Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 1, 620-627
2025
Publisher: Learning Gate
DOI: 10.55214/25768484.v9i1.4187

Development of a mobile-based application for leaf disease recognition in abaca (Musa textilis)

Dialogo, G. G.[1*]
[1]College of Computer Studies, Eastern Samar State University-Salcedo Campus, Salcedo, Eastern Samar, Philippines; dialogogil1992@gmail.com (D.G.G.).

Abstract: Abaca is a significant crop valued for its durable fibers, widely used in various industries. However, abaca cultivation is threatened by diseases that can severely impact yields and quality. Hence, timely detection and accurate diagnosis of these diseases are crucial for effective management. Thus, this research tries to develop a tool that can detect and diagnose leaf diseases to aid farmers in disease mitigation. This research utilized the Waterfall model in the development of the application. Ten types of disease image datasets were collected from Kaggle. Three testing strategies namely benchmark, alpha, and beta tests were conducted to evaluate the application using ISO/IEC 25010 software quality metrics. During these tests, the application gained an overall mean of 3.49, 3.76, and 4.52 respectively. The findings revealed that the application is already usable and functional. Additional features may be included for easy mitigation in the widespread of the disease in future studies.

Keywords: *Abaca, Mobile application, Plant disease recognition, Waterfall model.*

1. Introduction

Abaca is one of the primary materials in the textile industry. Such abaca is one of the sources of income-generating projects at Eastern Samar State University in Salcedo, Eastern Samar. Meanwhile, abaca production encounters a great challenge regarding pest and disease control management. Pest and disease have been a great risk in the production of abaca since this kind of plant is prone to pest and insect destruction.

One of the challenges that farmers are facing in the abaca industry is the early suppression or identification of abaca plant diseases. Farmers who rely solely on the traditional means of identifying leaf diseases in abaca face a lack of time efficiency and practical solutions, which may result in the widespread occurrence of diseases [1]. Because bananas, especially abaca, are sensitive to several diseases and result in large losses for farmers, disease identification in these crops has proven more challenging in the field [2]. In effect, the occurrence of pests and disease destroys the entire plant. Moreover, if pests and diseases are not controlled, they spread, lessen, and threaten the production of Abaca in the locality.

Specifically, to combat this growing problem, the researcher initiated this study on the "Development of a Mobile-based Application for Leaf Disease Recognition in Abaca (Musa Textilis)" which significantly helps farmers identify diseases present in the abaca leaves. Thus, the application of artificial intelligence and computer vision in the recognition of diseases present in various plants has been studied and has been used in agriculture. These techniques have proven to perform accurately and efficiently [3]. Several studies have demonstrated the effectiveness of machine learning for better prediction of plant diseases with high accuracy as mentioned by Flores, et al. [4].

The implementation in the study of [1] has been advantageous for the early detection and identification of abaca plant diseases. This is supported by the study of Dey, et al. [5] which showed that automatic vision-based systems have a promising performance with easy validation. Moreover, a study by Sladojevic, et al. [6] achieved 96.3% accuracy in classifying plant diseases using CNNs. CNN

History: Received: 13 December 2024; Revised: 3 January 2025; Accepted: 7 January 2025; Published: 10 January 2025
* Correspondence: dialogogil1992@gmail.com

performs well in image recognition however, it encounters challenges because of some morphological features such as shape and color [7]. Similarly, Hughes, et al. [8] used smartphone images combined with machine learning to detect diseases in cassava, with an accuracy rate of 93%. These studies indicate the potential for similar applications in abaca disease detection. However, for some commercially valuable crops, such as abaca, this kind of technology used for the detection and identification of plant diseases has not yet been fully explored and tested [9].

Knowing the advantageous effect of computer vision and artificial intelligence, this research tries to develop a mobile application that will aid farmers in the detection of diseases present in abaca which will help in the mitigation of the spread of pests and diseases. Thus, the development of a mobile-based application for leaf disease detection in Abaca supports various United Nations Sustainable Development Goals (SDGs) by fostering innovation in agriculture, contributing to food security for local farming communities, promoting economic growth for farmers, reducing hunger through improved productivity through technology-based solutions in farming which can pave the way for more industrial applications, sustainably improving productivity and lastly encouraging sustainable environmental practices.

