CHAPTER 1 INTRODUCTION TO THE STUDY

Background of the Study

The Philippines is one of the wealthiest countries in terms of agriculture. As an agricultural country, there are 10 million rice farmers in the Philippines. Palay production in the country went up to 19.32 million metric tons in 2020 from 19.07 million metric tons in 2018, or an average annual increase of 0.7 percent. 19.32 million metric tons of palay was posted in 2020 (PSA, 2021).

Nowadays, farmers face grave problems in rice plants due to diseases that affect the quality and quantity of the crops. According to Arida (2010), some causes of lower production rate are the lack of expert availability in farming, insufficient knowledge in fertilizer management, and unawareness of the diseases and pests. In addition, diseases in plants contribute indirectly and directly to particular environmental damage. Plants are affected mainly by various bacterial and fungal infections. Tungro, Leaf Blight, and Blast are among the Philippines' most common rice leaf diseases. Most of these diseases in rice crops appear as a spot on the leaves. Thus, it is necessary to diagnose the

disease correctly and on time to avoid significant harm to the rice crops.

This study proposed a mobile application that captures and recognizes images of rice leaf diseases using object detection algorithms. The object detection algorithm allows the application to detect rice leaves through the camera. Machine Learning (ML) technique is used to recognize the disease. Convolutional Neural Network (CNN) is the algorithm used to take image as input, assign importance (learnable weights and biases) to various aspects/objects in the image, and differentiate one disease from the other. In addition, it provides recommendations for the best approach in dealing with the disease. This mobile application aims to help the farmers become aware of the illnesses instantly as early diagnosis of rice disease is of utmost importance.

Theoretical Framework

Relevant theories

The theory of E. Gibson's Feature Theory (1966) states that Complex Stimuli are composed of distinctive and separable parts called features. Gibson's pattern recognition was

achieved by counting the presence or absence of the checklist feature.

According to this theory, a unique object is composed of a special feature determined by patterns. This theory leads the researchers to develop a disease detection and classification on rice leaves by determining the disease patterns on the datasets loaded.

Further, Beiderman's Recognition-By-Components Theory (1987) proposes recognition in humans that accounts for the successful identification of objects despite changes in the size or orientation of the image. The significant contribution of RBC is the proposal that the visual system extracts geons (geometricions) can be used to identify objects. Geons are simple volumes such as cubes, spheres, cylinders, and wedges. RBC proposes that representations of objects are stored in the brain as structural descriptions.

In this theory, the object identified by geons refers to an object's geometric components. The researchers believe that the geometric components of an object can increase the accuracy of determining the result of the disease of the rice.

Objectives of the Study

This study aimed to develop a mobile application that can detect diseases (e.g., bacterial leaf blight, tungro, and leaf blast) on rice leaves.

Specifically, the researchers aimed to:

1. Collect datasets and generate a classifier (model) for rice leaf disease detection.

2. Develop a mobile application that can capture and determine the disease on a rice leaf.

3. Provide suggestions/recommendations for a rice leaf disease diagnosis.

4. Evaluate the accuracy and precision of the generated model and the application's performance.

Significance of the Study

The results of the study will benefit the following:

Farmers and Agriculturists. The results will enable the farmers to understand the diseases suffered by rice crops in the early growth stage and better understand the conditions

4

related to the variety of rice. Aspiring agriculturists must have the knowledge of what they are dealing with in terms of rice crops. The abundance of the crop may depend on the type of disease it suffers. As technology grows, people tend to rely on the internet to find solutions to their problems. This app will benefit aspiring farmers and enthusiasts to monitor the possible diseases of rice crops every day as symptoms arise.

Students and Future Researchers. The data gathered and analyzed by the researchers will be used as a basis for how to further the study of identifying the rice disease using image analysis. It could be utilized to develop a better solution and a strategy for dealing with the issue of agriculture, specifically rice crops. At the end of this study, the results will be essential for the farmers, agriculturists, students, and future researchers to gain insight and guide them as diseases in their rice crops occur. In addition, this study will be used to formulate solutions to help specifically the agricultural sector problems.

Definition of Terms

For better understanding, the following terms were defined conceptually and operationally:

Bacterial Leaf Blight: Is caused by Xanthomonas oryzae pv. oryzae. It causes wilting of seedlings and yellowing and drying of leaves (IRRI Rice Knowledge Bank, 2021).

In this study, bacterial leaf blight was classified as one of the diseases in the rice leaves and could be identified and classified using the application.

Diagnosis*:* The process of determining the nature of a disease or disorder and distinguishing it from other possible conditions (Encyclopedia Britannica, 2018).

In this study, the diagnosis was classified as one of the system's outputs, identifying the kind of disease found. It can provide suggestions on managing, identifying, and why and where it occurs.

IDE: An IDE, or Integrated Development Environment, enables programmers to consolidate the different aspects of

writing a computer program. IDEs increase programmer productivity by combining everyday activities of writing software into a single application: editing source code, building executables, and debugging (https://www.codecademy.com/article/what-is-an-ide).

In this study, IDE is used in writing computer programs such as Gradle SDK, Apache Cordova, Java SDK, Node JS, and Programmer's Notebook.

Image Classification: A supervised learning problem: define a set of target classes (objects to identify in images) and train a model to recognize them using labeled example photos (https://developers.google.com/machinelearning/practica/image-classification).

In this study, image classification was used to recognize the rice leaf diseases uploaded and captured using the application.

Image Detection: A computer technology that processes the image and detects objects in it (https://azati.ai/).

In this study, image detection was one of the application's processes to identify rice leaves via live camera or upload image.

Leaf Blast: Caused by the fungus Magnaportheoryzae. It can affect all above-ground parts of a rice plant: leaf, collar, node, neck, parts of panicle, and sometimes leaf sheath (IRRI Rice Knowledge Bank, 2021).

In this study, leaf blast was a type of disease in rice leaves and could be identified and classified using the application.

Machine Learning: A method of data analysis that automates analytical model building. It is a branch of artificial intelligence based on the idea that systems can learn from data, identify patterns and make decisions with minimal human intervention (SAS Institute Inc, 2021).

In this study, machine learning is the method used in training datasets to generate models linked to the application that will help identify and classify images being taken and uploaded.

Mobile Application: A type of application designed to run on a mobile device. It can be a smartphone or tablet computer. Even if apps are usually small software units with limited function, they still manage to provide users with quality services and experiences (Mroczkowska, 2021).

In this study, a mobile application was an output that could be used on smartphones because it uses a rear camera to capture rice leaf images.

Tungro: Caused by the combination of two viruses transmitted by leafhoppers. It causes leaf discoloration, stunted growth, reduced tiller numbers, and sterile or partly filled grains (IRRI Rice Knowledge Bank, 2021).

In this study, tungro was defined as one of the diseases found on rice leaves.

Delimitation of the Study

This study primarily focused on detecting and classifying rice leaf diseases and identify its natural causes during the early growth stage. Through a mobile application, it is able to detect rice leaf diseases using a

smartphone camera or uploaded images. The study only puts emphasis on the identification of three rice leaf diseases namely: bacterial leaf blight, tungro and leaf blast.

The application can only be used on rice leaves and if used on different crops, it will not classify because the datasets used to generate the model are those of rice leaves. The system is able to interpret live capture and the clearer the image captured, the accurate the results and diagnosis.

CHAPTER 2 REVIEW OF RELATED STUDIES

Review of Existing and Related Studies

Different detection systems have been made for society to create a better and a productive solution in various fields. Identifying the images of pests and diseases is one motivating factor in making an application or a system. Pests and diseases are the main reason of agriculture-related problems contributing to the economy's loss.

In the study of Dengshan Li et al. (2020), a recognition method for rice plant disease and pests video detection based on deep convolutional networks was developed. The work discussed several factors affecting rice diseases include climatic conditions, lightning conditions, humidity, nutrients, fertilizer, water management, and different farming conditions. The study presented at that the time consumption in detecting and identifying diseases might affect the recognition accuracy and later mislead and give an improper diagnosis and misuse of pesticides. Furthermore, their work proposed a deep learning-based video detection architecture with a backbone for detecting plant diseases and pests, transforming the video into still images, sending the

frame to the still-image detector for detection, and synthesizing the structure into the video to quickly identify the diseases and pests. Using the faster-RCNN, they used image-training models to detect relatively blurry videos. They compared the different backbones to determine the most suitable detection of the untrained rice videos to have the most accurate detector for rice and pests' disease. Their results shows that faster-RCNN is more accurate than VGG16, and the ResNet-101 backbone system.

The proposal of Convolutional Neural Network (CNN), Region-based Convolutional Neural Network (R-CNN), Fast-RCNN, and Faster-RCNN developed the image object detection. One of the most effective methods in image processing is CNN because it uses convolutional methods to extract image features, achieving high-level fusion of semantics and deep extraction of features through multi-level networks. Video object detection has also developed. Video detection remains different from still-image detection in many ways. Changes in depth of field often lead to video defocus. Swift motion, such as leaf motion caused by wind, often leads to motion blur; because of the camera angle change, the leaves in the

foreground may occlude the lesion spots in the background, which leads to part occlusion.

Deep learning methods have been used in the recognition and detection of crop diseases. Techniques such as the grayscale method, image processing method, and the mean pixels value method, and many recognition and detection methods have been attempted in feature extraction and image segmentation of crop diseases by deep learning.

As Dengshan Li et al., (2020) analyzed the results of their study, some lesion spots in the videos are not detected or detected incorrectly, making the detection confidence not very high.

Meanwhile, in the study of Bhatt, Sarangi, and Pappula (2019), recognition of health conditions in crops helps to perform necessary treatment for the plants. Disease detection and recognition were automated to help the locale in estimating severity and spread on plantations. This automation is a small step closer to making a robotic farm and spraying shortly. Because of the improvement in Deep Neural Networks, Bhatt, P. V., Sarangi, S., and Pappula, S. (2019) leveraged the neural network-based method to perform the accurate and fast detection of diseases and pests in tea

leaves. Evaluating the various feature extraction networks and detection architectures aims to develop an accurate yet efficient detector in speed and memory. Images used to assess models have different resolutions, quality, brightness, and focus as captured by mobile phones with other cameras through a participatory sensing approach. The experimental detection system results effectively identify and locate the health condition of the tea leaves in the complex background with occlusion.

Their work have provided the information for disease diagnosis online and the edge devices like mobile phones to offer personalized advice to farmers based on the reported events. Bhatt, Sarangi, and Pappula encountered crucial parts in identifying the pests and diseases they distributed to scouting mobile applications. The farmers will submit a different image of diseased plants. More than one person is involved in data collection, challenges of uncontrolled conditions like varying backgrounds, resolutions, illumination, camera angle, and image quality taken. Research on detecting diseases is carried out for a long time, and there are several techniques to identify plant properties and

stress due to different conditions. Different methods include:

- Laboratory-based chemical analysis of the plant's infected area.
- Study of visible light.
- Assessment of hyperspectral or multispectral images.

However, research on pictures taken using a regular red, green and blue (RGB) camera provides accurate results, it gives real-time solution, computationally efficient, and low-cost. Recent advances in detection and localization methods based on detector categories such as Region-based Convolutional Neural Network (RCNN), You Only Look Once (YOLO), Region-based Fully Convolutional Network (R-FCN), and Single Shot Multibox Detector (SSD) have proven useful to get highly accurate results on such images.

"Plant Diseases and Pests detection based on Deep Learning: A Review" study by Liu and Wang (2021) discussed that Deep Learning had become a superior method researchers used to address the issues of great concern. Disease and pest detection on plants is a technology that uses machine vision equipment in acquiring an image to analyze and judge whether

> **Commented [u2]:** Kindly check this sentence.

there are pests or diseases present on the collected images. Machine vision-based equipment has been initially applied in agriculture and replaced the traditional naked eye identification.

The authors compared the deep learning technique to the traditional methods. The study outlines the difference of the network structure from three different aspects such classification network, detection network, and segmentation network, and the advantages and disadvantages. Also, the performances of other studies are analyzed because the survey may discuss the possible challenges in the practical applications of plant diseases and pests' detection based on deep learning. They also proposed potential solutions and research ideas for the challenges and several suggestions.

Deep learning unifies end-to-end feature extraction with broad development prospects and great potential compared to traditional image processing methods. With the development of artificial intelligence technology, research that focuses on plant disease and pests' detection based on machine vision shifted from classical image processing and machine learning methods to deep learning methods. Although the rapid development of plant disease and pests' detection

16

technology has been moving from academic research into an agricultural application, there is still a distance from the mature application in the actual natural environment. Problems are still to be solved. To fully explore the potential of this technology, joint efforts from the experts of relevant disciplines are needed. To effectively integrate the experience knowledge of agriculture and plant protection with deep learning algorithms and models to make plant disease and pests detection based on deep learning mature.

Meanwhile, in the study of Mohanty, Hughes, and Salathe (2016) "Using Deep Learning for Image-Based Plant Disease Detection", their work combined the factors of widespread smartphone penetration, high definition (HD) cameras, and high-performance processors in mobile devices leading to a situation where the diagnosis of plant diseases based on automated image recognition was made possible and available at an unprecedented scale. Using the deep convolutional neural network (CNN) to train a public dataset of 54,306 images of diseased and healthy plant leaves classifying crop species and disease status of 38 different classes, containing 14 crop species and 26 diseases that their work achieves an accuracy of over 99%. A total of 60 experimental

configurations used deep learning architecture such as AlexNet and GoogLeNet, Transfer Learning, and Training from Scratch as their training mechanism. Experimental structures used Color, Greyscale and Segmented datasets.

