GLOBUS PUBLICATION

Globus An International Journal of Medical Science, Engineering and Technology
A Refereed Research Journal
Vol 14 / No 1 / Jan-Jun 2025 ISSN: 2319-8958

"BUKOFIER": IMAGE RECOGNITION TECHNIQUES FOR EFFICIENT DIAGNOSIS OF COCONUT LEAF DISEASES: A MACHINE LEARNING APPROACH FOR COCONUT FARMERS' COOPERATIVE IN TANAY RIZAL

*Renjun A. Orain
**Merlin Marfa
***Elizabeth Sta. Ana
****Ian Tiao

Paper Received: 25.11.2024 / **Paper Accepted**: 14.12.2024 / **Paper Published**: 15.12.2024
Corresponding Author: Renjun A. Orain; **Email:** renjun.orain@email.lcup.edu.ph; doi:10.46360/globus.met.320251005

Abstract

One of the leading crops grown in the Philippines is coconut. It is a common occurrence that this crop can contract various diseases in the Philippines. Untrained individuals may have difficulty determining the type of disease because some illnesses are easily mistaken for others of the same kind. It is also time consuming and expensive to observe coconut leaf diseases on a farm. The goal of this project is to create a mobile application called Bukofier that uses computer vision to detect coconut leaf diseases. Through computer vision technology and convolutional neural network, this application gives the user capabilities to detect and diagnose the five common diseases in coconut leaves. After a diagnosis, the Bukofier also offers the recommended controls and treatments that can aid in stopping the disease's progress. Bukofier aims to assist coconut growers and owners in recognizing the disease and receiving information, reference pictures, and treatments or alternatives to control it.

Keywords: Coconut Tree, Diseases Detection, Computer Vision, Convolutional Neural Network, Image Processing.

Introduction

Coconut farming plays a vital role in the agricultural landscape of the Philippines, contributing significantly to the economy and the livelihoods of millions of farmers. As the world's largest producer of coconuts, the Philippines relies heavily on this crop, which is cultivated primarily on medium-sized farms. However, coconut growers face numerous challenges, particularly in the identification and management of diseases that affect coconut leaves. These diseases can lead to substantial crop losses if not diagnosed and treated promptly. Unfortunately, many farmers lack the necessary training to accurately identify these diseases, often resulting in misdiagnosis and ineffective treatment strategies.

In response to these challenges, this study introduces the Bukofier application, a cutting – edge tool designed to assist coconut growers in the detection and diagnosis of common coconut leaf diseases. Utilizing advances computer vision techniques and Convolutional Neural Networks (CNN), Bukofier aims to provide accurate disease identification, reference images and tailored treatment recommendations. By leveraging technology, the application seeks to empower farmers with the knowledge and resources needed to effectively manage crop health, ultimately enhancing productivity and sustainability in coconut farming.
This research not only addresses the pressing need for reliable disease detection methods but also aims to contribute to the broader field of agricultural technology. By integrating innovative solutions into traditional farming practices, the study aspires to improve the overall resilience of coconut farmers against the challenges posed by plant diseases, thereby supporting their economic stability and the sustainability of the coconut industry in the Philippines.

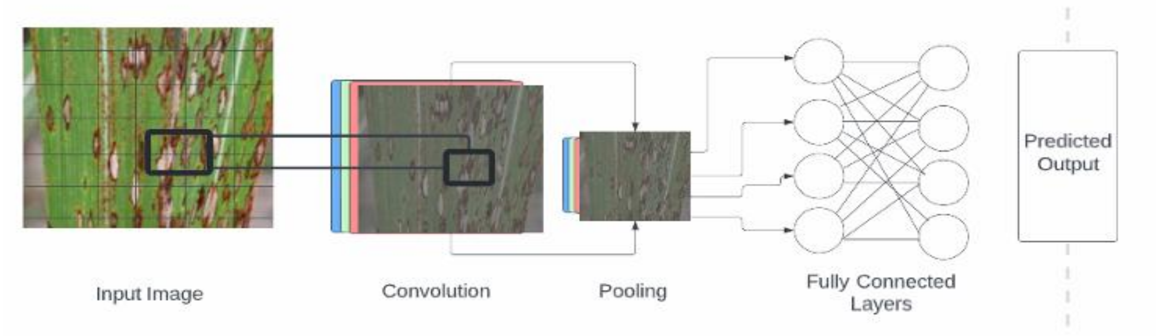
Theoretical Framework

The architecture of CNN is composed of neural networks that contain updating weights and biases. CNN image classification goal is to learn features

*-****Far Eastern University - Alabang, Philippines.

from the dataset and use it as a feature map for classification. The three main processes of CNN are Convolution, Non-Linearity and Pooling. In the figure 1 diagram, the input image is made up of kernels or pixels with their respective value. The coconut leaf images will enter the first feature extraction in the convolutional layer. The convolved coconut leaf image will then be reduced in size in the pooling layer. The value from the feature extraction layers will then be flattened or turned into a vector value to be passed to the fully connected layers. Fully connected layers compromise multiple neural networks with weights and biases. It will then predict if the coconut leaf image is healthy or contaminated with a disease.