2. Methodology

2.1. Data Collection

The data were collected from Kaggle (https://www.kaggle.com/) which is an online repository of various datasets. Ten (10) types of disease were identified as classes which are Abaca Mosaic, Bunchy Top, Cordana leafspot, Rhizome Rot, Bacterial Wilt, Leaf Blotch, Freckle, Yellow Sigatoka, Infectious Chlorosis, and Spinach Blight. One hundred (100) images for each species were used for training the model.

2.2. Conceptual Framework

Figure 1 presents the conceptual framework of the developed mobile application. It gives an overview of how the system will work. It starts with the mobile camera capturing an image of the abaca leaf which serves as the input to the classification model embedded in the mobile application. Once the image is fed to the model, the application will output the type of disease which is the result of the prediction.

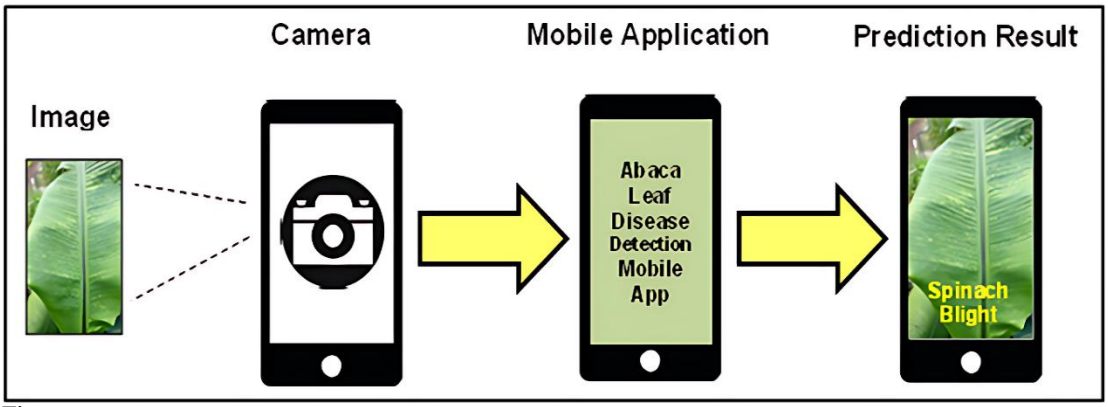

Figure 1.
Conceptual framework of the developed mobile application.

2.3. Research Design

Waterfall Model- System Development Life Cycle has been used in the development of the application because of its simplicity and ease of use. Figure 2 presents the theoretical framework of the Waterfall Model. There is no overlap between the model's phases, and each one must be finished before the next can start.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 1: 620-627, 2025
DOI: 10.55214/25768484.v9i1.4187

The first phase constitutes the collection, evaluation, and development of requirements. It is a vital step in the design and development of a system. Following the system's development, the researchers tested the created application with evaluators in various tests. Finally, some errors or design issues may arise during operation and maintenance. For continuity, all of the errors must be debugged. According to Gabry [10] modifying the system, involved repeating some or all of the previous phase.

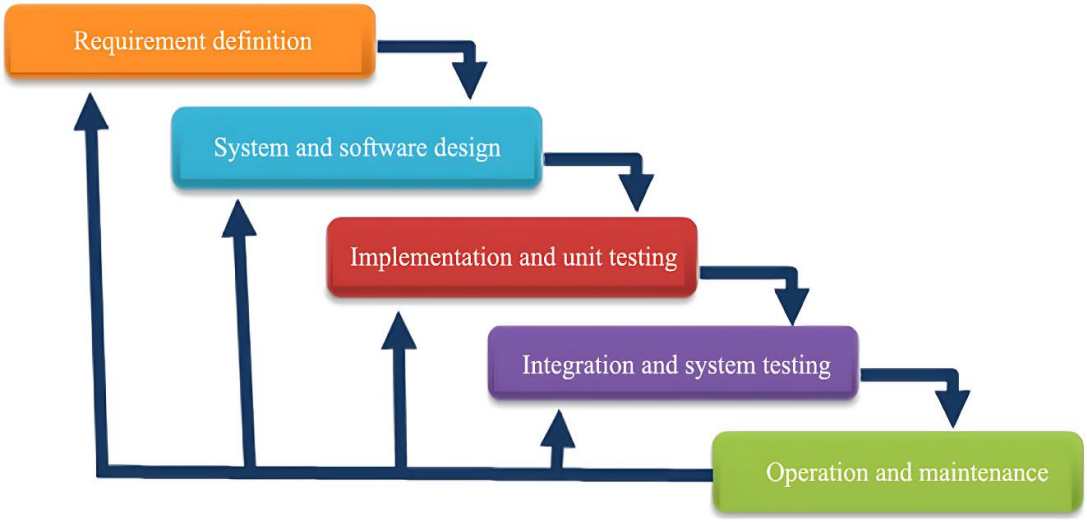

Figure 2.
Waterfall model – software development life cycle (SDLC).

