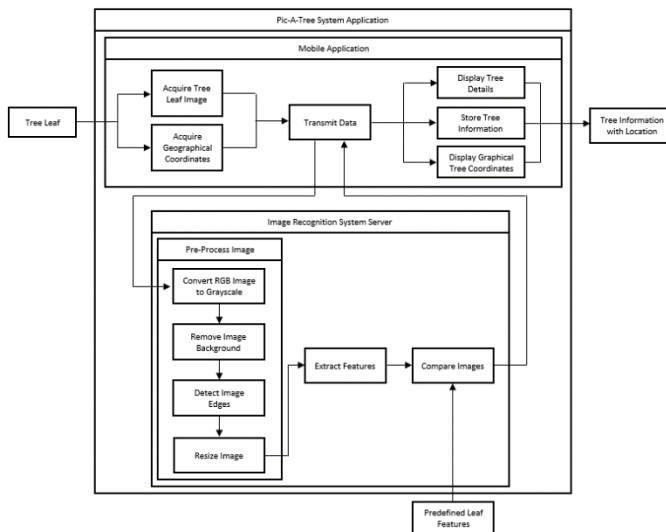


Figure 2. Glass box of system functions.

The result will be fed into the host device or other devices displaying information of a tree, which is the species and geographic coordinates pinpointing its location on the map.

2.2. System Flow

Figure 3 portrays the proposed system as a flowchart, explaining the full process structure starting from an image input ending on displaying the output. From the start,

the user will input the image of the tree leaf. After which, it will be sent to the webserver for the backend server to receive and where it will be processed using Python image processing. The first step in image processing is through preprocessing where the image will be simplified by gray image conversion, image segmentation, edge detection, and image resize. The simplified image will be processed in the feature descriptor algorithm extracting its potential feature key points as data. The features will be the essential data for the machine learning stage used to compare the trained data with the raw data. With this image recognition system process, the system will be able to classify the specie of the leaf and will save the information back to the webserver consisting of the tree species with its geographic location. The results will be observed externally on the Android application and API map interface integrated into the system comprising the identified tree species.

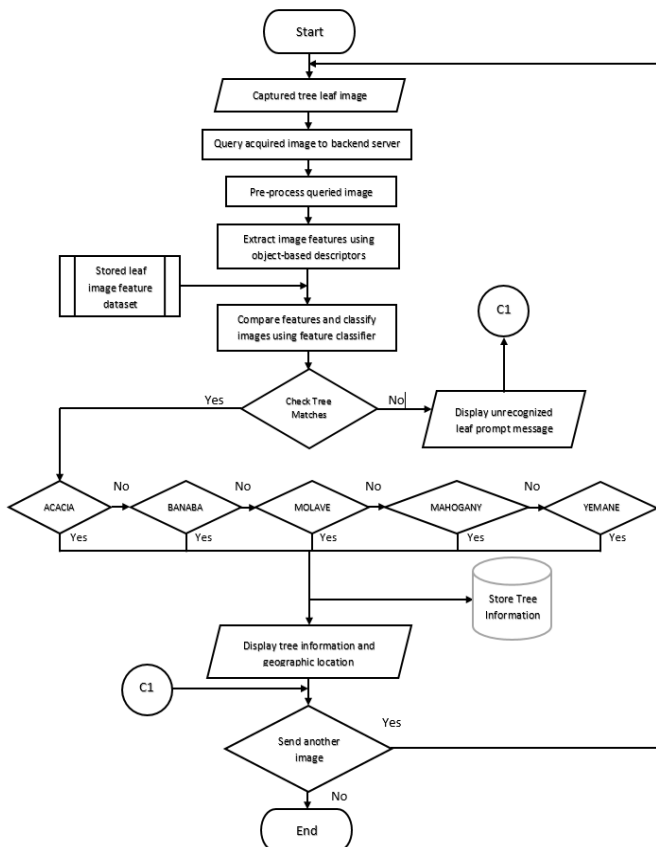


Figure 3. System flowchart.

2.3. Backend System (Image Recognition System)

The backend system comprises of the image processing and machine learning operated using OpenCV library and Python programming language. Primarily, the proponents used the Python to compile, debug and execute the command code necessary to perform the functions for image recognition. Python imports modules and libraries from other third-party entities making it useful for other external resources. OpenCV digital library will be used as the secondary resource compiler executing the image processing functions in Python. Those functions consist of preprocessing and feature extraction. The design system still relies on OpenCV libraries since most algorithms and functions needed to perform image processing and image recognition can be found in this module.