Figure 1: CNN Diagram Overview



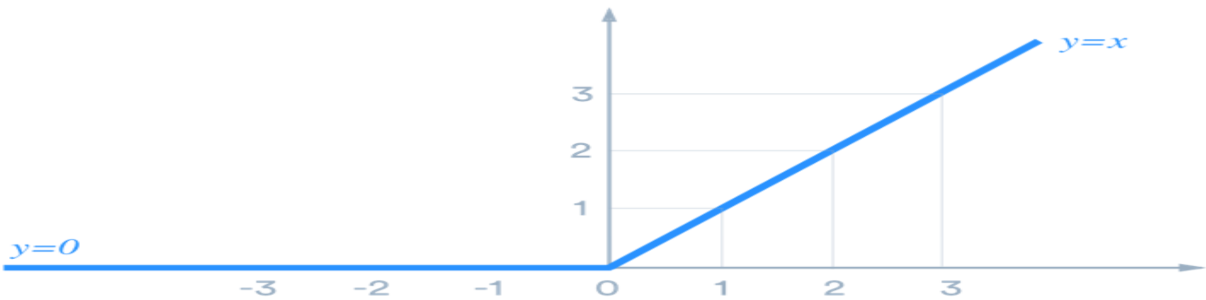
The Convolutional layer is the main layer of CNN, it convolves the image into a filter. It is a linear operation that multiplies a set of weights with the input image in the form of small matrix numbers. According to Dr. Sabili, this layer uses a filter that performs convolution while scanning the input which results in a feature map or activation map. As shown in figure 2, the $z_{i,\, j,\, k}$ represents the neuron output denoting row I and column j in the feature map k. The vertical and horizontal strides are represented by s_h and s_w. The height and width of the reception field are represented by f_h and f_w. The number of previous feature maps is represented by f_n. Bias will be represented by b_k.

Figure 2: Neuron Output Computation for Convolutional Layer

$$z_{i,j,k} = b_k + \sum_{u=0}^{f_h-1} \sum_{v=0}^{f_w-1} \sum_{k'=0}^{f_{n'}-1} x_{i',j',k'} \cdot w_{u,v,k',k} \quad \text{with} \begin{cases} i' = i \times s_h + u \\ j' = j \times s_w + v \end{cases}$$

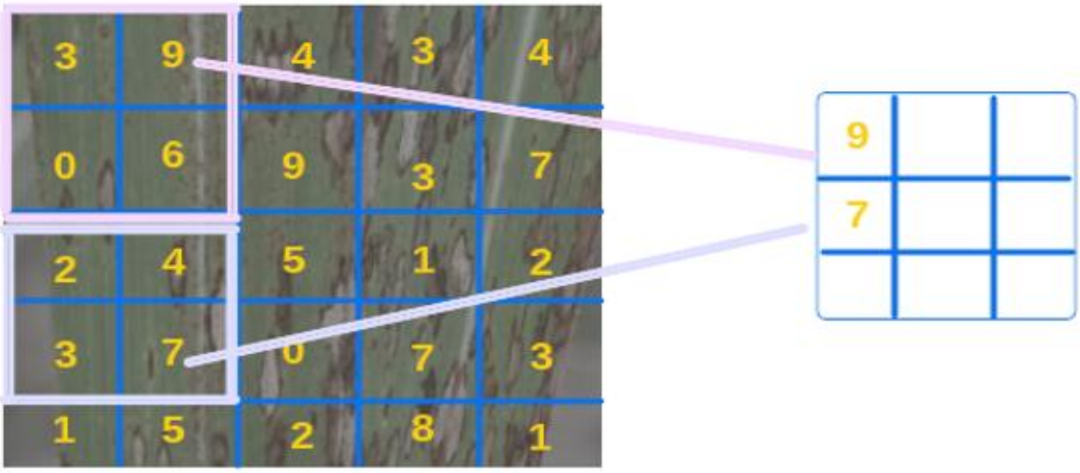
Rectified Linear Unit (ReLU) is a non-linearity activation function that will be applied after every convolution layer. ReLU is commonly used in CNN to process feature maps before the values are passed onward. It ensures that only node with a positive value will be passed onward. The value will either be zero or the value of x. if the x value is negative, it will be considered a zero value. This activation function makes sure that the minimum value fed is zero. As shown in the graphical representation in Figure 3, ReLU returns to zero value if any negative values are received. If the value is positive, then it will return the value back. It only activates when the input is greater than zero, The activation function formula of Rectified Linear Unit: **ReLU = max (0, x)**