In the study of Prajapati, H., Shah, J., and Dabhi, V., (2016), they detected and classify different rice plant diseases namely the (1) Bacterial Leaf Blight, (2) Brown Spot, and (3) Leaf Smut. The authors captured images of infected rice plants using a digital camera from a rice field. They then evaluated four techniques of background removal and three techniques of segmentation. Collection of data happened on a rice farm where datasets of leaves on white background are prepared. Their work applies the concepts of Machine Learning and Image Processing to solve the problem of automatic detection and classification of diseases of the rice plant, which is one of the important foods in India. Image processing operations is applied on the external appearances of infected plants.

Literature about Rice Leaf Diseases

According to PhilRice, they identified 4 Major Diseases of rice leaf namely the Blast, Tungro, Sheath Blight, and Bacterial Blight.

Blast or also known as *mata-mata (Bisaya), agupaw (Waray), and taya-taya (Cebuano in Mindanao)*, is described having elliptical and spindle-shaped spots with brown border and grey center; spots enlarge and band together resulting in drying of leaf. The factors that favor the disease are: susceptible variety, high relative humidity, long dew period, cloudy sky and intermittent rain, excess nitrogen and potassium fertilizers, and aerobic rice environment and soil moisture.

Tungro, also known as *tungro* in most dialects, is described as mottled young leaf, older leaves and yellow to yellow orange: stunted growth with slightly reduced number of tilers. The factors that favor the disease are susceptible variety, presence of infected field, especially with green leafhoppers vectors, continuous and asynchronous planting, absence of fallow period, and crop age (vegetative to tillering stage).

Sheath blight, also known as *labhagsa pal-ak(Cebuano), and masot (Pangasinan)*, is described as greenish-gray and oval spots near water lines: later enlarge and become grayish white with brown margin. The lesion spreads to the upper leaf sheaths and on leaves that come in contact with infected plant parts. Banded brownish lesion with gray-white center; lesions coalesce leading to blighting of leaf; affects panicle exertion when flag leaf is infected. The factors that favor the disease are susceptible variety, high humid and warm temperature, frequent rains, high rates of nitrogen fertilizer, and dense or close planting.

Meanwhile, Bacterial Blight, also known as *nauga nga dahon*, *natala nga tanum (Cebuano), nadurot nga tanum (Waray), naggapula (Ilonggo)*, is tagged when wilting of leaves of young plant, later entire plant dies (known as kresek). Watersoaked stripes along upper leaf margins; lesions enlarge and turn yellow in a few days affecting one or both sides of the leaf; later, lesions cover the entire leaf blade exhibiting white to grayish growth of saprophytic fungi; flag leaf may also be infected. The factors that favor the disease are susceptible variety, excess nitrogen and oxygen and magnesium increase disease incidence, phosphorus and potassium deficiency

Commented [u3]: Is this the correct word?

increase disease incidence, root and leaf injuries enhance bacterial invasion of plants, cloudy and humid conditions, wind and rain speed up the spread, and community flooded fields.

Meanwhile, In the study entitled: "Resilience and Adaptability of Rice Terrace Social-ecological systems: A Case Study Of A Local Community's Perception in Banaue, Philippines" Castonguay, Burhard, Muller, Horgan, and Settele, stated that according to the farmers, other pests, e.g., planthoppers (Nilaparvata lugens), leaffolders (Nephotettix spp.), leafrollers (Cnaphalocrocis medinalis), and stem borers (Sesamia inferens, Scirpophaga spp., and Chilo spp.), or diseases, e.g., leaf blight and tungro virus disease, have only a minor impact on rice production in the area.

In the article of Samanthi (2019), it discusses the difference between the two rice leaf diseases namely the Bacterial Leaf Blight and Bacterial Leaf Streak. The key difference is that the bacterial leaf blight causes wilting of seedlings as well as yellowing and drying of leaves, while bacterial leaf streak causes small, water-soaked, thin, yellow to brown color linear lesions on leaves.

Bacterial Leaf Blight is one of the serious bacterial diseases affecting rice and other crops. The causative agent of bacterial leaf blight is Xanthomonas oryzae pv. oryzae in rice. This bacterium enters the plant through wounds or stomata. The symptoms of this disease are wilting of seedlings, yellowing and drying of leaves. When this bacterium infects at the early stages of the plants, it causes severe yield losses. Furthermore, bacterial leaf blight could be transmitted through seeds. Bacterial spores are dispersed through wind and rainwater. Apart from that, use of a balanced amount of fertilizers, keeping the fields clean, management of proper drainage, and allowing fields to completely dry for some time are other preventive methods of bacterial leaf blight.

Bacterial leaf streak is another bacterial disease prevailing in rice and wheat. The causative agent for bacterial leaf streak in rice is Xanthomonas oryzae pv. oryzicola. The bacteria cause small thinner linear lesions on the leaves. The lesions on leaves will become brownish due to drying. Regionally, this disease can be seen in tropical and subtropical regions of Asia, Africa, South America, and Australia. Also, high temperature and high humidity often

favor this disease. Like bacterial leaf blight, Bacterial leaf streak also can be controlled effectively by planting resistant varieties. Furthermore, hot water treatment of seeds, keeping the field clean, use of balanced amounts of fertilizers and ensuring good drainage in the fields are some of the other methods that can prevent this disease.

Henceforth, Asibi, Chai, and Coulter (2019) conducted a study about rice blast. Rice blast is a serious fungal disease of rice (Oryza sativa L.) that is threatening global food security. It has been extensively studied due to the importance of rice production and consumption, and because of its vast distribution and destructiveness across the world. Rice blast, caused by Pyricularia oryzae Cavara 1892 (A), can infect above ground tissues of rice plants at any growth stage and cause total crop failure. The pathogen produces lesions on leaves (leaf blast), leaf collars (collar blast), culms, culm nodes, panicle neck nodes (neck rot), and panicles (panicles blast), which vary in color and shape depending on varietal resistance, environmental conditions, and age. Understanding how rice blast is affected by environmental conditions at the cellular and genetic level will provide critical insight into incidence of the disease in future

23

climates for effective decision-making and management. Integrative strategies are required for successful control of rice blast, including chemical use, biocontrol, selection of advanced breeding lines and cultivars with resistance genes, investigating genetic diversity and virulence of the pathogen, forecasting and mapping distribution of the disease and pathogen races, and examining the role of wild rice and weeds in rice blast epidemics. These tactics should be rotations, avoiding broadcast planting and double cropping, water management, and removal of yield-limiting factors for rice production. Such an approach, where chemical use is based on crop injury and estimated yield and economic losses, is fundamental for the sustainable control of rice blast to improve rice production for global food security.

A crucial challenge to rice production is rice blast, caused by the fungus Pyricularia oryzae Cavara 1892. Rice blast is one of the most serious and recurrent difficulties affecting lowland and upland rice production around the world. Rice blast is responsible for yield losses of about 10% to 30% annually. In favorable conditions, this disease can devastate entire rice plants within 15% to 20% and cause yield losses of up to 100%. Rice blast has become more difficult to

control because of the pathogen's ability to survive and multiply in harsh environmental conditions and easily spread to new fields. Varietal resistances have declined due to the appearance of new and more virulent strains of the pathogen, making management and control more challenging. Additionally, fungicides and plant breeding have failed to provide long-lasting control of rice blast because they are too static to deal with the dynamic interactions between the pathogen and rice, which are influenced by the surrounding environment. Understanding the effects of the rice blast pathogen, the efficacy of rice defense mechanisms, and the impact of climate change on rice blast are crucial for enhancing global food security.

Similarly, Reddy et al., (1979), conducted field experiments to evaluate the relationship between severity of bacterial leaf blight (BLB) on rice and on yields. Epidemics of BLB with different disease progress curves were encouraged by manipulating the initial dates and frequencies of subsequent inoculations, the use of bactericide, and selection of rice cultivars thought to differ in disease reaction to the causal organism, Xanthomonas oryzae. Late season epidemics (i.e., initiated after flowering) had no

measurable effect on grain yield or yield components. Epidemics that began before panicle initiation significantly reduced grain yield, panicle fertility, and kernel weight. The discovery of a significant, linear relationship between BLB severity at the soft dough stage of plant growth and grain yield allowed the construction of a critical point model to predict BLB-associated crop losses.

CHAPTER 3 RESEARCH DESIGN AND METHODOLOGY

Description of The Proposed Study

The proposed study developed a mobile application to capture images and classify rice leaf diseases. Additionally, it provided diagnosis and suggested treatments.

The application's disease detection function works when Internet connectivity is available as the model resides in the cloud. In the event that connectivity is unavailable, the user may capture images of leaves with possible diseases (those with spots) from the farm and have the detection once connectivity is available.

Also, this application generates a report wherein all the data/save results are presented on various charts. The location feature enables the farmer to add and locate the rice field area. The collected data is analyzed to provide a summary of events.

It could be used in the day-to-day monitoring of rice plants to help farmers and agriculture enthusiasts easily identify the rice leaf disease.

Method and Proposed Enhancements

Sources of information

Related Literature. The researchers reviewed the past and related studies to gather enough information in conducting the proposed study. Electronically published papers, articles, and journals of different researchers were used as a reference to be serving as a guide for the study.

Repositories. The researchers gathered the data from open-source codes such as W3Schools, Node.js were references to JavaScript and HyperText Markup Language. Open-sourced applications that exhibited relevance in the proposed study were reviewed and utilized to develop the application.

Experiences and Observations. Experiences and observations were the first sources of information in the initial stage of the study, with the help of Mr. Francisco A. Gonzaga, III, Head of LGU Cabatuan Department of Agriculture, the researchers were able to develop the idea that led to the development of the proposed study.

Evaluation. The software was evaluated based on the ISO/IEC 25010 Standard Software Evaluation Tool. This form is used to evaluate the qualities of a software or system. A form that determines the usability and effectiveness of the

application based on the ISO Standard Evaluation tool was answered by ten (10) randomly selected people who tried the application and further evaluated by the jurors as required.

Proposed Enhancements

    Real-Time capture. Capturing real-time images of rice leaf diseases yields a higher accuracy rate. Using Tracking.js, the Bounding Box was created and used to eliminate the area of the rice leaf that could be analyzed.

    Upload image. Unlike real-time capture, the proposed solution enables uploading images of rice leaves. Because the application requires an internet connection, the user can capture the images of rice leaf diseases beforehand and upload them if the connection is available.

    Location. For users with multiple areas of farmlands, identifying where the image was captured could be used, specifying that a disease or disease is present in different areas.

    Records of results. This proposed feature is to track the results and images of rice leaf diseases that have been captured or uploaded. The results show how many diseases were found and what dominant disease is present in the field.

Diagnosis. This feature is to help the user decide on the possible and most effective way to treat the disease that was found. Everything that the user would want to know about the disease, such as how it affects the crop, its occurrence, and how to identify and manage it, could be seen. Machine Learning was used upon creating the model and uploaded to the Google server. A Convolutional Neural network enables the application to differentiate the image taken or uploaded into different diseases.

Components and Design

Software Architecture

Figure 1 shows the software components of the application. The researchers used Hypertext Markup Language (HTML) for the user interface in this study. The live capture of images used tracking.js to extract the image frame. The software does not need to use tracking.js on the add photo because it is already an image. After capturing and adding photos, the process underwent rice leaf detection, which detected the rice leaf on the image. Next is the image classification, where the researchers used tensorflow.js to

integrate the algorithm in classifying what type of disease was found in the image captured/uploaded.