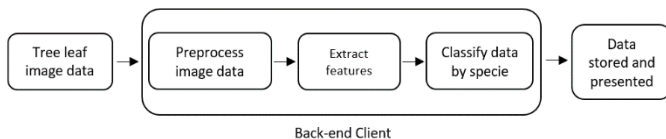


Figure 4. Image recognition block diagram.

Figure 4 shows the steps applied in the image processing. The captured images require pre-processing to ensure efficient and effective image analysis. The keypoint descriptor locates and defines the scale-invariant feature transform (SIFT) features and used it as dataset for the succeeding stage. The classification executes the support vector machine (SVM) classifier with training dataset and compare the queried image from the dataset to determine the best results of tree species.

2.3.1. Preprocessing

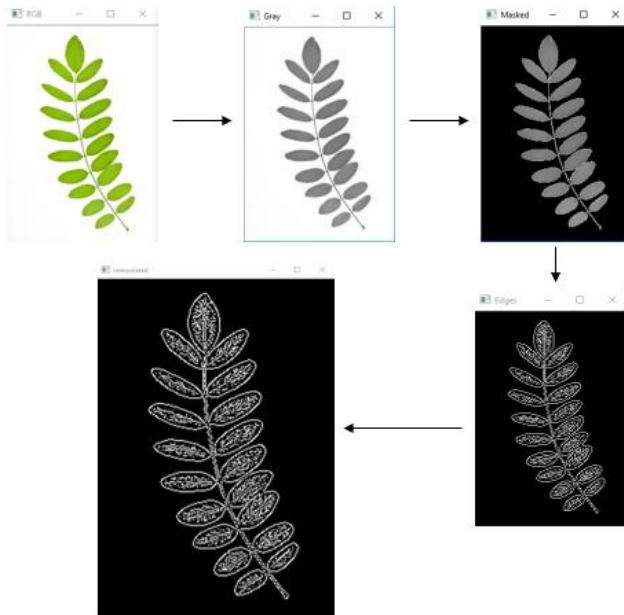


Figure 5. Preprocessing image output.

Figure 5 displays the output image of the image preprocessing stage using python compiler. The preprocessing methods proceed by converting the colored image into grayscale, followed by image segmentation, then by edge detection, and finally by resizing the image. The output shows the repetitive sequential steps during a request. With its capability to run multiple samples, the simulation only shows 1 sample request, which seems to be an acacia leaf.

2.3.2. Feature extraction

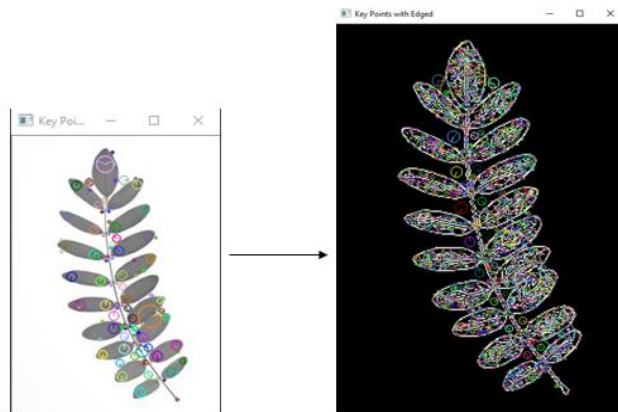


Figure 6. Feature keypoints extraction image output.

Figure 6 displays the output image of the feature extraction stage using Python compiler. Given the SIFT keypoint parameters, the image shows the comparison between images with the obtained key points from a grayscale image without completely undergoing the preprocessing steps. It proves that more key points can be obtained from the grayscale image.

Scale Invariant Feature Transform (SIFT) requires 4 major stages of computation to generate scale-invariant features on multiple possible scales and angles. The 4 major stages are: scale-space extrema detection, keypoint localization, orientation assignment, and keypoint descriptor [11, 12].

2.3.2.1. Scale-space extrema detection

$$(1) \quad D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y)$$

where $\sigma(k\sigma)$ = scaling parameter of the Laplacian of Gaussian (original scale), $L(x, y, k\sigma) = G(x, y, k\sigma) * I(x, y)$, and, $L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$

Equation (1) briefly explains the first stage of computation wherein it searches all possible scales and image locations, implemented efficiently by the difference-of-Gaussian (DoG) function to identify scale invariant and orientation keypoints.