Figure 3: Graphical Representation of ReLU



The Pooling layer goal is to shrink or subsample the feature maps to reduce computational load and the number of parameters. It aggregates the inputs using an aggregation function such as max or means. Max pooling is a type of pooling layer that searches for the largest or maximum input in the region to be saved in a new output feature map. As shown in the graphical representation in Figure 4, max pooling is being applied to shrink the size of the image.

Figure 4: Graphical Representation of Max Pooling

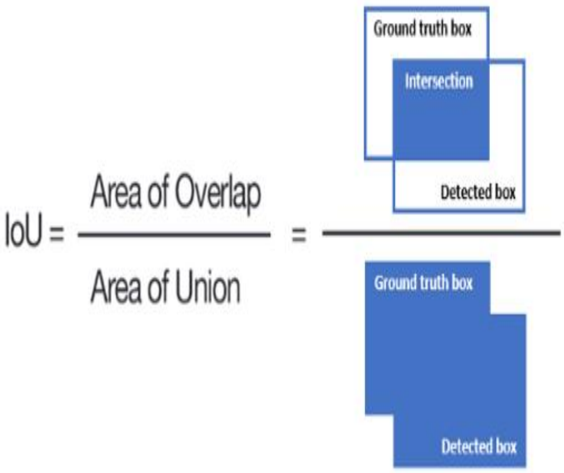


The process between the Convolution layer and the Pooling layer is the feature learning process. The Fully Connected (FC) layer is the classification layer. According to Tiao, after the feature learning process, it needs to classify the values into various classes, this can be done using Fully Connected Neural networks. This layer will also give the final probabilities for each class's output. The formula below shows the calculation of each layer in the neural network. The input vector with dimension is represented by x. The weight matrix is represented by W. The bias vector is represented by b. The activation function (ReLU) will be represented by g. The calculation formula for each neural network layer: **g (Wx + b)**

In the classification layer, a function called SoftMax has been used which outputs represent a categorical probability distribution. SoftMax is a mathematical function that converts vectors into probability values. The function provides which output has the highest probability among the classes. The formula below represents the SoftMax function. The number of classes is represented by K. The symbol e zi is the standard exponential summation of the value of each class. The activation formula for SoftMax:

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

Using bounding boxes will help identify which part of the input is being detected. According to the purpose of a bounding box is to help the users detect which object is being classified. It also reduces the search range of object features. The box will contain the coordinates and information of the object being detected. The coordinate of the area of interest will be extracted as its center coordinates. The center coordinates will help determine the anchor box. According to Fernandez, IoU quantifies the two boxes of ground truth and prediction. It can be calculated by taking the ratio between the area of the intersection and the area of the union of the two bounding boxes.