Meanwhile, base64.js was used to call the image uploaded or tracked in the disease classification result. HTML was used for use interface and design. In the database section, the researchers used phpMyAdmin to connect the different data and information inputted by the user, like images and profile information. In every detection and image classification process, there is an interaction in which the researchers train rice leaf datasets to acquire accurate results. Apache Cordova was used to wrap HTML/JavaScript apps into a container that can access the device function in different platforms.
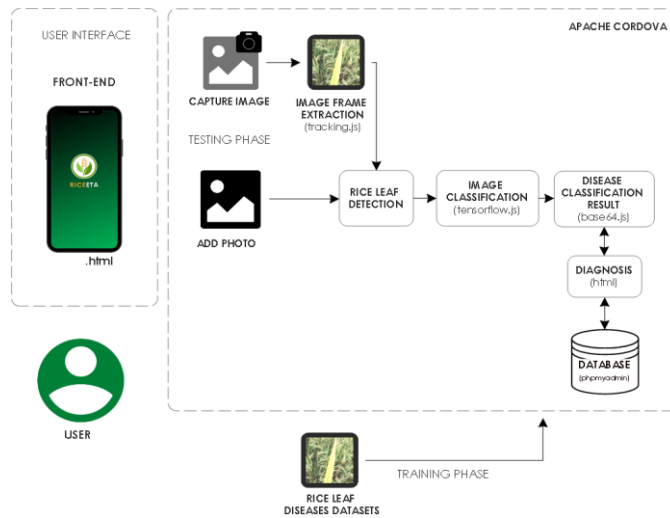
Figure 1. Software Architecture

System Architecture

Figure 2 shows how the information and data travel in the system. The rice detection model was generated on the cloud, wherein the different disease was classified. Models were uploaded to the application and accepted image or video inputs. Every time the application accepts inputs, the image will be compared to which classification of the disease it will fall under.
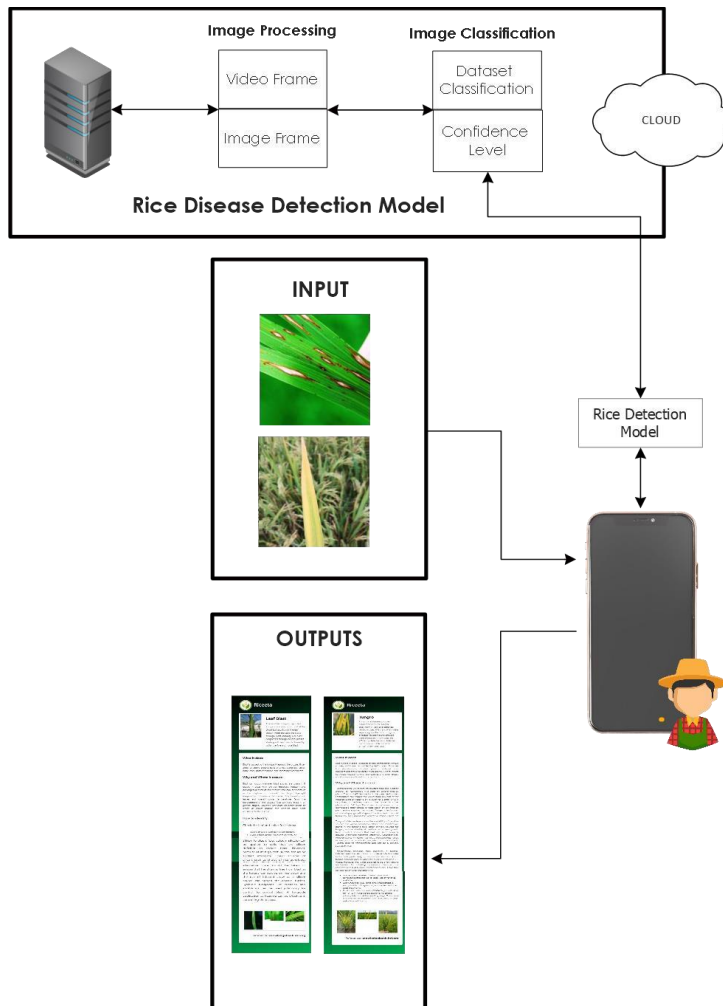
Figure 2. System Architecture

33

Database Design

Figure 3 shows the table: Users, Area, Images, and Diseases. On the users table, elements such as user_id, Username, password, first_name, middle_name, last_name, address, and role are found. The primary key is the user_id and the unique id is the username.

On the areas table, elements are the area_id, area_name, area_location, and user_id. The primary key is the area_id, and the foreign key is the user_id.

On the disease table, elements are disease_id, disease_name, disease_description, and the user_id. The primary key is the disease_id, and the foreign key is the user_id.

Meanwhile, in the images table, elements like image_id, image_, image_datetime, image_value, user_id, area_id, and disease_id. The primary key is the image_id, and the foreign keys are the user_id, area_id, and the disease_id.

One user can generate many images, areas, and diseases. In an area, there could be many images and diseases. In image, it could be classified into one disease. Many images could be found by the user.
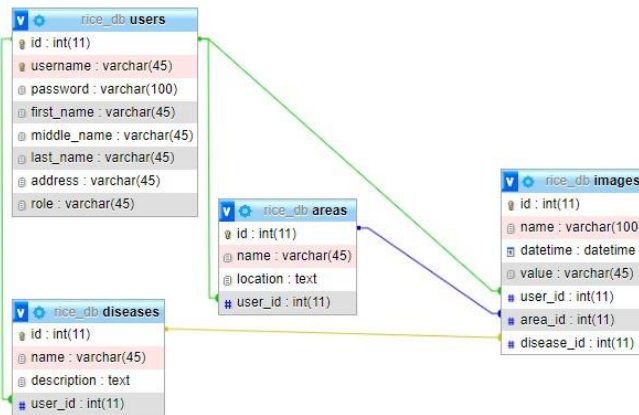
Figure 3. Database Design

Sign Up and Login Process

Figure 4 shows the researchers' proposed application, namely Riceeta: On-Device Inference for Rice Leaf Disease Diagnosis and Treatment.

To access the application, the user must have an account; if not, they must create an account so that the credentials entered are added to the database. After logging in, the user will be redirected to the homepage, where various events occur. If the user wishes to leave the application, he presses the back button on navigation.

35
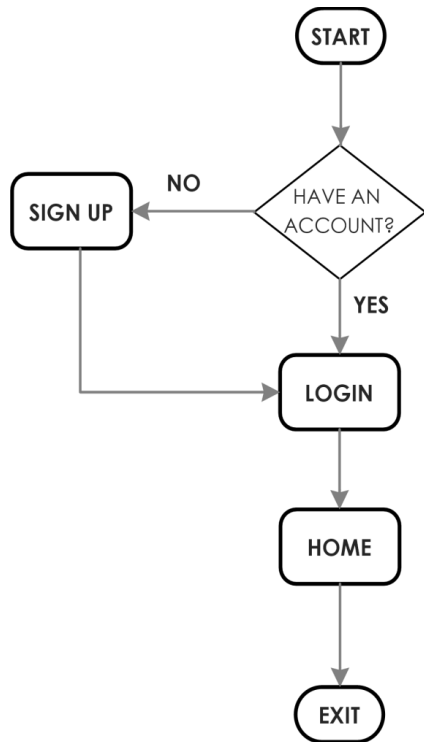
Figure 4. Sign Up and Login Process

Overall Process Design of Application

You can select from Camera, Add Photo, Report and Profile buttons on the Homepage screen. On-Camera, you will capture an image while you can attach an image in the Add Photo. After doing so, either of the two processes will analyze the leaf, and it will give results and diagnosis. The data of the classified image will be saved in the database.

On reports, there are charts where all the classified images are tallied.

On Locations, you can add our state farm name and where it is located.

In Profile, enables the user to view and update the information the user inputted.
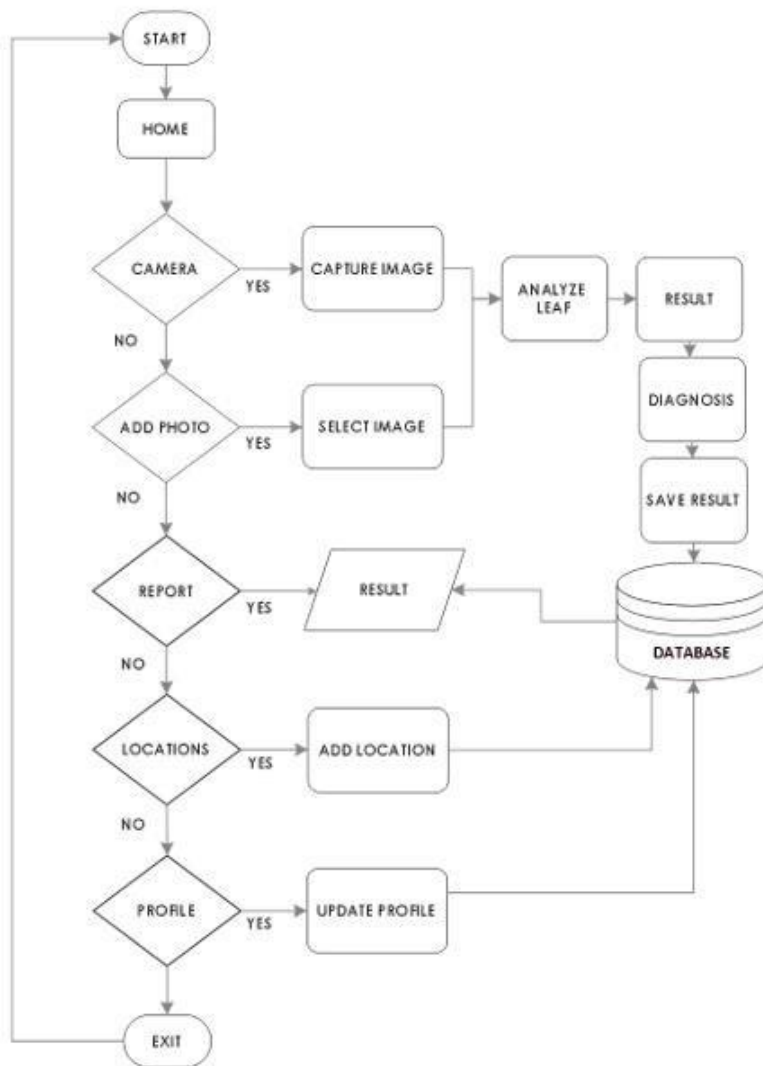
Figure 5. Overall Process Design of Application
Detection and Classification Process

38

Figure 6 shows where the detection and classification process starts from the feeding image to the application; after that, object detection will follow. Image result is collected and is used in the disease classification process wherein the image will be tested in three different diseases of rice leaf, namely Bacterial Leaf Blight, Tungro, and Leaf Blast.

Next in the process, the system will count the percentage by threshold control to know how much value the image will fall on the three different leaf diseases.

Lastly, the process will project the output in which it will give a diagnosis of the type of disease and information on how to check and control it.
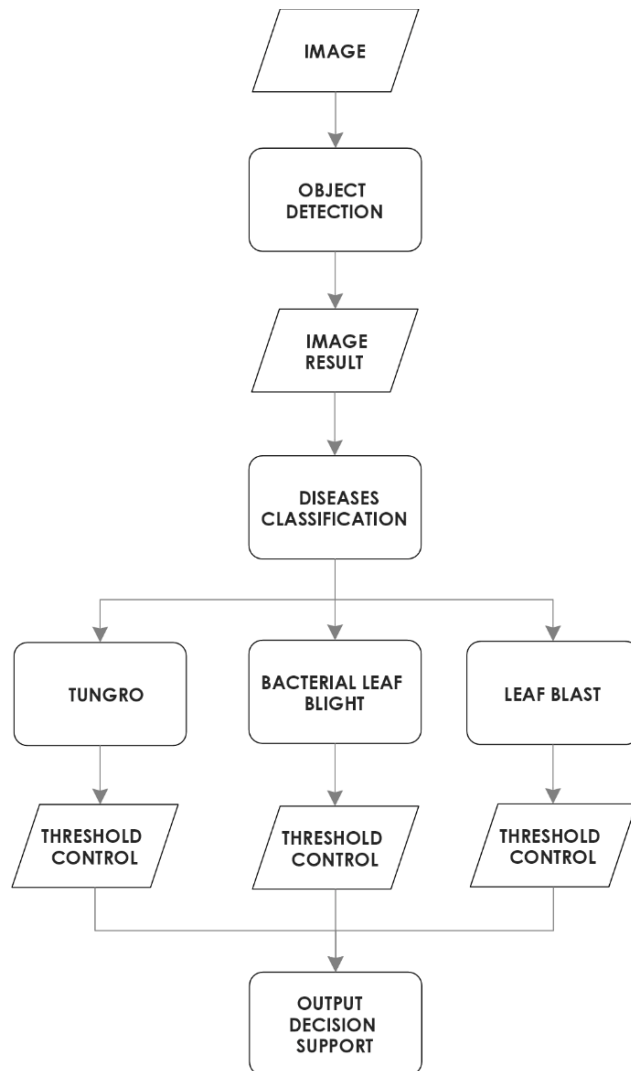
Figure 6. Detection and Classification Process

Model Building Process Design

Figure 7 shows the process design of the model building. It shows the data of different rice leaf diseases, namely the Bacterial Leaf Blight, Tungro, and Leaf Blast. After that, those data or images are trained by grouping each disease. Next is the validation of images. The researchers check if the training model could detect and differentiate rice leaf diseases using training sets of images.

Export follows this process. Once the training sets in the validation stage are accurate, the researchers export the trained model by downloading the file in TensorFlow file format.
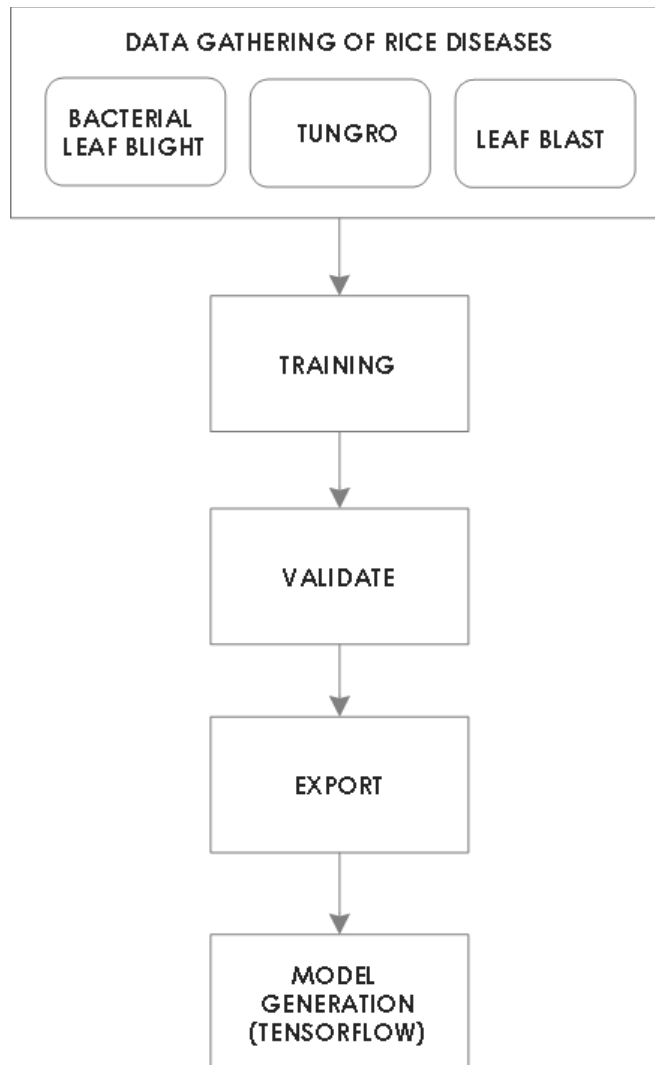
Figure 7. Model Building Process Design

System Development Life Cycle

Figure 8 illustrates the agile development methodology, which is the approach for software development in this study. The researchers developed the system incrementally where certain revisions of prototypes and extensions of the system's features were made to meet the requirements of the research adviser and the expert.