2.3.2.2. Keypoint localization

$$(2) \quad D(\hat{x}) = D + \frac{1}{\sqrt{24}} \frac{\partial D}{\partial x} \hat{x}$$

where D and derivatives are evaluated through the sample point, $x = (x, y, \sigma)^T$ as the offset point

Equation (2) explains the computation to select candidates of location and scale wherein the keypoints can be measured for stability.

2.3.2.3. Orientation assignment

The scales of sample keypoints are computed in scale invariant manner with applied selection of Gaussian smoothed image $L(x, y)$, with the nearest scale, but recomputed with pixel differences.

$$(3) \quad m(x, y) = \sqrt{(L(x + 1, y) - L(x - 1, y))^2 + (L(x, y + 1) - L(x, y - 1))^2}$$

$$(4) \quad \theta(x, y) = \tan^{-1}((L(x, y + 1) - L(x, y - 1)) / (L(x + 1, y) - L(x - 1, y)))$$

where $L(x, y)$ is the scale, $m(x, y)$ the gradient magnitude, and $\theta(x, y)$ the orientation

Equation (3) and (4) generate various orientations assigned to each keypoint location based on image gradient vectors.

2.3.2.4. Keypoint Descriptor

The keypoint descriptor allows significant shift in gradient positions by creating orientation of histograms over 4x4 sample regions computed from a 16x16 sample array. To avoid boundary affecting the descriptor abruptly changing from one histogram to another, it uses trilinear interpolation to distribute the value of each gradient sample into histogram bins. Therefore, the formula to compute for adjacent histogram bins is,

(5) *Trilinear Interpolation* = *Sample Entry bin* * (1 - *d*)

where *d* is the distance of the sample from central value of the bin as measured in units of the histogram bin spacing

Equation (5) computes local image gradients to be measured in a region of a scale at a keypoint. The transformation allows significant levels of local shape distortion and change in illumination.

2.3.3. Image comparison and classification

Figure 7 displays the procedural presentation of the image recognition stage from extracting SIFT features of images for training to testing the SVM classifier for classifying unknown images. In the above diagram, it clearly shows the sequential procedure to achieve image recognition:

- a) Obtain sample leaf images for training
- b) After preprocessing the image, obtain SIFT feature keypoints
- c) Compute SIFT descriptors
- d) Cluster feature descriptors and create dictionary
- e) Create and train the Bag of Words (BoW), save as feature dataset, and train SVM
- f) Obtain unrecognized image for testing
- g) After preprocessing, extract its SIFT keypoints
- h) Compute the SIFT descriptors
- i) Compare and match the feature descriptor with the trained SVM classifier for leaf classification

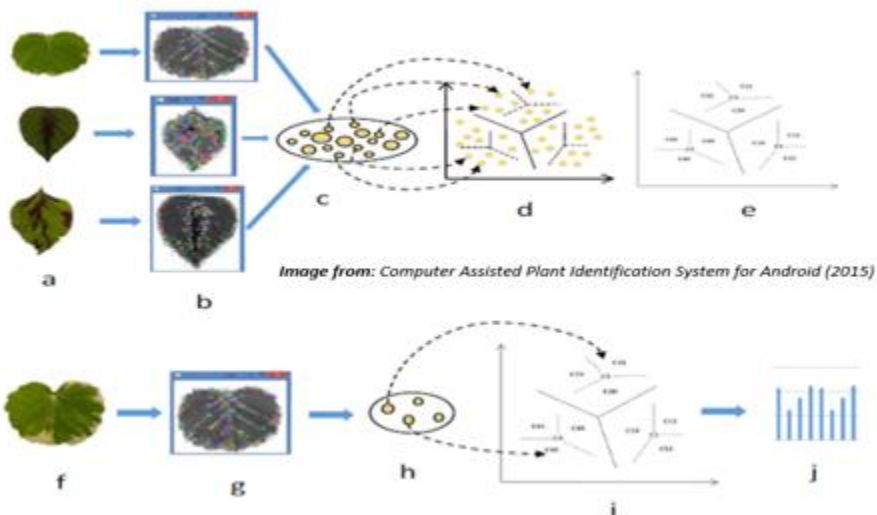


Figure 7. Image recognition phase from training through classification.

The training of tree leaves should be abundant and consistent with its physical attributes to avoid misclassification from the SVM. Python only recognizes the classes or BoW labels as numbers during classification wherein 0 – Acacia, 1 – Banaba, 2 – Mahogany, 3 – Molave, and 4 – Yemane. With the identification of the leaf, Python will throw the result back to the client server using the same method to communicate with the SQL server.