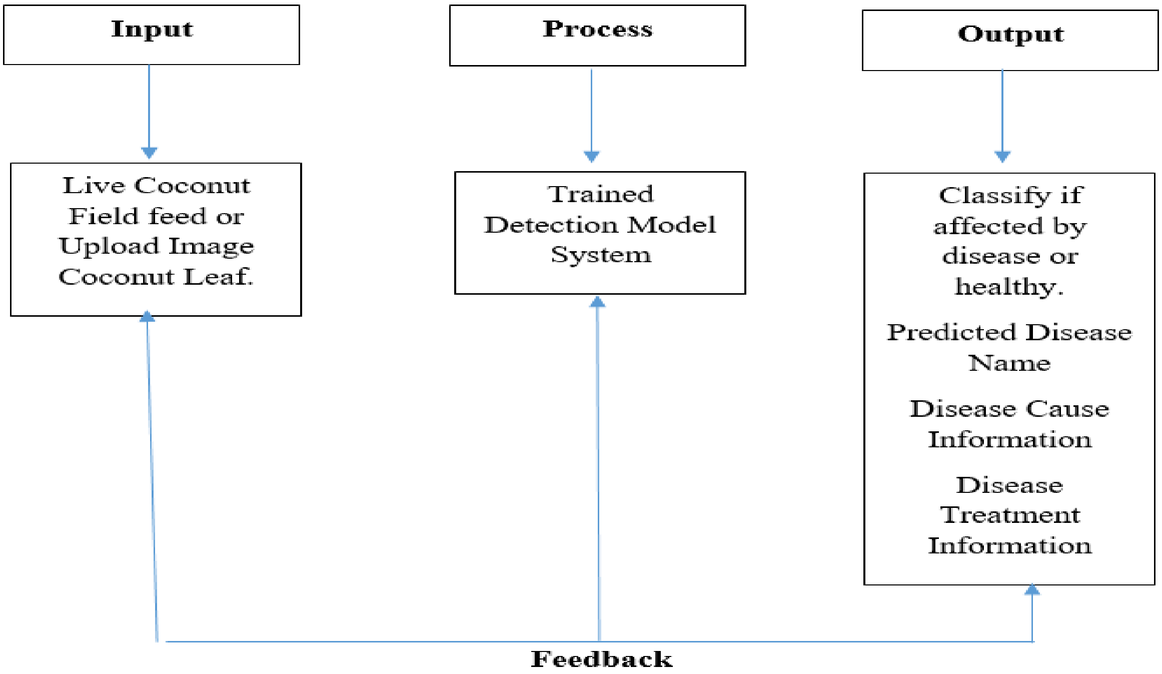
Conceptual Framework

The study used the input, process, output strategy or the IPO diagram **Input box** shows live coconut field feed or upload image of the lead of Coconut Leaf. On the other hand, the **process box** contains the validation of questionnaires, statistical treatment of data and analyses of the data gathered through the questionnaires using the following statistical tools: percentage, frequency distribution, weighted mean, and multiple regression analysis. It also contains the trained detection Model System.

Informal interviews were employed to supplement the data elicited for the questionnaires and the tabular and narrative presentation of findings and interpretations.

The **output box** reflects all the actual results of the study, classification if affected by disease or health as well as predicted disease name, disease cause information and disease treatment information.

Figure 5: Conceptual Framework of the Study



Statement of the Problem

The researcher aimed to determine the status, development and evaluate the Bukofier: Image Recognition Techniques for Efficient Diagnosis of Coconut Leaf Diseases: A Machine Learning Approach for Coconut Farmers Cooperative in Tanay Rizal. Specifically, it sought to answer the following questions:

1. How to say the following features be developed and integrated in system in terms of:
   1.1 Computer Vision in detecting coconut leaf diseases.
   1.2 Utilization of CNN (Convolutional Neural Network)
   1.3 Image Data Set
2. How many prototyping be utilized as the software development methodology in the study?
3. How may the system be tested in terms of:
   3.1 White Box Testing
   3.2 Black Box Testing
   3.3 Integration Testing
4. How to evaluate the Detection of Coconut Leaf Diseases using Computer Vision with CNN using FURFS software evaluation criteria in terms of:
   4.1 Functionality
   4.2 Usability
   4.3 Reliability
   4.4 Performance
   4.5 Supportability

Scope and Delimitation

Scope

The study focuses on developing and evaluating the Bukofier app to detect five coconut leaf diseases in the Tanay Province. The app uses CNN to identify diseases from images captured by mobile or drone cameras. The study also evaluates the apps' usability and collects data to train the CNN model. An agile development methodology is used for iterative improvements.

Delimitations

The study is limited to Tanay Province and focuses on five specific coconut leaf diseases. The research will not address diseases not found in the Philippines or issues related to poor image quality. The study will adhere to ethical guidelines, ensuring participant confidentiality and obtaining informed consent.

Methodology

Research Design
The research aims to develop a mobile application using CNN to detect coconut leaf diseases and provide disease information. The goal is to help coconut growers and crop protection experts identify and manage these diseases, which are causing significant damage to the Philippine coconut farms. The agile methodology is used to ensure efficient and adaptive development, incorporating user feedback and delivering a high – quality product.

Participants/Respondents of the Study
The study will involve 30 respondents from Tanay Province, including coconut growers, plant pathologists and agriculturists. The participants must have experience in identifying coconut leaf diseases and be knowledgeable in plant pathology and crop protection.

Instrument/s of the Study
A survey questionnaire was used to collect data from respondents, including coconut growers, plant pathologists and agriculturists. The questionnaire was validated by experts and used a Likert scale to measure satisfaction. The accuracy of the disease detection system will be validated by Dr. Mark Angelo Balendres.

Data Collection and Analysis
The researcher gathered coconut leaf diseases images through web scraping and requests from the Institute of Plant Breeding at UPLB. Data augmentation was used to increase the dataset size. The researcher also conducted interviews with plant pathologists and distributed questionnaires to validate the data and ensure software quality. The predictive disease output will be validated by Dr. Mark Angelo Balendres.