2.4. System Development

The researcher used Kaggle (https://www.kaggle.com/), an online collection of diverse datasets, to collect photos of abaca leaves with diseases to construct the algorithm. The researcher gathered the necessary hardware and software requirements. Making flowcharts and data flow diagrams followed as a guide in the development. By operationally getting all the acquired data, these served as inputs in the application's development.

The mobile application was designed and developed with the aid of Android Studio. Before the application was made available to end users, it was tested to make sure that it was free of problems and that the design was appropriate. The investigator employed a range of testing methodologies, including benchmark, alpha, and beta tests, to address all identified issues and fulfill user requirements. Since they were the primary testers and the application's final users, the instructors, students, and farmers assessed this series of tests.

2.5. System Deployment

The end user needs a mobile device running Android OS (at least Marshmallow) to install the generated application. Shareit, Bluetooth, and other file transfer apps can be used to transfer the produced application.

2.6. Hardware Requirements

The hardware requirements utilized in the development and deployment of the program are displayed in Table 1. A faster CPU and a larger hard drive were needed for the application's development. Additionally, a keyboard and mouse were needed for the developer for program coding, and click mouse buttons to carry out program operations. Users are required to have smartphones to be able to navigate the application.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 1: 620-627, 2025
DOI: 10.55214/25768484.v9i1.4187

Table 1.
Hardware requirements.

Hardware	Minimum requirements	Actual used
Development		
Processors	AMD Ryzen 3 5300U, 2600 Mhz, 4 Core(s), 8 Logical Processor(s)	AMD Ryzen 3 5300U, 2600 Mhz, 4 Core(s), 8 Logical Processor(s)
RAM	At least 4GB	4GB
Hard disk drive	At least 50 GB	150 GB
Mouse	Any Mouse	Acer mouse
Deployment		
Android phone	Any smartphone	Realme mobile

2.7. Software Requirements

The software specifications utilized in the development and deployment of the application are shown in Table 2. The Microsoft Operating System (OS) is required for the application to interface with the computer's hardware during development. Android Studio program was needed for the design and development of the application. Android OS-powered mobile phones were utilized for the application's deployment.

Table 2.
Software requirements.

Software	Minimum Requirement	Actual Requirement
Development		
Android Studio IDE	Android Studio v. 3.2	Android Studio v. 3.2
Microsoft OS	Windows 7	Windows 10
Deployment		
Android OS	Android 6.0 (Marshmallow)	Android 8.0 (Oreo)

3. Results and Discussion

3.1. Acceptability Test

The application has undergone a series of tests to evaluate its quality after the development period. This made it possible for the researcher to check whether the specifications made during the development phase had been satisfied. The evaluators check what requirements have been met and need improvement.

The Benchmark test was conducted initially. After the Benchmark test is the Pilot test. It consists of Alpha and Beta tests. In the audio-visual room of the College of Computer Studies (CCS) building, ten (10) faculty members of the CCS participated in an alpha test, scoring the developed application by utilizing the ISO scorecard. During the last test, which was the beta test, farmers who were the application's end users assessed it using the same scorecard.

The application was evaluated using ISO scorecards. The functionality, reliability, usability, efficiency, maintainability, and portability are the six (6) qualities or criteria that make up the software quality model described in the ISO/IEC 25010 standard. This focuses on evaluating a software product's internal quality.

Table 3 presents the categories, codes, and descriptions of the 5-point Lickert rating scale:

Table 3.
System usability scale.

Scale	Code	Description
4.20 – 5.00	5	Excellent
3.40 – 4.19	4	Very Good
2.60 – 3.39	3	Good
1.80 – 2.59	2	Poor
1.00 – 1.79	1	Very Poor

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 1: 620-627, 2025
DOI: 10.55214/25768484.v9i1.4187

Frequency counts and a weighted grading system was used to assess the collected data. The frequency counts were used to calculate weighted means, and the comparable statistical representation's average was used to calculate the overall mean score.