Support Vector Machine (SVM) uses the basic formulation for generating hyperplanes given by [13]:

$$(6) \quad f(x) = \beta_0 + \beta^T x$$

where β as the weight vector and β_0 as the bias

but can also be applied as,

$$(7) \quad |\beta_0 + \beta^T x| = 1$$

where x represents the training closest to the hyperplane

Given the distance of support vectors between a sample point of x and a hyperplane (β, β_0)

$$(8) \quad \min_{\beta, \beta_0} L(\beta) = \frac{1}{2} \|\beta\|^2 \text{ subject to } y_i(\beta^T x_i + \beta_0) \geq 1 \forall i.$$

where y_i represents each of the labels of the training sample

For the optimization of the vector margin, the Lagrange multipliers of Equation (8) will help obtain the weight vector β and the bias β_0 of the optimal hyperplane.

Frontend System (Graphical User Interface)

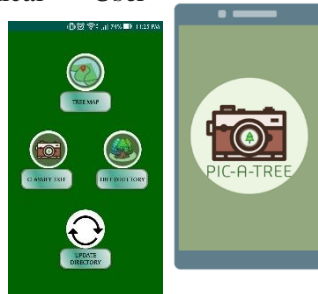


Figure 8. Application front page and main page.

Figure 8 shows the front page and main menu screen of the application. The main menu of the application displays 4 button functions. The first button on the left is the camera function, while the top button would be the map function, and the right button would be the tree directory. Finally, the last button is the update function for uploading and downloading tree data.



Figure 9 Application image request page.

Figure 9 shows the page when taking a picture of the tree leaf. The function allows user to capture leaf image, upload as request, and view request. The upload button requires internet connection and GPS to function properly.

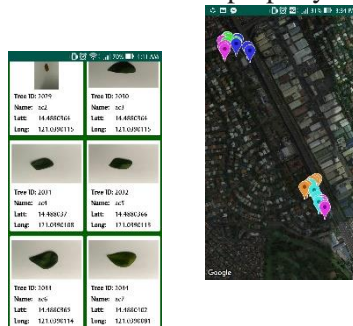


Figure 10. Application tree map and tree directory page.

Figure 10 shows the tree map and tree directory list of registered trees in the application. Google maps allows users to manipulate the area screen and view marked trees within the area, while the tree directory display all registered trees according to its species with an option of directing users where the tree resides.

3.0. RESULT AND DISCUSSIONS

3.1. Project Specification

Table I describes the procedural specification of the target functions. The first column contains the primary and secondary functions, followed by the theoretical or proposed system specification, and finally, the actual specification done. The comparison between the proposed and the actual specification reviews the summary of changes or introduced during the design and development phase of the prototype.

TABLE I. Function-Specification

Functions	Proposed Specifications	Actual Specifications
Acquire Tree Leaf Image	The leaf image will be acquired using the built-in camera in the selected phone at above 16mp	Image was captured using Sony Xperia XA Ultra phone with 21MP rear camera Captured images are compressed in JPEG format
Acquire	The geographic	Sony Xperia is supported by GPS, A-

Geographic Coordinates	coordinates that include both longitude and latitude will be acquired using the built-in GPS technology feature of smartphones	GPS, GLONASS, and/or BDS Google Maps is used as the API module for displaying geographical coordinates of captured and recognized tree leaves.	
Transmit Data	The data transmitted through the internet will send and receive data to and from the dedicated server	Images sent by mobile client devices will be queried as request data, will be stored in the web server database, and accessed remotely by Python	
Display Tree Details	The result will display the tree information to devices connected from the network server	The recognized tree leaves, stored from local and/or online database, will be displayed on their respective tree directories	
Display Graphical Tree Coordinates	The tree geographic location will be visual presented on the integrated API map of the application	The map module, supported by Google Map API, displays the GPS coordinates from the list of classified trees and respective tree species from the tree directories	
Pre-process Image	Convert RGB Image to Grayscale	The acquired leaf image will be converted from a colored image to grayscale image	Random images received from the web server via Python will be saved and converted into a grayscale image using BGR to gray conversion of python
	Remove Image Background	The grayscale image will be segmented by representing the ROI and the outside region as black and white	Gray images will undergo segmentation of the ROI using region growing method generating the ROI from a seeded pixel, and mask the image to remove the background
	Detect Image Edge	After segmentation, the function highlights and contours the potential features from the segmented image	Edges will be extracted from the segmented ROI of the previous image focusing only the pixel values having a 200 in the G(x) and G(y) intensity threshold vectors
	Resize Image	The modified image will use interpolation method to increase resolution while maintaining image	The edged image will be interpolated by multiplying dimensions by 2 times