Ethical Considerations
The researcher will maintain thorough records, respect participant dignity, and obtain informed consent. Data confidentiality will be ensured, and plagiarism will be avoided by citing sources. The research will be conducted independently, and biases will be minimized to ensure accurate and impartial results.

Results and Discussion
The researcher will present the tallied survey results gathered from the respondents. The survey is composed of thirteen (13) questions that are based on the Functionality, Usability, Reliability, Performance and Supportability Models.

Legend:

5 – Very Satisfied (VS)

4 – Satisfied (S)

3 - Neutral (N)

2 – Dissatisfied (D)

1 – Very Dissatisfied (VD)

Table 1: Functionality Test Tally and Average Score

Items	VD	D	N	S	VS	Average
Did the application's feauture correctly identify the disease?	0	5	8	13	4	3.53
Did the application immediately determine the disease after the detection?	0	2	10	8	10	3.87
Did the application properly identify the possibile treatment?	0	3	6	16	5	3.77
Total Average						**3.73**

The average for each item is calculated using standard mean calculation, given that all the responses for Functionality were totaled and compiled. Functionality survey items had an average value of 3.73 as Satisfied.

Table 2: Usability Test Tally and Average Score

Items	VD	D	N	S	VS	Average
Does the application's buttons are user-friendly and easy to understand?	0	1	5	13	11	4.13
Is the application easy to use?	0	1	4	15	10	4.13
Is the application useful?	0	0	6	10	14	4.27
Total Average						**4.18**

The average for each item is calculated using the standard mean calculation, given that all the responses for Usability were totaled and compiled. Usability survey items had an average value of 4.18 interpreted as Satisfied.

Table 3: Reliability Test Tally and Average Score

Items	VD	D	N	S	VS	Average
Did the application disease detection feature provides an accurate prediction?	0	4	16	10	0	3.2
Does the application provide correct treatments and measures for the identified disease?	0	5	6	15	4	3.6
Did the application detection feature provide accurate and reliable prediction?	0	5	10	14	1	3.37
Total Average						**3.39**

The average for each item is calculated using the standard mean calculation, given that all the responses for Reliability were totaled and compiled. Reliability survey items had an average value of 3.39 interpreted as Neutral.

Table 4: Performance Test Tally and Average Score

Items	VD	D	N	S	VS	Average
Does the application respond quickly?	0	4	13	11	2	3.37
Is the application consistent on accurate detection?	0	6	13	11	0	3.17
Total Average						**3.27**

The average for each item is calculated using the standard mean calculation, given that all the responses for Performance were totaled and compiled. Performance survey items had an average value of 3.27 interpreted as Neutral.

Table 5: Supportability Test Tally and Average Score

Items	VD	D	N	S	VS	Average
Is the application easy to install and configure?	0	0	5	17	8	4.1
Is the application 's "Guide / Help" feature useful?	0	0	4	15	11	4.23
Total Average						**4.17**

The average for each item is calculated using the standard mean calculation, given that all the responses for Supportability were totaled and compiled. Supportability survey items had an average value of 4.17 interpreted as Satisfied.