The planning phase is the first stage of the study, where the researchers identifies the problem and the possible solution they could provide. After the approval of the proposal, researchers learn and find ways to develop the system, what type of programming languages should be used, the layout, the designs of the system, and the possible algorithms to be used.

The Data Gathering phase is where the researchers gather information from the internet for the related studies. They also conducted an interview that provided insights into the development and improvement of the system.

In the System Requirements phase, the researchers studied and analyzed the information gathered, the possible specifications of the system, and further analysis and understanding of the problem were made.

Commented [u4]: Please be consistent with your tenses.

43

The Design phase is being studied. Navigation, icons and background, and colors must be pleasing to the eye. The application must be user-friendly and easily accessible and finalize the design and functionalities of the system using the java programming language.

In the Programming phase, the researchers studied and laid out the GUI of the system. Wrap the Html and JavaScript into a container using Cordova to access the device functions of several functions.

Debugging phase is where the prototype of the application is deployed. The researchers debugged the codes to eliminate and fix the errors that might conflict with the application. The researchers test the application on mobile phones to see if it works and see if it can detect the rice leaf disease pointed at the camera.

In the Deployment phase, the prototype of the said application was deployed to an android smartphone. The bugs and errors were fixed during the deployment process, and the application was ready to use once the iterations were done.
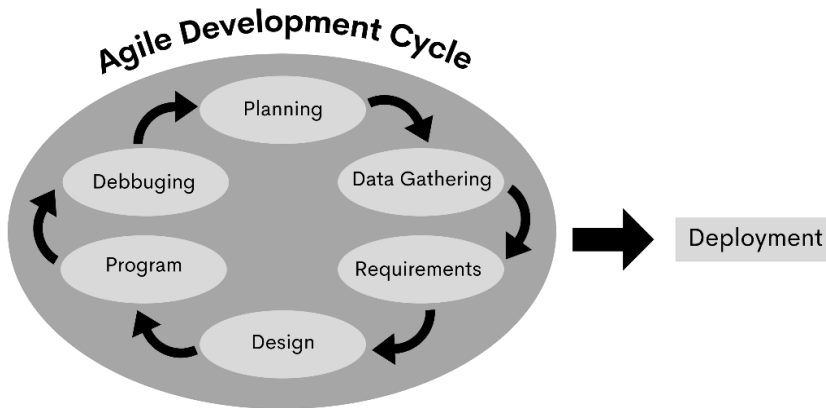
Figure 8. Agile Development Cycle

CHAPTER 4 RESULTS AND DISCUSSION

Implementation

The system was proposed as a mobile application to capture and upload images of rice leaf diseases. Researchers gathered different data of the top 3 most common rice leaf, namely the Bacterial Leaf Blast, Bacterial Leaf Blight, and Tungro. Images were classified and trained using the Teachable Machine, and the model was created. Moreover, the model was validated and tested by uploading stored images and real-time capture on the field. The initial testing showed a 100% accuracy rate of identifying disease. The developed application can work on Windows and Android platforms.

Technical Specification

Specifications are needed before the implementation of the proposed system. These requirements were identified to guarantee the effectiveness of the system. The following are software, hardware, and user specifications.

Software Specification

The system runs on Android Operating System and uses Apache Cordova for building mobile apps with HTML5, CSS, and JavaScript. A teachable Machine was used to create the model integrated into the application. Models made with Teachable Machine are TensorFlow.js models (a mathematical program that predicts which class you are showing it). TensorFlow's highlevel APIs are for defining and training Convolutional Neural Networks. The model is published to google cloud. IDE alternative used as Programmer's Notepad to develop an application that captures and interprets rice leaf diseases. The system's minimum requirements include an Android 7.1-9, while the recommended requirement of the system includes an Android v-11 and the use of MySQL as a database.

Software	Description
Operating System	Android
Development Tool	Apache Cordova
Integrated Development Environment (IDE)	Programmer's Notepad
Software Development Kit	Android SDK Kit
Other Tools Used	Teachable Machine

Table 1. Software Specification

Hardware Specification

 The minimum specifications of the system are smartphones
with a rear camera of an 8MP and a minimum system requirement
of 2.0 GHz Octa-core, 4GB RAM and up, and 6GB of storage and
up.

Hardware	Description
Android	11
Processor	2.0 GHz Octa-core
RAM	6.00 GB
Camera	16+8+2 Mp AI Triple rear Camera
Display	1080 x 2340 pixels, 19.5:9 ratio (~404 ppi density)

Table 2. Hardware Specification

    The generation of the model starts by collecting the

data on diseases that could be seen only in the Philippines.

Images were classified into certain groups by identifying

what type of disease they belonged. Images were uploaded on

Teachable Machine and trained each type to generate the model.

The researchers then tried to test if the trained model could

classify an image from one rice leaf disease to another. The

researchers found that the generated model can accurately

identify rice leaf diseases 100%. However, one image of

bacterial leaf blight was identical to leaf blast. Images

were tested by uploading rice leaves on white background and

on-field (realtime) images. Due to the researchers'
availability of applications, the Programmer's Notepad was
used in encoding the application. Apache Cordova and Android
SDK Kit were used so that Html and java codes were utilized
as a mobile application to integrate and make the codes work
on mobile.

User Specification

The system was developed for old and new farmers and
agriculturists. The user must have a farm wherein he/she could
find different types of diseases, namely Bacterial Leaf
Blight, Tungro, and Leaf Blast. It also required that the
user follow the given instructions on executing the steps.

System Inputs and Outputs

The Riceeta application requires the user to have
an Android operating system for the system to run and a camera
to enable rice leaf disease recognition.

The input of the user's camera recognized the rice leaf
having a disease present on the different parts of the leaf.

The researchers' system then analyzes the live image captured or the stored image on the gallery, detects the disease, and then classifies what type of disease it is. Also, there is an additional system functionality where the user can view the images analyzed. The system outputs the disease's kind, as well as its diagnosis and outcomes.

Figure 9 is the Login page. On this page, the user will log in using the email and password.
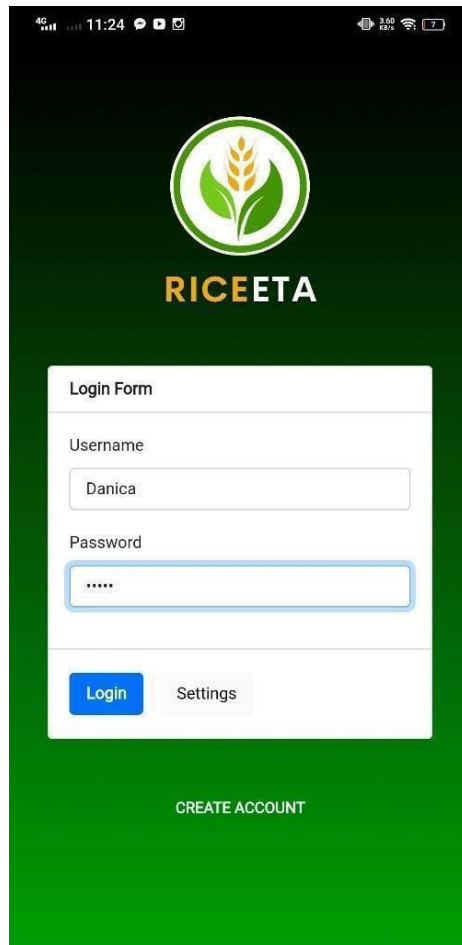
Figure 9. Login page

Figure 10 shows the create account pane where the user must enter his/her information such as username, password, first name, middle name, last name, and address.

Figure 10. Create Account

Figure 11 shows the setting menu where the user needs to enter the model URL and the server URL. This will enable the application to connect the models to the server. Connecting the application to the localhost will enable the images to be stored and can have access to information inside the database.
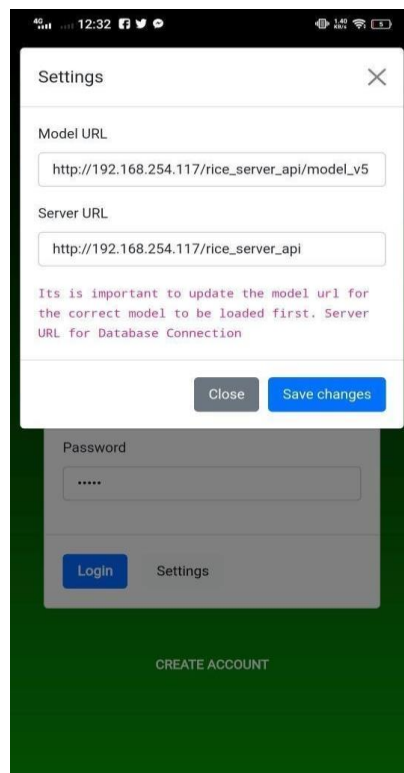
Figure 11. Settings

Figure 12 shows the homepage of the application. It includes features, namely Camera, Add Photo, Report, and Profile.
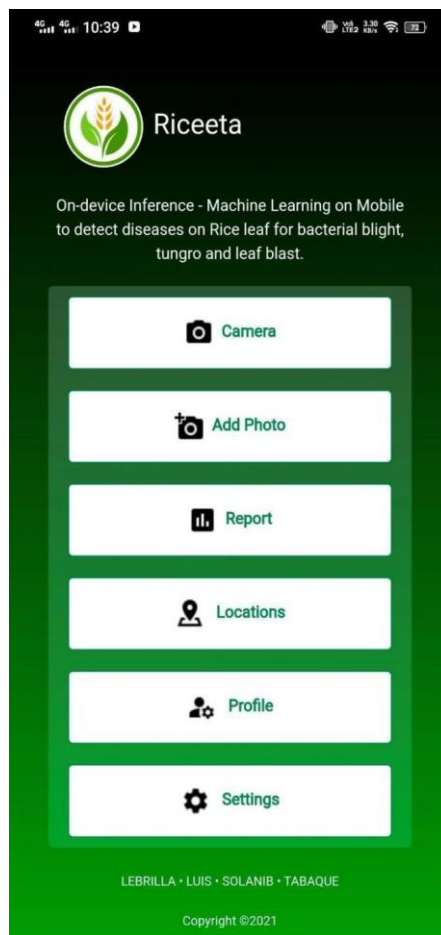

Figure 12. Homepage

54

Figure 13 shows how the camera is used. The user must make the camera clear of vision to identify the rice leaf. Before clicking the analyzed leaf, select the farm first, and the bounding box is visible on the screen so that the image will be captured and identified. After analyzing the leaf, the disease percentage is projected. This part showcases the value of the rice leaf disease in each disease. A short description or diagnosis will prompt. As seen in the picture, there is a linked text that, if clicked, will proceed to the diagnosis and results page.

Identical to the camera, the difference between the added photo in figure 14 is that the user can select from the stored images in the gallery.

The save results button enables the user to save the image captured in the results area.

Figure 13. Camera

Figure 14. Add Photo

The diagnosis and results page shows the type of disease detected on rice leaf diseases. It includes a description of the disease, what it does, how to identify it, and how to manage it.