	quality
Extract Features	The system will extract the different object-based unvarying keypoints and descriptors set as features Features are extracted using SIFT algorithm, and mass produce the SIFT keypoint descriptors from potential features found on the leaf edges
Compare Image Testing	The system will compare the processed image to different pre-defined images that will search for matching feature of tree species identities Image classification uses predefined datasets formed into a training dictionary reference for comparison with the unknown data sent by the client The trained dataset and features of an image is fitted and classified through the SVM

3.2. Project Evaluation

3.2.1. Accuracy Testing

The test aims to prove the accuracy of the system. The test conducted 20 samples of leaf images per specie with white background to attain optimum result. Adding more samples and training sample, in this case, higher than 20 per tree species, would certainly boost the accuracy towards a higher metric grade. By averaging the correctly identified samples from the number of trials of all 5 species, the metric is computed and arrived with a result of 91% accuracy.

3.2.2. Reliability Testing

The test aims to prove the reliability of the system. The precision level of the averaged 5 tree species does not diverse away from each other, therefore producing a more stable and comprehensible state in image processing. The reliability or the precision is computed using the difference of 100% and the standard deviation of the 5 tree species and produced a result of 93.2% reliability of the system.

3.2.3. User-friendly Testing

The test aims to prove the user’s ability to understand the system’s operation and how user-friendly the user interface. The survey is conducted to the foresters at the DENR MIMAROPA sector to measure the user-friendliness of the system. The rating would range from 1 to 5, which is disappointing to exceptional respectively. The survey consists of 6 questions related to the design project’s user-friendly features.

The application’s ease of usage, overall design and view of the interface, user’s overall experience, recommendation to a colleague and for the application, and insights regarding system’s process and functions are the main focus of the survey questions.

Overall design and the expected output received a positive feedback however ease of use of the application is quite low. The foresters also provided feedback with

application as a possible solution for tree identification and some possible effect in the operation or execution application if tree species will be increased in the database.

3.2.4. Consistency Testing

The test aims to prove the consistency of the system according to the distance of the camera with the target leaf. The test conducted 20 samples of leaf images per specie originally from the first evaluation. The distance values were estimated producing an average of 26.85cm rating, which likely falls on the satisfactory level.

3.2.5. Responsiveness Testing

The test aims to prove the responsiveness of the system according to the overall process time consumed from the frontend and backend system. The test conducted 20 samples of leaf images per specie originally from the first evaluation. The average process time resulted 52.7 seconds. Being inefficient in its performance, the system can be considered very responsive during the uploading of images.

TABLE II. Summary of Findings

Objective	Measured Parameter			Scale	Rating
	Min	Max	Ave		
Accuracy (%)	80	100	91	88.36% - 92.42%	Good
Reliability (%)	93.2			90.41% - 93.31%	Fair
Consistency (cm)	15	32.4	26.85	26cm- 49.99cm	Fair
Responsiveness of process time (seconds)	28.4	64.2	52.7	Above 5.56s	Poor
User-friendliness	3	4	3.5	3.00 to 3.99	Good

Table II shows the results evaluated from the testing and matched it with the proposed objective metrics. It shows the objectives, the measured parameters, rating, and scale. The findings were subjected for evaluation of results based on metric table reflected in Chapter 1 of this study. The measured parameters are actual values that has met the proponent's objectives for this study. The result for the accurate detection is 91%, which falls under the scale '88.36%-92.42%' evaluated as "Good" in the proponent's metrics. For the reliability, the result is 93.2%, which falls under the scale of '90.41%- 93.31%' evaluated as "Fair." Consistency is 26.85cm, which falls under the scale of '26cm to 49.99cm' evaluated as "Fair." For the responsiveness in terms of the application, the average duration of the system was 52.7 seconds, which is in the scale of 'Above 5.56 seconds' evaluated as "Poor." Most of the measured parameters for the objectives are in average values.