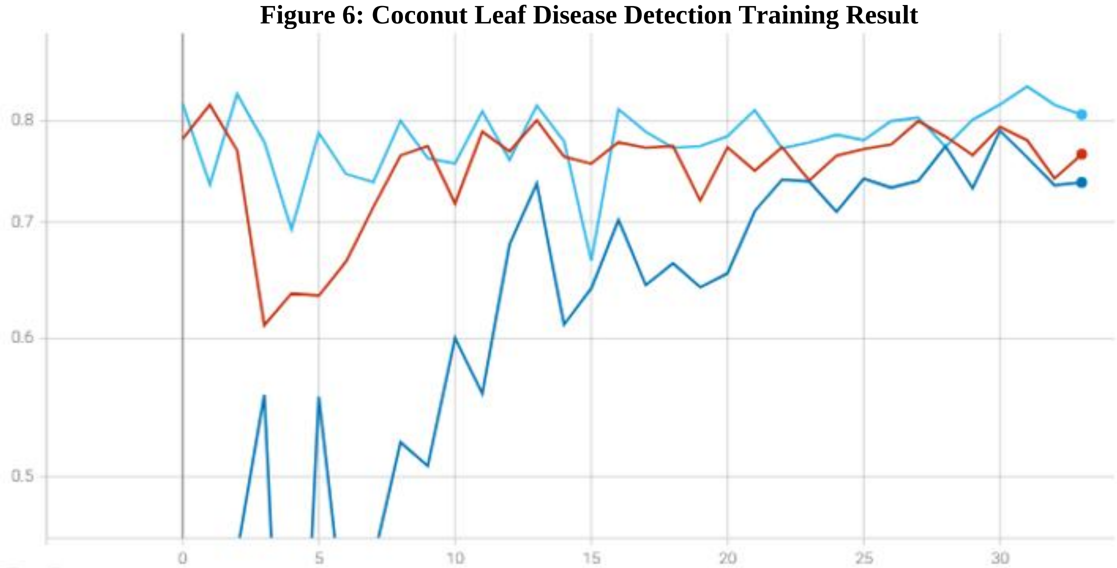
Table 6: Total Average Score

Criteria	Total Respondents	Score	Interpretation
Functionality	30	3.73	Satisfied
Usability	30	4.18	Satisfied
Reliability	30	3.39	Neutral
Performance	30	3.27	Neutral
Supportability	30	4.17	Satisfied
Total Average Score		3.75	Satisfied

The table shows the total average score based on the FURPS criteria. It shows that the total average score for the application based on the average scores of all categories is 3.75 which is interpreted as Satisfied.

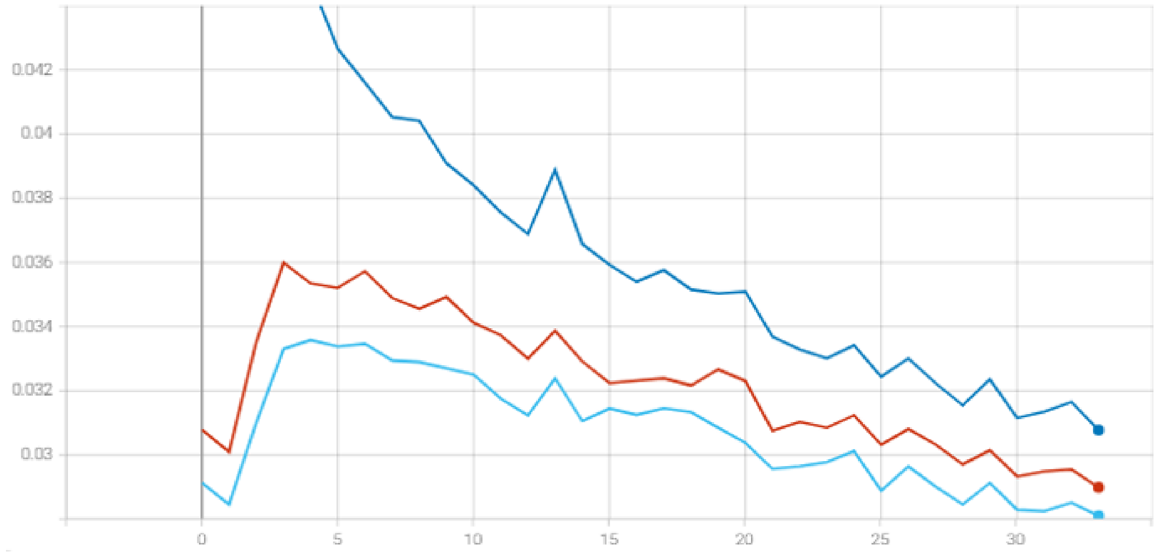
Model Training Results

Figure 6: Coconut Leaf Disease Detection Training Result



The graph represents the precision metric of the training in detecting coconut leaf diseases. The data being trained consists of six (6) classes which are healthy, leaflets, caterpillars, yellowing, drying and flaccidity. The training consists of three runs with 34 epochs each. It reached an 83% accuracy rate during the 31st epoch of the third run. There is a possibility of overfitting during the 3rd run due to the lack of dataset.

Figure 7: Loss of Bounding Boxes



The graph above shows the loss of the bounding boxes. The loss represents the measurement of errors. Each run gradually decreases in value until it reaches a loss value of 0.028. The lower the value of a loss metric the better a model will perform. The researcher stopped the training due to the possibility of overfitting based on the inference of the saved model.

Conclusion

Based on the results that were gathered by the researcher, the Bukofier application achieved a satisfactory rating from the target respondents. The researcher managed to create an object detection model that can detect five types of coconut diseases and a healthy class of coconut leaf. The model was integrated into an application which can detect and classify coconut leaf diseases using a phone or drone camera. The application also has a database that stores the predicted output from an uploaded image or a captured image. The CNN model reached an 83% accuracy rate with a 0.028 loss value. Based on the survey results, and model evaluation, the CNN model and application still need more improvements for more accurate detection and a satisfactory interface.