Figure 15. Diagnosis and Results Page

In this section, reports of how many diseases were put into charts. Also, one can see the list of images that the user saved, the classified disease, the date and time photo/image saved, and the confidence level. On the last part of the page, a summary of the total count of diseases is projected in percentage form. The equation to solve this percent is (Number of Disease/Total Disease Count) * 100.
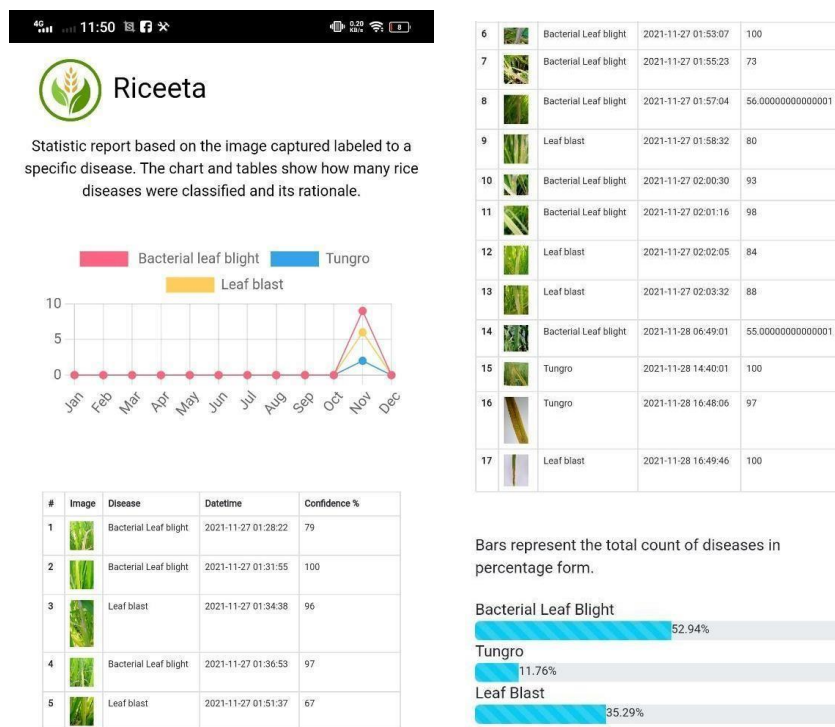
Figure 16. Reports

On the location page, the user adds the farm number and name of the area his farm is located, which could be used in identifying the location before saving the results.

Figure 17. Location Page

The profile page enables the user to change his/her credentials. The user can change his First Name, Middle Name, Last Name, Address, and Password.

Figure 18. Profile Page

Results Interpretation and Analysis

61

Model Generation

In generating the disease classification model, the more images loaded as a training dataset and the clearer the images for training, the higher the accuracy of the results. Some images could be labeled as similar to other classes, which confused the application, thus, lessening the accuracy rate.

Model Validation

The researchers loaded 89 bacterial leaf blight images, 86 Tungro images, and 89 leaf blast images for the model validation. The model had 100% accurately identified all test data for the Leaf Blast and Tungro diseases. However, one bacterial leaf blight image that was trained was mismatched to be a leaf blast. It showed that one image has similar characteristics to the leaf blast class.

Figure 19 shows the percentage and accuracy of each sample dataset loaded on the model. Here, the accuracy per class is calculated using the test samples.

The confusion matrix summarizes how accurate the model's predictions are. This matrix was used to determine which classes in the model get confused about. The class represents

the samples, while the prediction represents the class that
the model guesses after learning those samples.

Accuracy per class ⓘ

CLASS	ACCURACY	# SAMPLES
bacterial_leaf_bli...	0.99	89
tungro	1.00	86
leaf_blast	1.00	89

Confusion Matrix ⓘ

	bacterial_le...	tungro	leaf_blast
bacterial_le...	88	0	1
tungro	0	86	0
leaf_blast	0	0	89

scaleCou 89 — 0
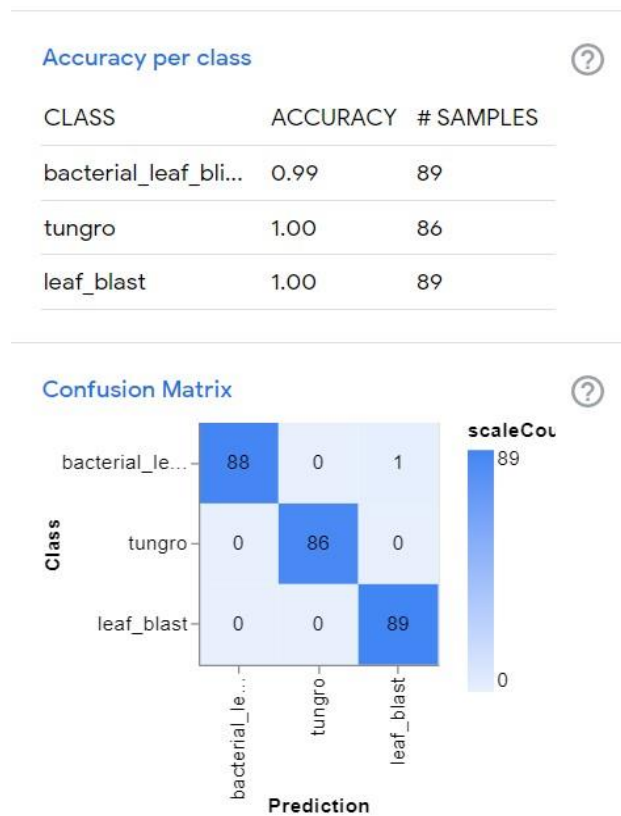
Class / Prediction

Figure 19. Accuracy Per Class and Confusion Matrix

Accuracy is the percentage of classification that a model gets right during training. It means that the model's prediction is 0.86 out of 1.

Loss is a measure for evaluating how well a model has learned to predict the correct classifications for a given set of examples. This model's prediction is perfect because the test loss is closer to zero.
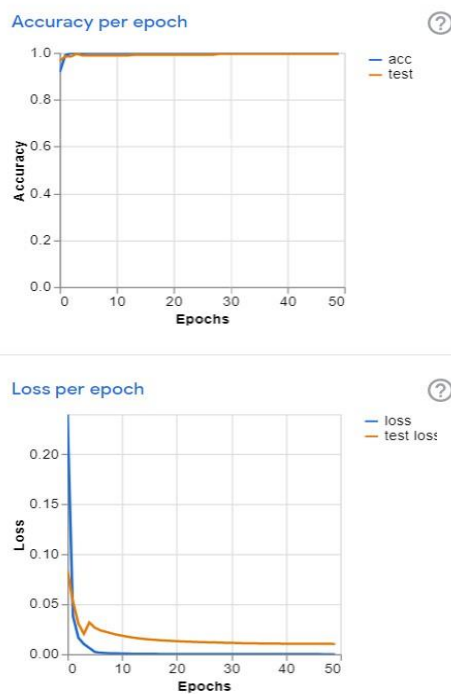
Figure 20. Accuracy and Loss Per Epoch

System Evaluation Results

The proposed system was presented to ten farmers to determine its quality. The College of Information and Communication Technology (CICT) constructed a software evaluation form covering ten criteria.

The criteria for evaluation involved the following: reliability, usability, efficiency, understandability, navigation, usefulness, appropriateness, correctness, User interface, and integrity. The following showed the result of the proposed system evaluated by the farmers. The mean of each area was computed by adding all the gathered scores in that area, then divided by the number of respondents.

mean = sum of area scores / the number of respondents

CRITERIA	SCORE	MEAN	%
1. Software adequately meets its objectives.	44	4.4	0.88
2. Software is usable without reference manual or user help.	45	4.5	0.9
3. Bug free: program runs properly.	38	3.8	0.76

4. Are buttons varied, obvious, and easy to use?	42	4.2	0.84
5. Is the software easy to learn the first time you use it?	48	4.8	0.96
6. Is it easy to navigate through the software?	42	4.2	0.84
7. How appealing is the user interface design.	50	5	1
8. The system uses a terminology that is understandable by the user.	48	4.8	0.96
9. How useful is the application?	50	5	1
10. How practical is it to use the software?	48	4.8	0.96
OVERALL EVALUATION	**456**	**5**	**0.91**

Table 3. QUESTIONNAIRES SCORE

Rating Scale	Description
5	Excellent
4	Good
3	Satisfactory
2	Below Average
1	Poor

Based on the scores found in the table, it can be observed that:

1.Functional Suitability. Functional Completeness, the system covers all the specified tasks. It shows that 88% of the participants agreed that software adequately meets its objectives. The majority that agreed with this question concluded that the researchers had met their objectives.

2.Usability. Operability, the system has attributes that make it easy to operate and control. It shows that 90% of the participants agreed that software is usable without reference manual or user help, while 10% still need assistance. It concludes that most users can easily use the software without any help or manual.

3.Reliability. Fault Tolerance: The system operates as intended despite hardware or software faults. 76% of the participants found the app bug-free, and the software runs smoothly. It assumes that some participants found the app to not run smoothly.

4.Usability. Appropriateness Recognizability allows users to recognize if it is appropriate for their needs. 84% of the participants have found the buttons varied, prominent,

This means 96% of the participants can easily understand the terminology used in the software.

9.Usability. Appropriateness Recognizability. The system allows users to recognize if it is appropriate for their needs. It signifies that 100% (All Participants) of the participants found the software helpful.

10. Usability. Appropriateness Recognizability. The system allows users to recognize if it is appropriate for their needs. It means 96% of the participants found the software practical to use.

Based on the information reflected in this table, it is evident that the participants found the software's user interface to be appealing. It reveals that the participants prefer to have a simple, clean interface that is easy to understand.

It can also be assumed that the participants found the software helpful. All participants rated the software as "excellent" regarding its usefulness. Question 3 scored the lowest, which means the participants found the software to be slow in loading time.

CHAPTER 5 SUMMARY, CONCLUSIONS AND RECOMMENDATIONS

Summary of the Proposed Study Design and Implementation

The proposed system, Riceeta: On-device inference for rice leaf disease diagnosis and treatment is a mobile application that aims to assist the farmers in diagnosing the diseases of rice leaf. The proposed mobile application can capture real-time rice leaves and determine the disease of the rice using a Convolutional Neural Network algorithm. With the help of an object detection algorithm, it allows the application to classify the disease on the rice leaf by extracting the disease features. The rice leaf image will serve as an input. The program will display the condition and its diagnosis.

Summary of Findings

The study's finding on "*RICEETA: On-Device Interference for Rice Leaf Disease Diagnosis and Treatment*" helps farmers decide how to deal with the diseases on their rice crops.

Results of the evaluation of the application showed a 100% accuracy and that it correctly identified the type of disease found.

The system was tested by farmers and the Department of Agriculture of Cabatuan. They found the system to be helpful in terms of functionality. The Department of Agriculture of Cabatuan posited that the application is beneficial nowadays for farmers to be in trend using smartphones to solve relevant problems in their field of expertise. Also, the study's objectives were concerning the top diseases of rice in the Philippines.

To develop and test the system, the researchers needed assistance from the Department of Agriculture to ensure the system reached its objectives. The system serves as a practical approach to determining the disease in a rice leaf.

Conclusions

Rice plant diseases is one of the major factors that affects rice crop yield. Rice just like any other plants are affected by bacterial and fungal diseases. Most of these diseases manifest as spots on the leaves. The prompt detection

of these diseases is vital to minimize substantial damage to the rice crops. This study presented a smartphone application that collects and recognizes image of rice leaf diseases and provides diagnosis and solutions. The application is able to recognize and detect rice leaf diseases such as bacterial leaf blight, tungro, and leaf blast.

Lastly, the accuracy of the result is directly affected by the quality of the dataset used. A good dataset should possess qualities of Clarity of the image, Proper size of the image, and the correct angle of the image. The validation result showed that the better image dataset yields better accuracy.

Recommendations

Riceeta: On-device Inference for Rice Leaf Disease Diagnosis and Treatment is effective and therefore benefits the agricultural sector. Nevertheless, the problem does not stop after successfully determining the disease and providing treatment. The disease spreads throughout the entire cultivation, which can cause devastating damage to the rice crops.

The researchers wish to make some recommendations that, if taken into consideration, might bring some positive changes and upgrade from the current approach. The recommendations are as follows:

1. To develop a position mapping system that can map the possible infected areas from the last position where the user/farmer found the rice leaf disease. Weekly risk alerts are also a profound solution associated with vulnerable rice areas reported within the application.

2. Weather forecasting to help the farmers identify when is the best time to plant rice crops while minimizing the damage.

3. Adds more rice diseases to expand the study's scope and help more farmers identify and properly diagnose rice leaf diseases. In addition to diseases, there is a need to find and gather datasets and generate an updated model.

4. The system should be available offline and online. Also, the data should be stored on the phone storage and can manage multiple users simultaneously.

References

Commented [u7]: Please check the format use by your college.

Agoot,L. (2020). Philippine News Agency:" PH logs highest   rice production rate at 19.44M metric tons: DA.".
https://www.pna.gov.ph/articles/1125473

Arida, I. A. (2010). Problems in rice farming: a Filipino farmers' perspective. Philippine Journal of Crop Science. 34,   42.
https://agris.fao.org/agrissearch/search.do?recordID=P
H2009001446#:~:text=The%20result
s%20revealed%20that%20common,and%20diseases%20and%20
irrigation%20system.

Azim, M. A., Islam, M. K., Rahman, M. M., & Jahan, F.
2021). An effective feature extraction method for
rice leaf disease classification. TELKOMNIKA
(Telecommunication Computing Electronics and Control), 19,
463-470.10.12928/telkomnika.v19i2.16488. Retrieved from:
https://www.researchgate.net/publication/349409277_An
effective_feature_extraction_method_for_rice_leaf_
disease_classification

Bakar, M. A., Abdullah, A. H., Rahim, N. A., Yazid,

H., Misman, S. N., & Masnan, M. J. (2018). Rice leaf blast disease detection using multi-level colour image thresholding. Journal of Telecommunication, Electronic and Computer Engineering (JTEC), 10(1-15), 1-6. https://www.researchgate.net/publication/326800397_Rice_Leaf_Blast_Disease_Detection_Using_Multi-Level_Color_Image_Thresholding

Bashir, K., Rehman, M., & Bari, M. (2019). *Detection and Classification of Rice Diseases: An Automated Approach Using Textural Features*. January 2019, 38(1), 239–250. https://doi.org/10.22581/muet1982.1901.20

Chen, C. J., Huang, Y. Y., Li, Y. S., Chang, C. Y., & Huang, Y. M. (2020). *An AIoT Based Smart Agricultural System for Pests Detection.* IEEE Access, 8, 180750–180761. https://doi.org/10.1109/access.2020.3024891

Chen, W.-C., Chiou, T.-Y., Delgado, A. L., & Liao, C.-S. (2019, June 23). The control of Rice Blast Disease by the novel biofungicide formulations. MDPI. Retrieved April 11, 2022, from https://www.mdpi.com/2071-1050/11/12/3449/htm.