The results of the different testing strategies conducted are shown in Figure 3. The six (6) criteria in the ISO/IEC 25010 software quality metrics were evaluated in every test.

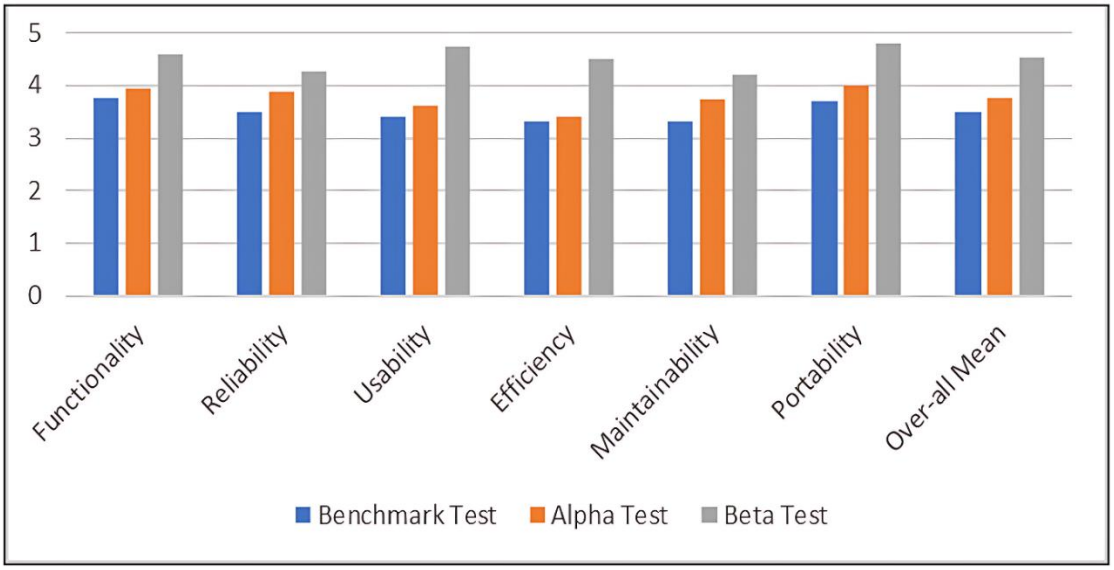

Figure 3.
Acceptance testing results.

Functionality. Functionality is defined as adaptability as the quality of being well-suited to fulfill a purpose in real-world situations. The application gained a total score of 3.75 on the Benchmark test and 3.95 on the Alpha test which is interpreted as "Very Good". However, during the Beta test conducted by the end-users, it achieved a mean score of 4.60 which is interpreted as "Excellent". The scores reveal that the application has been enhanced in terms of its functionality from the various testing strategies conducted and is already functional.

Reliability. The ability to be dependable or consistently perform well was defined by the reliability perspective. The developed application achieved 3.50, 3.87, and 4.25 scores from the different tests conducted namely benchmark, alpha, and beta test in terms of its Reliability. Both Benchmark and alpha tests are interpreted as "Very Good" while the last test conducted which is the Beta test is interpreted as "Excellent". This shows that the application is reliable in terms of the outputted results.

Usability. The usability perspective defines the extent to which anything is suitable or able to be used. Based on the tests conducted, the application obtained mean scores of 3.40 and 3.60 for the Benchmark and Alpha tests which is interpreted as "Very Good". On the other hand, during the last test conducted, the developed application received a grand mean of 4.75 from the end-users which is interpreted as "Excellent". This finding shows that the application is usable and user-friendly.

Efficiency. Efficiency was defined as a state or attribute by the Efficiency perspective. It includes the resources utilized and the processing time the application takes. The result during the Benchmark and the Alpha test shows that the applicated was rated with a mean score of 3.30 and 3.60 which is interpreted as "Very Good" while on the Beta test, it gained an overall mean score of 4.50 which is interpreted as "Excellent". It shows that the applications perform and give the results efficiently.

Maintainability. The maintainability perspective describes the likelihood of completing a repair task within a certain time frame. The application received an overall mean of 3.30 and 3.40 during the Benchmark test and Alpha test respectively which is interpreted as "Very Good". On the other hand,

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 1: 620-627, 2025
DOI: 10.55214/25768484.v9i1.4187

during the Beta test, the application obtained a grand man of 4.20 which is interpreted as "Excellent". The result means that the application has improved from the first test until the last test.