Recommendations

Despite being a satisfactory application, more improvements must be made to the CNN model and application. According to the responses, crop experts suggest that Bukofier detect complex and dominant diseases and provide a list of possible causal organisms and treatments. They also indicate that Bukofier must differentiate the symptoms for abiotic and pathogen diseases. The visual mapping of disease severity or incidence is also recommended for large – scale farms. The researcher recommends preparing at least a minimum of 2000 unique images for each class without the use of data augmentation. Data augmentation will not be enough if the prepared datasets have only 50 below unique images. The researcher also recommends finding a drone that is suitable for connecting with a mobile phone. Different drone systems must be able to connect with the main source code. The researcher also recommends using a sturdy drone as some of the coconut fields are susceptible to strong wind and rainy weather. The researcher also recommends testing different hyperparameters and optimizers for each batch of trail training. The researcher recommends using a higher batch size than 16 and higher epochs than 64 for better results. The researcher also recommends using GPU rather than CPU as an environment for training a model.

Conflicts of Interest

The authors declare that there is no conflict of interest in this manuscript. Results have been deduced objectively; nothing influenced this result by any individual, monetary, or institutional interests. In effect, utmost endeavors were in order to conduct a bias-free operation of gathering and analyzing the data into interpretation and producing a study based on merit, away from research inconsistencies.

References

1. Burns, E., & Brush, K. (2021). What is Deep Learning and How Does It Work?. Tech Target. Retrieved from https://www.techtarget.com/searchenterpriseai/definition/deep-learning-deep-neural-network
2. Sanadzakirizvi, M. (2020). CNN Image Classification: Image Classification Using CNN. Retrieved from https://www.analyticsvidhya.com/blog/2020/02/learn-image-classification-cnn-convoluti on-neural-networks-3-datasets/
3. Biswal, A. (2022). Convolutional Neural Network Tutorial. Simply Learn. Retrieved from https://www.techtarget.com/searchenterpriseai/definition/deep-learning-deep-neural-network
4. Amidi, A., & Amidi, S. Convolutional Neural Networks cheatsheet. CS 230 - Deep Learning. Retrieved from https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks
5. West M. (2019) Convolutional Neural Networks: The Theory. Bouvet Norge. Retrieved from https://www.bouvet.no/bouvet-deler/understanding-convolutional-neural-networks-part- 1
6. Brownlee, J. (2019). A Gentle Introduction to Pooling Layers for Convolutional Neural Networks. Deep Learning for Computer Vision. Retrieved from https://machinelearningmastery.com/pooling-layers-for-convolutional-neural-networks/
7. Singh, A. (2020). Demystifying the Mathematics Behind Convolutional Neural Networks (CNNs). Analytics Vidhya. Retrieved from https://www.analyticsvidhya.com/blog/2020/02/mathematics-behind-convolutional-neural-network
8. Ognjanovski, G. (2019). Everything you need to know about Neural Networks and Backpropagation - Machine Learning Easy and Fun. Towards Data Science. Retrieved from https://towardsdatascience.com/everything-you-need-to-know-about-neural-networks-and-backpropagation-machine-learning-made-easy-e5285bc2be3a
9. Brownlee, J. (2020). Softmax Activation Function with Python. Machine Learning Mastery. Retrieved from https://machinelearningmastery.com/softmax-activation-function-with-python/
10. Johnson, D. (2022) Back Propagation Neural Network: What is Backpropagation Algorithm in Machine Learning? Guru99. Retrieved from https://www.guru99.com/backpropogation-neural-network.html
11. Dabbura, I. (2017). Gradient Descent Algorithm and Its Variants. Towards Data Science. Retrieved from https://towardsdatascience.com/gradient-descent-algorithm-and-its-variants-10f652806a3
12. Boureau, YL., Le Roux, N., Bach, F., Ponce, J., & Lecun, Y. (2011). IEEE 2011 IEEE international conference on computer vision (ICCV) - Barcelona, Spain (2011.11.6–2011.11.13). 2011 international conference on computer vision - ask the locals: multi-way local pooling for image recognition; p. 2651–8.
13. Zeiler, MD., & Fergus R. (2013). Stochastic pooling for regularization of deep convolutional neural networks. Eprint Arxiv. arXiv:1301.3557.
14. Zeiler, M. D., and Fergus, R. (2014). Visualizing and understanding convolutional networks. CoRR abs/1311.2901. DOI: 10.1007/978-3-319-10590-1_53
15. Al-Hiary, H., Bani-Ahmad, S., Reyalat, M., Braik, M., & ALRahamneh, Z. (2011). Fast and accurate detection and classification of plant diseases. International Journal of Computer Applications 17:31–38. DOI:10.5120/ijca.
16. Poornam S., & Devaraj, A. (2021). Image-based Plant leaf disease detection using Deep learning. International Journal of Computer Communication and Informatics. Retrieved from https://pdfs.semanticscholar.org/74e0/df8198bf0c8cb0284f2d48561b36c472257b.pdf
17. Hasan, R., Yusuf, S., & Alzubaidi L. (2020). Review of the State of the Art of Deep Learning for Plant Diseases: A Broad Analysis and Discussion. Plants, 9(10), 1302. DOI: 10.3390/plants9101302
18. Mostafa, A., Kumar, S., Meraj, T., Rauf, H., Alnuaim, A., & Alkhayyal, M. (2022). Guava Disease Detection Using Deep Convolutional Neural Networks: A Case Study of Guava Plants. Applied Sciences (2076-3417), 12(1), 239, 19p. DOI: 10.3390/app12010239
19. Arunnehru, J., Vidhyasagar, B.S., & Anwar Basha, H. (2020). Plant Leaf Diseases Recognition Using Convolutional Neural Network and Transfer Learning. International Conference on Communication, Computing and Electronics Systems. Lecture Notes in Electrical Engineering, 637. DOI:10.1007/978-981-15-2612-1_21
20. Lu, J., Tan, L., & Jiang, H. (2021). Review on Convolutional Neural Network (CNN) Applied to Plant Leaf Disease Classification. Agriculture, 11(8), 707. DOI: 10.3390/agriculture11080707
21. Kulkarni, A., & Patil, R. (2012). Applying image processing techniques to detect plant