Daniya, T. & Vigneshwari, Dr. S. (2019). A Review on Machine Learning Techniques for Rice Plant Disease Detection in Agricultural Research. 28. 49-62. https://www.researchgate.net/publication/340087925_A_Review _on_Machine_Learning_Techniques_for_Rice_Plant_Disease_Dete ction_in_Agricultural_Research

IRRI. Rice Today (2020). Philippines records its highest rice production in 2020. Retrieved April 11, 2022, from https://ricetoday.irri.org/philippines-records-itshighest- rice-production-in-2020/

J. P. Shah, H. B. Prajapati and V. K. Dabhi, "A survey on detection and classification of rice plant diseases," 2016 IEEE International Conference on Current Trends in Advanced Computing (ICCTAC), 2016, pp. 1-8, doi: 10.1109/ICCTAC.2016.7567333. https://ieeexplore.ieee.org/document/7567333

Kulkarni, M. P. (2020). *Rice Plant Disease Detection.*

International Journal for Research in Applied Science and Engineering Technology. 8(6), 237–241. https://doi.org/10.22214/ijraset.2020.6033

Li, D., Wang, R., Xie, C., Liu, L., Zhang, J., Li, R., Wang, F., Zhou, M., & Liu, W. (2020). *A Recognition Method for Rice Plant Diseases and Pests Video Detection Based on Deep Convolutional Neural Network*. Sensors, 20(3), 578. https://doi.org/10.3390/s20030578

Li, H., Deng, L., Yang, C., Liu, J., & Gu, Z. (2021). *Enhanced YOLO v3 Tiny Network for Real-Time Ship Detection From Visual Image*. IEEE Access, 9, 16692–16706. https://doi.org/10.1109/access.2021.3053956

M. N. Abu Bakar1 et al.(nd). *Rice Leaf Blast Disease Detection YOLO v3: An Incremental Improvements In MultiLevel Colour Image Thresholding.* Journal of Telecommunication, Electronic and Computer Engineering. e-ISSN: 2289-8131 Vol.

Mahlein, A. K. (2016a). *Plant Disease Detection by*

Imaging Sensors – Parallels and Specific Demands for Precision Agriculture and Plant Phenotyping. Plant Disease, 100(2), 241–251. https://doi.org/10.1094/pdis-03-15-0340-fe

Meshram, V., Patil, K., Meshram, V., Hanchate, D., & Ramkteke, S. D. (2021, October 2). Machine learning in agriculture domain: A state-of-art survey. Artificial Intelligence in the Life Sciences. Retrieved April 11, 2022, from https://www.sciencedirect.com/science/article/pii/S2667318521000106.

Mirandilla, J. R. & Paringit, E. C. (2019). DETECTION AND DIFFERENTIATION OF SELECTED RICE DISEASES USING HYPERSPECTRAL DATA. https://a-a-rs.org/proceeding/ACRS2019/TuP-6.pdf.

Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). *Using Deep Learning for Image-Based Plant Disease Detection.* Frontiers in Plant Science, 7. https://doi.org/10.3389/fpls.2016.01419

Mutka, A. M., & Bart, R. S. (2015). *Image-based phenotyping of plant disease symptoms*. Frontiers in Plant Science, 5. https://doi.org/10.3389/fpls.2014.00734 Liu, J., & Wang, X. (2021). *Plant diseases and pests detection based on deep learning: a review*. Plant Methods, 17(1). https://doi.org/10.1186/s13007-021-00722-9

Palay production in the Philippines - psa.gov.ph. (n.d.). Retrieved April 8, 2022, from https://psa.gov.ph/sites/default/files/Palay%20Production%20in%20the%20Philippines%2C%202017-2019.pdf

Pest, diseases, weeds of Rice & their management ... - farmer. (n.d.). Retrieved April 11, 2022, from http://farmer.gov.in/imagedefault/pestanddiseasescrops/pestanddiseasesofrice.pdf.

Prakruti, V.(2019). *Detection of diseases and pests on images captured in uncontrolled conditions from tea plantations*. Proc. SPIE 11008, Autonomous Air and Ground Sensing Systems for Agricultural Optimization and Phenotyping IV, 1100808 doi: 10.1117/12.2518868.

Rathore, N.P.S., & Prasad, Dr.L. (2020). Automatic Rice Plant Disease Recognition and Identification using Convolutional Neural Network. Vol 7, 2394-512. https://www.researchgate.net/publication/344688862_journal_ of_critical_reviews_automatic_rice_plant_disease_recognitio n_and_identification_using_convolutional_neural_network

Rice. Rice: Diseases and Pests, Description, Uses, Propagation. n.d.). Retrieved April 11, 2022, from https://plantvillage.psu.edu/topics/rice/infos

S. Archana, K., & Sahayadhas, A. (2018). *Automatic Rice Leaf Disease Segmentation Using Image Processing Techniques*. International Journal of Engineering & Technology, 7(3.27), 182. https://doi.org/10.14419/ijet.v7i3.27.17756

Sethy, Prabira, Baishalee Negi, and Nilamani Bhoi. (2017). *Detection of Healthy & Defected Diseased Leaf of Rice Crop using K-Means Clustering Technique.* International Journal of Computer Applications 157, no. 1: 0975-8887.

Sethy, Prabira & Negi, Baishalee & Barpanda, Nalini & Behera, Santi & Rath, Amiya. (2018). Measurement of Disease Severity of Rice Crop Using Machine Learning and Computational Intelligence. 10.1007/978-981-10-6698-6_1. https://www.researchgate.net/publication/322026075_Measurement_of_Disease_Severity_of_Rice_Crop_Using_Machine_Learning_and_Computational_Intelligence

Towards paddy rice smart farming: A review on Big Data, machine learning, and rice production tasks. IEEE Xplore. (n.d.). Retrieved April 11, 2022, from https://ieeexplore.ieee.org/document/9389541?source=authoralert.

Temniranrat, P., Kiratiratanapruk, K., Kitvimonrat, A., Sinthupinyo, W., & Patarapuwadol, S. (2021). *A system for automatic rice disease detection from rice paddy images serviced via a Chatbot*. Computers and Electronics in Agriculture, 185, 106156. https://doi.org/10.1016/j.compag.2021.106156 Vibhute, Anup & S.K. Bodhe,. (2012). Application of

Image Processing in Agriculture: A Survey. International
Journal   of   Computer   Applications.   52.   34-40.
10.5120/81761495.
https://www.researchgate.net/publication/253908767_Applicat
ion_of_Image_Processing_in_Agriculture_A_Survey

Appendices

Appendix A

Letter to the Adviser

West Visayas State University

College of Information and Communication Technology

La Paz, Iloilo City

March 2, 2021

DR. REGIN A. CABACAS
Assist. Professor IV
West Visayas State University
Luna St., La Paz,
Iloilo City 5000 Iloilo, Philippines
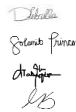
Dear Dr. Cabacas,

The undersigned are BS Information Technology Research 1/Thesis 1 students of CICT, this university. Our thesis/capstone project title is *"RICEETA: On-Device Interference for Rice Leaf Disease Diagnosis and Treatment"*.

Knowing of your expertise in research and on the subject matter, we would like to request you to be our **ADVISER**.
We are positively hoping for your acceptance. Kindly check the corresponding box and affix your signature in the space provided. Thank you very much.

Respectfully yours,

1.   Lebrilla, Danica Marie

2.   Solanib, Prince Deo

3.   Tabaque, Aris Ernst

4.   Luis, Lennox
  PS:

Advisers, are task to work with the students in providing direction and assistance as needed in their thesis/capstone project. They shall meet with the students weekly or as needed to provide direction, check on progress and assist in resolving problems until such a time that the students passed their defenses and submit their final requirements, as well as, preparing their evaluations and grades.

Action Taken:

⬤ I Accept.

◯ Sorry. I don't accept.

REGIN A. CABACAS

Signature over printed name of the Adviser

Appendix B

Request Letter for Interview

West Visayas State University
La Paz, Iloilo City College of Information and
Communication Technology

November 22, 2021
Mr. FRANCISCO A. GONZAGA III
Head of Department of Agriculture
Municipality of Cabatuan
Cabatuan, Iloilo City, 5031

Mr. Gonzaga:

Good Day!

We are the fourth-year students at West Visayas State
University – College of Information and Communications
Technology taking up Bachelor of Science in Information
Technology (BSIT). As part of our Undergraduate Thesis
requirement, we are conducting a study entitled "RICEETA:
OnDevice Interference for Rice Leaf Disease Diagnosis and
Treatment." The study aims to help new and old farmers
identify different rice leaf diseases by creating an
application to classify the image captured.
In connection with this, we would appreciate getting your
insights concerning our study. We believe that your
contribution is immeasurable and will greatly benefit us to
become globally competitive lifelong learners. We are looking
forward to your positive response.

Truly Yours,

DANICA MARIE A. LEBRILLA
LENNOX G. LUIS
PRINCE DEO S. SOLANIB
ARIS ERNST TABAQUE

Appendix C

Letter of Request to the Editor

West Visayas State University College of Information and
Communication Technology La Paz, Iloilo City

April 27, 2022

BONNA SOBREPEÑA PALMA, PH.D
Faculty
College of Education
West Visayas State University
Luna St, La Paz, Iloilo City, 5000 Iloilo

Dear Dr. Palma,

The undersigned are BS Information Technology fourth year
students at College of Information and Communications
Technology in this University. Our thesis/capstone project
title is "RICEETA: On-device Interference for Rice leaf
Diagnosis and Treatment".

Knowing of your expertise in research and on the subject
matter, we would like to request you to be our **THESIS
GRAMMARIAN.** We are positively hoping for your acceptance.

Thank you very much.

Respectfully Yours,

DANICA MARIE A. LEBRILLA
LENNOX G. LUIS
PRINCE DEO S. SOLANIB
ARIS ERNST TABAQUE

Appendix D

Gantt Chart

#	Name	Duration	Start	Finish
1	Title Ideation	5 d	01/02/2021	05/02/2021
2	Title Proposal	1 d	08/02/2021	08/02/2021
3	DA Consultation	1 d	04/03/2021	04/03/2021
4	Data Gathering	168 d	22/03/2021	10/11/2021
5	Adviser Consultation	1 d	04/03/2021	04/03/2021
6	System Analysis	93 d	02/04/2021	10/08/2021
7	Selection of Software Planning	8 d	20/09/2021	29/09/2021
8	Prototyping	25 d	01/09/2021	05/10/2021
9	Coding and Programming	164 d	12/04/2021	25/11/2021
10	Group Meeting	1 d	25/11/2021	25/11/2021
11	System Testing Debuging	10 d	15/11/2021	26/11/2021
12	Farmers Evaluation	5 d	22/11/2021	26/11/2021
13	DA Evaluation	1 d	29/11/2021	29/11/2021
14	Field Testing	1 d	29/11/2021	29/11/2021
15	Documention	174 d	02/04/2021	01/12/2021

Appendix E

Data Dictionary

Rice_db

Area

Column	Type	Null	Default	Comments
id *(Primary)*	int(11)	No		
name	varchar(45)	Yes	*NULL*	
location	text	Yes	*NULL*	
user_id	int(11)	No		

Index

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	id	2	A	No	
fk_areas_users1_idx	BTREE	No	No	user_id	2	A	No	

Diseases

Column	Type	Null	Default	Comments
id *(Primary)*	int(11)	No		
name	varchar(45)	Yes	*NULL*	
description	text	Yes	*NULL*	
user_id	int(11)	No		

Index

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	id	3	A	No	
fk_diseases_users1_idx	BTREE	No	No	user_id	3	A	No	

Images

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	id	12	A	No	
fk_images_users1_idx	BTREE	No	No	user_id	2	A	No	
fk_images_areas1_idx	BTREE	No	No	area_id	2	A	No	
fk_images_diseases1_idx	BTREE	No	No	disease_id	6	A	No	

Index

Column	Type	Null	Default	Comments
id *(Primary)*	int(11)	No		
name	varchar(100)	Yes	NULL	
datetime	datetime	Yes	NULL	
value	varchar(45)	Yes	NULL	
user_id	int(11)	No		
area_id	int(11)	No		
disease_id	int(11)	No		

Users

Column	Type	Null	Default	Comments
id *(Primary)*	int(11)	No		
username	varchar(45)	Yes	NULL	
password	varchar(100)	Yes	NULL	
first_name	varchar(45)	Yes	NULL	
middle_name	varchar(45)	Yes	NULL	
last_name	varchar(45)	Yes	NULL	
address	varchar(45)	Yes	NULL	
role	varchar(45)	Yes	NULL	