Portability. Portability is the capacity of software to be installed and shared with other devices. During the conduct of the tests, the developed application achieved an overall mean of 3.70 in the Benchmark test and 4.0 in the Alpha test which is interpreted as "Very Good". Lastly, the application gained a grand mean of 4.80 during the beta test which is interpreted as "Excellent". Based on the findings of the study, shows that the application can be transferred, shared, and run on any platform.

Result Summary. From the three testing strategies, the application gained an overall mean during the Benchmark test of 3.40 and 3.72 during the Alpha test. Lastly, it achieved a grand mean of 4.52 on the Beta test. It implies that the application has been modified and upgraded based on the suggestions of the different respondents.

3.2. Screenshots

Figure 4 displays the main interface of the application. The user will select whether he/she will choose an image from the phone gallery or may take real-time images through a mobile phone camera.

Figure 4.
Main interface of the application.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 1: 620-627, 2025
DOI: 10.55214/25768484.v9i1.4187

Figure 5 (a) illustrates the interface for cropping the image. Once the image has been selected, it will let the user crop the image as to what region of the image will be used for detection. Figure 5 (b) presents the interface where recognition of the disease takes place. It displays what type of disease is present in the abaca leaves.

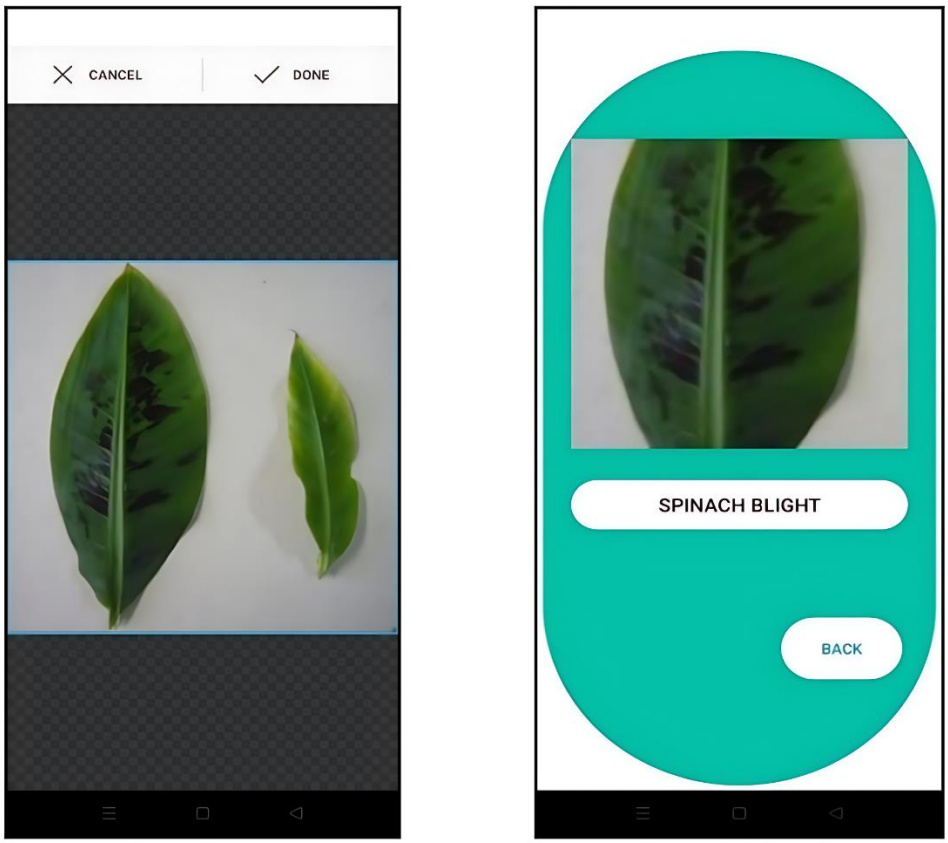

Figure 5.
(a) Interface for cropping the image (b) Interface for disease recognition.