diseases. Int J Mod Eng Res, 2 (5) pp. 3661-3664.
22. Jadhav, S., Udupi, V., & Patil, S. (2021). Identification of plant diseases using convolutional neural networks. Int. j. inf. technol. 13, 2461–2470. DOI: 10.1007/s41870-020-00437-5
23. Mahlein, A., Oerke, E., Steiner, U., & Dehne, H. (2012). Recent advances in sensing plant diseases for precision crop protection. Eur J Plant Pathol 133, 197-209. DOI: 10.1007/s10658-011-9878-z
24. Singh, V., & Misra, A. (2017). Detection of plant leaf diseases using image segmentation and soft computing techniques. Information Processing in Agriculture, 4(1), 41-49. https://doi.org/10.1016/j.inpa.2016.10.005
25. Pathak, A., Pandey, M., & Rautaray, R. (2018). Application of Deep Learning for Object Detection. Procedia Computer Science, 132, 1706-1717. DOI: 10.1016/j.procs.2018.05.144
26. Militante S., Gerardo B., & Medina R. (2019). Sugarcane Disease Recognition using Deep Learning. IEEE Eurasia Conference on IOT, Communication, and Engineering (ECICE). DOI: 10.1109/ECICE47484.2019.8942690
27. What is Applied Research? + [Types, Examples & Method]. (2020). Formplus. Retrieved from https://www.formpl.us/blog/applied-research
28. Software Product Development: Definition, Types, Methodologies & Process! (2021). Bit Blog. Retrieved from https://blog.bit.ai/software-product-development/
29. Peek, S. (2022). What Is Agile Scrum Methodology? Business News Daily. Retrieved from https://www.businessnewsdaily.com/4987-what-is-agile-scrum-methodology.html
30. Losby, J., & Wetmore, A. (2012). CDC Coffee Break: Using Likert Scales in Evaluation Survey Work. National Center for Chronic Disease Prevention and Health Promotion. Retrieved from https://www.cdc.gov/dhdsp/pubs/docs/cb_february_14_2012.pdf
31. Arby, F.H., & Amin, I.H.A. (2022). Implementation of YOLO-v5 for real-time Social Distancing Detection. Journal of Applied Informatics and Computing (JAIC), 6(1), 01-06.
32. Infolks Pvt Ltd. (2019). How Does Bounding Box Enables Object Detection?. Retrieved from https://medium.com/tech-break/how-bounding-box-enables-object-detection-999b3059 974e
33. Anwar, A. (2022). What is Average Precision in Object Detection & Localization Algorithms and how to calculate it?. Retrieved from https://towardsdatascience.com/what-is-average-precision-in-object-detection-localizati on-algorithms-and-how-to-calculate-it-3f330efe697b
34. Hui, J. (2018). mAP (mean Average Precision) for Object Detection. Retrieved from https://jonathan-hui.medium.com/map-mean-average-precision-for-object-detection-45 c121a31173