Index

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	id	3	A	No	
username_UNIQUE	BTREE	Yes	No	username	3	A	Yes	

Appendix F

Sample Program Codes

Recognize Add photo Images

```html
<html>
    <head>
        <meta charset="utf-8">
        <!--
        Customize this policy to fit your own app's needs. For more
guidance, see:
            https://github.com/apache/cordova-
pluginwhitelist/blob/master/README.md#content-security-policy
        Some notes:
* gap: is required only on iOS (when using UIWebView) and is needed
  for JS->native communication
* https://ssl.gstatic.com is required only on Android and is needed
  for TalkBack to function properly
* Disables use of inline scripts in order to mitigate risk of XSS
  vulnerabilities. To change this:
* Enable inline JS: add 'unsafe-inline' to default-src
        -->
        <meta name="format-detection" content="telephone=no">
        <meta name="msapplication-tap-highlight" content="no">
<meta name="viewport" content="initial-scale=1, width=devicewidth,
viewport-fit=cover">

        <meta name="color-scheme" content="light dark">
```

```
<link rel="stylesheet" href="css/bootstrap.min.css">

<link rel="stylesheet" href="css/animate.css">

<style>

        html, body, .ion-app, .ion-content {

        height: 100%;

    padding: 10px;

background-size: cover;


        background: #000000;  /* fallback for old browsers
*/          background: -webkit-linear-gradient(to top, #0f9b0f,
#000000);  /* Chrome 10-25, Safari 5.1-6 */

        background: linear-gradient(to top, #0f9b0f,
#000000); /* W3C, IE 10+/ Edge, Firefox 16+, Chrome 26+, Opera 12+,
Safari 7+ */

        background-attachment: fixed;

        }

        h1 {

            padding-left: 10px;

        }

        #inputVideo{

    width:100%;

  visibility:hidden

        }

    </style>

  <title>Riceeta</title>
```

```
    </head>

    <body  class="animate__animated   animate__fadeIn">


       <div class="container ">
               <h1 style="color:white; font-size: 30px;" ><img
style="width:60px"  src="img/riceeta.png"/> Riceeta</h1>
               <p style="color:white; font-size: 10px; text-align:
center;" class="fs-6">Upload a photo of rice leaf that you classified
having a disease.</p>
               <div class="w3-center topp-padding w3-padding-
bottom">

                   <center><img id="selected-image" class="w3-
round adjust-image" src="img/default.jpg" width="80%" alt="Image for
analysis"></center>

               </div>
               <div class="w3-center" id="progress-holder">
                   <div class="progress">
                    <div id="progress-bar" class="progress-bar
progress-bar-striped progress-bar-animated" role="progressbar"
style="width: 0%"></div>
                   </div>
                   <p>Model is loading...</p></div>
               </div>


               <form id="addPhotoForm" action="" type="post"
```

```
enctype="multipart/form-data" class="form p-2">

                    <div class="form-group">

                        <label

for="formGroupExampleInput"style="color:white;"><strong>Image</strong><
/label>

                            <input name="file" type="file"
class="form-control" id="fileImage" placeholder="" required>

                    </div>


                    <div class="form-group">

                        <label

for="formGroupExampleInput"style="color:white;"><strong>Select
Farm</strong></label>

                            <select name="area_id" id="areaForm"
class="form-control" required>

                             </select>

                    </div>

                    <br>

                    <div class="form-group">

                        <label

for="formGroupExampleInput"style="color:white;"><strong>Disease</strong
></label>

                            <input name="disease_id" type="text"
class="form-control" id="diseaseInput" placeholder="" required>

                    </div>
```

```
                    <div class="form-group">

                        <label for="formGroupExampleInput"

style="color:white;"><strong>Value</strong></label>

                            <input name="value" type="text"

class="form-control" id="valueInput" placeholder="" required>

                    </div>

                    <br>

                    <button class="btn btn-primary"

id="analyzeLeaf" type="button" ><strong> Analyze Leaf

</strong></button>

                        <button class="btn btn-secondary" type="submit"

id="save_results"style="margin:10px;"><strong> Save Results

</strong></button>

                </form>

                <div class="w3-center w3-border add-margin side-

margin w3-round w3-sand text-color space-letters bottom-margin"

style="color:white; font-size:

16px;">

     <p id="status" class='space-letters text-color mob-font1'></p>

<p style="color:white; font-size: 15px; padding-left: 10px;"

class="fs-6">Note: Click "Save Results" before viewing the

diagnosis.</p>

     <p style="color:white; font-size: 15px; padding-left: 10px;"

class="fs-6"><strong>Disease Percentage:</strong></strong>.</p>

     <ol class='w3-left-align text-color' id='prediction-list'></ol>

</div>
```

```
        <div class="p-3">

  <p style="color:white; font-size: 15px; text-align: left;"
id="diagnosis"></p>

  <p style="color:white; font-size: 15px; text-align: left;"
id="uploadMessage"></p>

              <img id="fromServer" class="img-fluid">


        </div>


        </div>
              <script src="cordova.js"></script>
              <script src="js/jquery-3.6.0.min.js"></script>
              <!-- Load TensorFlow.js -->
              <script src="js/tf.min.js"></script>
              <script src="js/bootstrap.min.js"></script>


              <script>
                      $(function(){
              server_ip =
window.localStorage.getItem("server_url");
                        model_ip =
window.localStorage.getItem("model_url");
                        _user =
JSON.parse(window.localStorage.getItem("user"));
$('#save_results').hide();
```

```
$('#diseaseInput,#valueInput').val("");


TARGET_CLASSES = {
  0: "bacterial_leaf_blight",
  1: "tungro",
  2: "leaf_blast"
};
DIAGNOSIS_CLASSES = {
    "bacterial_leaf_blight":"An early symptom of
bacterial leaf spot is small (less than 0.25 inch in diameter),
watersoaked leaf spots on the older leaves of the plant. These lesions
are typically bordered by leaf veins and angular in shape. Lesions
quickly turn black (a diagnostic characteristic of this disease).",
        "tungro":"Rice tungro disease is caused by
the combination of two viruses, which are transmitted by leafhoppers.
It causes leaf discoloration, stunted growth, reduced tiller numbers
and sterile or partly filled grains. ",
        "leaf_blast":"Blast is caused by the fungus
Magnaporthe oryzae. It can affect all above ground parts of a rice
plant: leaf, collar, node, neck, parts of panicle, and sometimes leaf
sheath."
        };
        function getBase64(file) {    return new
        Promise((resolve, reject) => { const
```

```
reader = new FileReader();

reader.readAsDataURL(file);

    reader.onload = () =>

resolve(reader.result);

        reader.onerror = error => reject(error);

  });

}


//load model

let model;

  (async function () {


    model = await

tf.loadLayersModel(model_ip+'/model.json',

        {

            onProgress: function(p){

                $('#progress-

bar').css({'width':p*100+'%'});

                }

        });


        $("#selected-image").attr("src",

"img/default.jpg")

        // Hide the model loading spinner

        $('#progress-holder').hide()
```

```
})();

$.ajax({

        type: "POST",

        url: server_ip+"/api.php",


data:"list_area=tru&user_id="+_user.id

            }).done(function(response) {

        areas = JSON.parse(response);



            areas.forEach(function(item){

        $('#areaForm').append("<option

value='"+item.id+"'>"+item.name+" "+item.location+"</option>");



            })



        });



    $('#addPhotoForm').submit(function(){




            _data =
$(this).serialize()+'&upload_image=true&user_id='+_user.id;

        alert('uploading please wait...');

        var file =
```

```
document.querySelector('#fileImage').files[0];

                        dataURL = getBase64(file).then(

                        data => {

                            setTimeout(function(){

                            //console.log(data);

                                $.ajax({

                        type: "POST",

                                        url:

server_ip+"/api.php",


        data:_data+'&imgBase64='+encodeURIComponent(data)

                                    }).done(function(response)

{

                                        console.log('saved: ' +

response);


        $('#fromServer').attr("src",server_ip+'/'+response);

                                        alert('saved: ' +

response);


        $('#uploadMessage').empty().append('saved: ' + response);


                            });

                        }, 1000)

                    }
```

101

```
                                return false;

                        })


                $("#fileImage").change(function () {

                        let reader = new FileReader();

                        reader.onload = function () {

                                let dataURL = reader.result;

                        $("#selected-image").attr("src",

dataURL);

                                $("#prediction-list").empty();

                        }

                                let file =

$(this).prop('files')[0];

                                reader.readAsDataURL(file);


                });


                $("#analyzeLeaf").click(async function () {

                let image = $('#selected-image').get(0);


                        // Pre-process the image

        let tensor = tf.browser.fromPixels(image)

                        .resizeNearestNeighbor([224,224]) //

change the image size here

                        .toFloat()
```

```
.div(tf.scalar(255.0))

.expandDims();


// Pass the tensor to the model and call
predict on it.
// Predict returns a tensor.
// data() loads the values of the output
tensor and returns
// a promise of a typed array when the
computation is complete.
// Notice the await and async keywords
are used together.
let predictions = await
model.predict(tensor).data();
let top5 = Array.from(predictions)
    .map(function (p, i) { // this is
Array.map

        return {
probability: p,
className:
TARGET_CLASSES[i] // we are selecting the value from the obj
            };
```

```
                            }).sort(function (a, b) {

                    return b.probability -

a.probability;


                            }).slice(0, 3);
                $("#prediction-list").empty();

                console.log(top5)
     $('#valueInput').val(top5[0].probability.toFixed(2)*100);

                $('#diseaseInput').val(top5[0].className);

                $('#diagnosis').empty().append('<h3

style="color:white; font-size: 18px; text-align: center; font-weight:

bold;" >Diagnosis for <a href="d_'+top5[0].className+'.html">

'+top5[0].className+'</a></h3><br/>').append(DIAGNOSIS_CLASSES[top5[0].

className]);

                $('#save_results').show();


                top5.forEach(function (p) {


                        $("#prediction-

list").append(`<li>${p.className}: ${p.probability.toFixed(2) *100}%

</li>`)

                        });
                    });

            });           </script>

    </body>
</html>
```

Recognize Captured live Images

```
<html>
    <head>
        <meta charset="utf-8">
        <!--
        Customize this policy to fit your own app's needs. For more
guidance, see:
            https://github.com/apache/cordova-
pluginwhitelist/blob/master/README.md#content-security-policy
        Some notes:
* gap: is required only on iOS (when using UIWebView) and is needed
  for JS->native communication
* https://ssl.gstatic.com is required only on Android and is needed
  for TalkBack to function properly
* Disables use of inline scripts in order to mitigate risk of XSS
  vulnerabilities. To change this:
* Enable inline JS: add 'unsafe-inline' to default-src
        -->
        <meta name="format-detection" content="telephone=no">
        <meta name="msapplication-tap-highlight" content="no">
<meta name="viewport" content="initial-scale=1, width=devicewidth,
viewport-fit=cover">
        <meta name="color-scheme" content="light dark">
        <link rel="stylesheet" href="css/bootstrap.min.css">
        <script
```

```
src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/1.0.0/p5.min.js"></sc
ript>
            <script
src="https://unpkg.com/ml5@latest/dist/ml5.min.js"></script>
            <style>
                html, body, .ion-app, .ion-content {
                 background-color: transparent;
background: url(img/bg2.jpg) no-repeat;
background-size: cover;                    height: 100%;


                }
                #inputVideo{
      width:480px;
height:360px;
                }
                #overlay{ display:none}
            </style>
        <title>Riceeta</title>
    </head>
    <body>
        <div class="app">
            <div style="position: relative;padding:10%"
id="videoHolder">
                    <video id="inputVideo" autoplay muted class="img-
fluid"></video>
```

106

```
            <canvas id="overlay" class="img-fluid"></canvas>


            <div id="results_here"></div>
            <div class="d-grid gap-2 mt-5">

                <button type="button" class="btn btn-light
btn-block btn-lg" data-bs-toggle="modal" data-bstarget="#exampleModal">

                    Show Rice Receta

                </button>

                </div>

            </div>

        <!-- Modal -->

        <div class="modal fade" id="exampleModal" tabindex="-1"
aria-labelledby="exampleModalLabel" aria-hidden="true">

            <div class="modal-dialog modal-fullscreen">

                <div class="modal-content">

                  <div class="modal-header">

                <h5 class="modal-title"
id="exampleModalLabel">Rice Receta</h5>

                    <button type="button" class="btn-close" data-
bs-dismiss="modal" aria-label="Close"></button>

                  </div>

                  <div class="modal-body">

                    <img src="img/brownspot.jpg" class="img-fluid"
alt="...">
```

```
                    <h1>Brown Spot</h1>

                    <p class="lead">Brown spot has been
historically largely ignored as one of the most common and most
damaging rice diseases. Brown spot has been historically largely
ignored as one of the most common and most damaging rice diseases. What
it does. Brown spot is a fungal disease that infects the coleoptile,
leaves, leaf sheath, panicle branches, glumes, and spikelets.. Its most
observable damage is the numerous big spots on the leaves which can
kill the whole leaf.</p>

                    </div>

                    <div class="modal-footer">

                        <button type="button" class=btn btn-secondary"
data-bs-dismiss="modal">Close</button>

                    </div>

                </div>

            </div>

        <script src="cordova.js"></script>

            <script defer src="js/jquery-3.6.0.min.js"></script>

            <script defer src="js/bootstrap.min.js"></script>

        <script src="js/index.js"></script>

    </body>

</html>
```