4. Conclusion and Recommendations

4.1. Conclusion

The research utilized the Waterfall model System Development model which served as a guide in the development of the application. An image dataset was collected from Kaggle to be used during the model's training. There were ten (10) types of diseases included in the study. Android Studio was utilized for designing and development of the mobile application. Three testing strategies namely benchmark, alpha, and beta tests were conducted. Different participants evaluated each test to ensure the quality of the product. ISO/IEC software quality metrics were utilized to assess the application. In the benchmark test, the application gained an overall mean of 3.49 while on the alpha test, it obtained an overall mean of 3.76. On the other hand, the application received an overall mean score of 4.52. It implies that the application has been tested and gained improvements in every test conducted. It revealed that the application is user-friendly and portable as it can work offline to detect abaca leaf diseases. The application is already usable and functional and is ready to be deployed to the end-user. The development of a mobile-based application for leaf disease detection in Abaca helps foster innovation in agriculture, promoting economic growth for farmers, reducing hunger through improved productivity, and encouraging sustainable environmental practices in the Abaca industry. A further

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 1: 620-627, 2025
DOI: 10.55214/25768484.v9i1.4187

study may be conducted which includes additional features for easy mitigation of the spread of the disease.

4.2. Recommendation

The conclusions have led to the introduction of the following suggestions to provide suitable outcomes:

1. Upon thorough testing of the application, the researcher recommends to the institution for the utilization of the application to aid the farm workers in management in the abaca farm of the university.
2. The user must download and install the application to experience and navigate the application.
3. A further study may be conducted which includes recommendations or possible actions to mitigate the spread of the disease of abaca.

Acknowledgment:

The author owed his esteemed status to the following individuals who in many ways had helped in the conduct of the study: Dr. Mir-i-nisa Y. Boco, Engr. Renato A. Padullo, Dr. Hershey R. Alburo and Prof. Charito B. Lacasa.

References

[1] L. T. Buenconsejo and N. B. Linsangan, "Classification of healthy and unhealthy Abaca leafs using a convolutional neural network (CNN)," presented at the IEEE 13th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM), Manila, Philippines, 2021.

[2] K. Seetharaman and T. Mahendran, "Leaf disease detection in banana plant using gabor extraction and region-based convolution neural network (RCNN)," *Journal of The Institution of Engineers (India): Series A*, vol. 103, no. 2, pp. 501-507, 2022. https://doi.org/10.1007/s40030-022-00628-2

[3] M. I. R. Abucejo and J. C. Cuizon, "Sequential pattern mining of tourist spatiotemporal movement," *Recoletos Multidisciplinary Research Journal*, vol. 9, no. 1, pp. 69-78, 2021. https://doi.org/10.32871/rmrj2109.01.07

[4] G. Flores, A. Figueroa, R. Tumamak, and N. J. M. Berdon, "A sound-based machine learning to predict traffic vehicle density," *Recoletos Multidisciplinary Research Journal*, vol. 9, no. 1, pp. 55-62, 2021. https://doi.org/10.32871/rmrj2109.01.05

[5] A. K. Dey, M. Sharma, and M. Meshram, "Image processing-based leaf rot disease, detection of betel vine (Piper Betle L.)," *Procedia Computer Science*, vol. 85, pp. 748-754, 2016. https://doi.org/10.1016/j.procs.2016.05.262

[6] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic, "Deep neural networks based recognition of plant diseases by leaf image classification," *Computational Intelligence and Neuroscience*, vol. 2016, no. 1, p. 3289801, 2016. https://doi.org/10.1155/2016/3289801

[7] G. G. Dialogo, L. S. Feliscuzo, and M. E. A., "Fish species detection application (FiSDA) in Leyte Gulf using convolutional neural network," *Proceedings of Engineering and Technology Innovation*, vol. 19, pp. 16–27, 2021. https://doi.org/10.46604/peti.2021.7892

[8] D. Hughes, A. Ramcharan, K. Baranowski, P. McCloskey, B. Ahmed, and J. Legg, "Deep learning for image-based cassava disease detection," *Frontiers in Plant Science*, vol. 8, pp. 1-7, 2017. https://doi.org/10.3389/fpls.2017.01852

[9] U. B. Patayon and R. V. Crisostomo, "Automatic identification of abaca bunchy top disease using deep learning models," *Procedia Computer Science*, vol. 179, pp. 321-329, 2021. https://doi.org/10.1016/j.procs.2021.01.012

[10] O. E. Gabry, "Software engineering – software process and software process models (Part 2)," Retrieved: https://www.medium.com. 2017.