4.0. CONCLUSION AND RECOMMENDATIONS

4.1. Conclusions

- The system achieved 91 out of 100 correct identification considered as Good-Fair rating level.
- The system is prone to the luminosity undergoing in the image background and foreground
- The standard distance estimated for the camera distance arrived with a Fair result in consistency.
- The system utilized a private server hosted by Hostinger.com for uploading data online and resulted with 0 failure of transmission.
- From the 12-21mp camera, the response time delays in the backend system due to high resolution images but uploads faster at any network speed.
- The system prototype proves to be easily operated with a 3.5 rating in user-friendliness, evaluated as Good, based on the foresters' survey.

4.2. Recommendations

- Focus on indigenous tree species of the Philippines to assist the DENR foresters in their actual duties
- Aside from image identification, capture other tree characteristics such as wood and tree nomenclature
- Improve the UI and system code for stability and a more user-friendly application
- Add filtering methods in preprocessing, like luminance filtering, and HAAR Cascade technique to crop images, to achieve efficiency, robustness, and accurate results
- Focus on lower Android version, image resolution, or image quality
- Migrate the backend system into the cloud for efficient processing.
- Develop a server-less tree identification Android system in reference to current trends in technology

REFERENCES

- [1] V. Balasubramanian, K. Adalarasu and R. Regulapati, "Comparing dynamic and stationary standing postures in an assembly task", *International Journal of Industrial Ergonomics*, vol. 39, no. 5, pp. 649-654, 2009.
- [2] L. Rozakis, *Schaum's Quick Guide to Writing Great Research Papers*. New York: McGraw-Hill, 2007.
- [1] R. V. Oliva, "Philippine Forest Wildlife Law Enforcement: Situationer and Core Issues," *Philippine Environmental Governance 2 Project*, pp. 1-38, 2007.

- [2] M. Zaid, M. Shahid Akhtar, S. A. Patekar and M. G. Sohani, "A Mobile Application for Quick Classification of Plant Leaf Based on Color and Shape.," vol. 02, no. 10, pp. 70-80, 2015.
- [3] S. Shejwal, P. Nikale, A. Datir, A. Kadus, P. Bhade and R. Pawar, "Automatic Plant Leaf Classification on Mobile Field Guide," pp. 1-5, 2015.
- [4] S. Mouine, I. Yahiaoui, A. Verroust, L. Joyeux, S. Selmi and H. Goëau, "An Android Application for Leaf-based Plant Identification," no. 10.1145/2461466.2461520, 2013.
- [5] H. A. Chathura Priyankara and D. K. Withanage, "Computer Assisted Plant Identification System for Android," in *2015 Moratuwa Engineering Research Conference (MERCon)*, 2015.
- [6] H. Goëau, P. Bonnet, A. Joly, V. Bakic, J. Barbe, I. Yahiaoui, S. Selmi, J. Carré, D. Barthélémy, N. Boujemaa, Molino, Jean-François, G. Duché and A. Péronnet, "PI@ntNet Mobile App," in *Proceedings of the 21st ACM international conference on Multimedia*, 2013.
- [7] H. Goëau, P. Bonnet, A. Joly, A. Affouard, V. Bakic, J. Barbe, S. Dufour, S. Selmi, I. Yahiaoui, C. Vignau, D. Barthélémy and N. Boujemaa, "PI@ntNet Mobile 2014: Android port and new features," in *Proceedings of the ACM International Conference on Multimedia*, 2014.
- [8] T. Vijayashree and A. Gopal, "Authentication of herbal medicinal leaf image processing using Raspberry Pi processor," in *2017 International Conference on Intelligent Computing and Control Systems (ICICCS)*, 2017.
- [9] L. Zhang, P. Weckler, N. Wang, D. Xiao and X. Chai, "Individual leaf identification from horticultural crop images based on the leaf skeleton," *Computers and Electronics in Agriculture*, pp. 127:184-196, 2016.
- [10] J. D. S. Selda, R. M. R. Ellera, L. C. Cajayon and N. B. Linsangan, "Plant Identification by Image Processing of Leaf Veins," in *Conference: the International Conference*, 2017.
- [11] A. Mordvintsev and A. K., "Introduction to SIFT (Scale Invariant Feature Transform)," 2013. [Online]. Available: https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_feature2d/py_sift_intro/py_sift_intro.html. [Accessed 24 September 2018].
- [12] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, pp. 1-28, 2004.
- [13] OpenCV dev team, "Introduction to Support Vector Machines," OpenCV dev team, 23 September 2018. [Online]. Available: https://docs.opencv.org/2.4/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html. [Accessed 24 September 2018].