West Visayas State University
COLLEGE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY
La Paz, Iloilo City

Integrating of model

```php
?php include 'common.php';
//handle login request
if(isset($_POST['login'])){
$stmt = $db-
>prepare("SELECT * FROM
users WHERE
username=:username AND
password=:password LIMIT
1");
    $stmt->execute(['username' =>
$_POST['username'],'password'=>$_POST['password']]);
    $user = $stmt->fetch(PDO::FETCH_ASSOC);


    if($user){
        $stmt = $db->prepare("SELECT * FROM areas WHERE
user_id=".$user['id']);
        $stmt->execute();
        $farms = $stmt->fetchAll(PDO::FETCH_ASSOC);
        $user['farms'] = $farms;
    }


    echo json_encode($user);
}
//handle update profile if(isset($_POST['update_profile'])){
```

```php
        $stmt = $db->prepare("UPDATE `users` SET `password` = :password,
`first_name` = :first_name, `middle_name` = :middle_name, `last_name` =
:last_name, `address` = :address WHERE `users`.`id` = :user_id;");
if($stmt->execute($_POST['data'])){

            echo "1";

        }else{

        echo "0";

        }
}
//handle add area if(isset($_POST['add_area'])){


        $stmt = $db->prepare("INSERT INTO `areas` (`id`, `name`,
`location`, `user_id`) VALUES (NULL, :name, :location, :user_id)");
if($stmt->execute($_POST['data'])){


            $stmt1 = $db->prepare("SELECT * FROM areas WHERE
user_id=:user_id ");

            $stmt1->execute(['user_id' => $_POST['data']['user_id']]);

            $areas = $stmt1->fetchAll(PDO::FETCH_ASSOC);



            echo json_encode($areas);
```

```
        }else{

echo "0";

        }

}

//handle list area if(isset($_POST['list_area'])){


        $stmt1 = $db->prepare("SELECT * FROM areas WHERE user_id=:user_id
");

        $stmt1->execute(['user_id' => $_POST['user_id']]);

        $areas = $stmt1->fetchAll(PDO::FETCH_ASSOC);




        echo json_encode($areas);


}

//handle delete area if(isset($_POST['delete_area'])){


        $stmt = $db->prepare("DELETE FROM `areas` WHERE `areas`.`id` =
:id");        if($stmt->execute(['id' =>
$_POST['id']])){


            $stmt1 = $db->prepare("SELECT * FROM areas WHERE
user_id=:user_id ");
```

111

```php
            $stmt1->execute(['user_id' =>
$_POST['user_id']]);          $areas = $stmt1-
>fetchAll(PDO::FETCH_ASSOC);         echo json_encode($areas);
      }else{
echo "0";
      }
}


//handle create_account profile
if(isset($_POST['create_account'])){ try
{
          $stmt = $db->prepare("INSERT INTO `users` (`id`,
`username`, `password`, `first_name`, `middle_name`, `last_name`,
`address`, `role`) VALUES (NULL, :username, :password, :first_name,
:middle_name, :last_name, :address, 'user')");
          if($stmt->execute($_POST['data'])){
                echo "1";
          }else{
                echo "Failed on creating Account, Try again later.";
          }
      } catch (PDOException $e) {
echo $e->errorInfo[2];
      }
}
//handle report request if(isset($_POST['get_report'])){
```

```
//SELECT images.*,month(datetime) as month, diseases.name as

disease_name, diseases.description as disease_description, areas.name

as area_name, areas.location as area_location FROM `images` left join

diseases on images.disease_id = diseases.id left join areas on

images.area_id = areas.id


     //all user report

     //$stmt = $db->prepare("SELECT images.*,month(images.datetime)

as month, diseases.name as disease_name, diseases.description as

disease_description, areas.name as area_name, areas.location as

area_location FROM `images` left join diseases on images.disease_id =

diseases.id left join areas on images.area_id = areas.id");

     //$stmt->execute();


     $stmt = $db->prepare("SELECT images.*,month(images.datetime) as

month, diseases.name as disease_name, diseases.description as

disease_description, areas.name as area_name, areas.location as

area_location FROM `images` left join diseases on images.disease_id =

diseases.id left join areas on images.area_id = areas.id WHERE

images.user_id=:user_id");

     $stmt->execute(['user_id' => $_POST['user_id']]);

$images = $stmt->fetchAll(PDO::FETCH_ASSOC); echo '<table

class="table table-sm table-bordered" style="fontsize:.5em;">

                    <thead>

                    <tr>
```

113

```php
            <th scope="col">#</th>

            <th scope="col">Image</th>

            <th scope="col">Disease</th>

            <th scope="col">Datetime</th>

            <th scope="col">Confidence %</th>

        </tr>

    </thead>

    <tbody>';

    $counter = 1;

    foreach($images as $image){

        $img_url =
str_replace('api.php','','http://'.$_SERVER['SERVER_NAME'].''.$_SERVER[
'REQUEST_URI'].''.$image['name']);

        echo '

            <tr>

          <th scope="row">'.$counter.'</th>

          <td> <image src="'.$img_url.'"
style="width:20px;"/></td>

            <td>'.$image['disease_name'].'</td>

            <td>'.$image['datetime'].'</td>

            <td>'.$image['value'].'</td>

        </tr>

        ';


        $counter++;
```

114

```
                }


echo '                </tbody>
            </table>


            ';
            $_disease = [
            '1'=>[0,0,0,0,0,0,0,0,0,0,0,0],
        '2'=>[0,0,0,0,0,0,0,0,0,0,0,0],
    '3'=>[0,0,0,0,0,0,0,0,0,0,0,0],
            ];
            $_disease_count = ['1'=>0,'2'=>0,'3'=>0];


            $_total_count = 0;
    foreach($images as $image){


            $_disease[$image['disease_id']][$image['month']-1] += 1;


            $_disease_count[$image['disease_id']]++;
            $_total_count++;
            }
                echo "<p class='fs-6'>Bars represent the total count
    of diseases in percentage form.</p>";
```

```
echo 'Leaf Blight <div class="progress">

<div class="progress-bar progress-bar-striped

progress-bar-animated bg-info" role="progressbar" style="width:

'.(($_disease_count[1]*100) / $_total_count).'%"></div>

'.number_format(($_disease_count[1]*100) /

$_total_count,2).'%</div>';

echo 'Tungro <div class="progress">

<div class="progress-bar progress-bar-striped

progress-bar-animated bg-info" role="progressbar" style="width:

'.(($_disease_count[2]*100) / $_total_count).'%"></div>

'.number_format(($_disease_count[2]*100) /

$_total_count,2).'%</div>';

echo 'Leaf Blast <div class="progress">

<div class="progress-bar progress-bar-striped

progress-bar-animated bg-info" role="progressbar" style="width:

'.(($_disease_count[3]*100) / $_total_count).'%"></div>

'.number_format(($_disease_count[3]*100) /

$_total_count,2).'%</div>';

echo '

<br/>

<table class="table table-sm table-bordered"

style="font-size:.8em">

<thead>

<tr>

<th scope="col">#</th>
```

```
                        <th scope="col">Disease</th>

                        <th scope="col">Total Count</th>

                    </tr>

                </thead>

                <tbody>

                    <tr>

                        <th scope="row">1</th>

                        <td>Leaf Blight </td>

                        <td>'.$_disease_count[1].'</td>

                    </tr>

                    <tr>

                        <th scope="row">2</th>

                        <td>Tungro</td>

                        <td>'.$_disease_count[2].'</td>

                    </tr>

                    <tr>

                        <th scope="row">3</th>

                        <td>Leaf Blast </td>

                        <td>'.$_disease_count[3].'</td>

                    </tr>

                </tbody>

            </table>

        ';

    echo

     "
```

```
    <script>          dataSets = {

    labels: ['Jan', 'Feb', 'Mar',

 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec'],

                        datasets: [{

                label: 'leaf blight',

                        data:
[".implode(',',$_disease[1])."],

                        backgroundColor: [

                            'rgba(255, 99, 132,
1)',

                        ],

                        borderColor: [

                            'rgba(255, 99, 132,
1)',

                        ],

                        borderWidth: 1

                    },


                    {

                        label: 'tungro',

                        data:
[".implode(',',$_disease[2])."],

                        backgroundColor: [

                            'rgba(54, 162, 235,
1)',
```

118

```
                              ],

                              borderColor: [

                                    'rgba(54, 162, 235,

1)',

                              ],

                              borderWidth: 1

                  },


            {

                              label: 'leaf blast',

                              data:

[".implode(',',$_disease[3])."],

                              backgroundColor: [

                                    'rgba(255, 206, 86,

1)',

                              ],

                              borderColor: [

                                    'rgba(255, 206, 86,

1)',

                              ],

                              borderWidth: 1

                  }

                  ]

            };

            const ctx =
```

```
document.getElementById('chart').getContext('2d');

                    const myChart = new Chart(ctx, {

                        type: 'line',

                    data: dataSets,

            options: {

            scales: {

                                y: {

                                    beginAtZero: true

                                }

                        }

                    }

                });

        </script>

";


}

//handle upload Image version 1 if(isset($_POST['upload_image'])){

        $image =  imagesaver($_POST['imgBase64']); //use full base64 data


        $_diseases = [

          "bacterial_leaf_blight"=>1,

          "brown_spot"=>2,

          "leaf_smut"=>3

          ];

        $data = [
```

120

```php
            'name'=>$image,

            'datetime'=>date('Y-m-d H:i:s'),

            'value'=>$_POST['value'],

            'user_id'=>$_POST['user_id'],

            'area_id'=>$_POST['area_id'],

            'disease_id'=>$_diseases[$_POST['disease_id']],
        ];

        $stmt = $db->prepare("

            INSERT INTO `images` (`id`, `name`, `datetime`, `value`,
`user_id`, `area_id`, `disease_id`) VALUES

            (NULL, :name, :datetime, :value, :user_id, :area_id,
:disease_id)

            ");

        $stmt->execute($data);

echo $image;

}

function imagesaver($image_data){


    list($type, $data) = explode(';', $image_data); // exploding data
for later checking and validating

    if (preg_match('/^data:image\/(\w+);base64,/', $image_data,
$type))
{

        $data = substr($data, strpos($data, ',') + 1);

$type = strtolower($type[1]); // jpg, png, gif
```

```php
        if (!in_array($type, [ 'jpg', 'jpeg', 'gif', 'png' ]))
{           throw new \Exception('invalid image type');
        }

        $data = base64_decode($data);
        if ($data === false) {              throw new
\Exception('base64_decode failed');
        }       } else {          throw new \Exception('did not match data
URI with image data');
    }


    $fullname = 'images/'.uniqid().'.'.$type;
     if(file_put_contents($fullname,
$data)){

        $result = $fullname;

    }else{

        $result =  "error";

    }
    /* it will return image name if image is saved successfully
or it will return error on failing to save image. */
return $result;
}
```

Appendix G

Software Quality Evaluation Form

West Visayas State University
La Paz, Iloilo City
College of Information and Communication Technology

Software Quality Evaluation Form
(For Department of Agriculture)

Name:_____   Age:_____

Evaluation Guide: 6 Excellent, 5 -Very Good, 4 – Good, 3 -Fair, 2 -Poor,1 -Very Poor

Rating Scale	Poor	Below average	Satisfac tory	Good	Very Good	Excellent
Evaluation Section	1	2	3	4	5	6
1. Functional Correctness. The system provides the correct results with the needed degree of precision.						
2. Time Behaviour. The system's response and processing times and throughput rates when performing its functions, meet requirements.						
3. Interoperability. The system can exchange information and use the information that has been exchanged.						
4. Appropriateness Recognizability. The system allows users to recognize if it is appropriate for their needs.						
5.Learnability. The system can be used by specified users to achieve specified goals of learning to use the application with effectiveness, efficiency, freedom from risk and satisfaction in a specified context of use.						
6 Operability. The system has attributes that make it easy to operate and control.						
7.User Interaction Aesthetics. The system's user interface enables pleasing and satisfying interaction for the user.						
8. Confidentiality. The system ensures that data are accessible only to those authorized to have access.						
9. Integrity. The system prevents unauthorized access to, or modification of, computer programs or data.						
10. Non-repudiation. The system can be proven to have taken place, so that the events or actions cannot be repudiated later.						

Evaluated by: _____   Juror No.:____Date: _____
Signature over Printed Name

*adopted from Questionnaire for ISO 25010: SOFTWARE QUALITY STANDARDS

Appendix H

Disclaimer

This software project and its corresponding documentation entitled "RICEETA: ON-DEVICE INFERENCE FOR RICE LEAF DISEASE DIAGNOSIS AND TREATMENT" is submitted to the College of Information and Communication Technology, West Visayas State University, in partial fulfillment of the requirements for the degree, Bachelor of Science in Information Technology. It is the product of our own work, except where indicated text.

We hereby grant the College of Information and Communication Technology permission to freely use, publish in local or international journal/conference, reproduce, or distribute publicly the paper and electronic copies of this software project and its corresponding documentation in whole or in part, provided that we are acknowledged.

DANICA MARIE A. LEBRILLA   LENNOX G. LUIS

PRINCE DEO S. SOLANIB   ARIS ERNST TABAQUE

June 2022