DIGITAL IMAGE PROCESSING: IMPLEMENTING YOLOV5 ALGORITHM IN PLANT CLASSIFICATION WEB APPLICATION THROUGH LEAF DETECTION

Billy Joe L. Andrade
Computer Engineering Department
College of Engineering and Technology
Quezon City, Philiipines
bjlandrade2019@plm.edu.ph

Theron Adrianne A. Bueno
Computer Engineering Department
College of Engineering and Technology
Manila, Philippines
taabueno2019@plm.edu.ph

Miguel T. Cayaban
Computer Engineering Department
College of Engineering and Technology
Cavite, Philippines
mtcayaban2019@plm.edu.ph

Mc Angelo R. Lacson
Computer Engineering Department
College of Engineering and Technology
Manila, Philippines
marlacson2019@plm.edu.ph

Marc Chester D. Masakayan
Computer Engineering Department
College of Engineering and Technology
Quezon Province, Philippines
mcdmasakayan2019@plm.edu.ph

Christian Dave M. Vila
Computer Engineering Department
College of Engineering and Technology
Mandaluyong City, Philippines
cdmvila2019@plm.edu.ph

***Abstract*— This paper is all about creating a model to implement digital image processing. The model produced is a web application entitled "IdentiPlant" which is developed using the YOLOv5 (You Only Look Once version 5) algorithm along with ReactJS, HTML, CSS, and JavaScript. The machine learning algorithm needs to be trained to produce accurate results. This is done by using datasets that are composed of a specific number of images for each type or class. The datasets are photos of leaves that belong to varied species of plants collected through manual image capturing.**

The web application, IdentiPlant, used the YOLOv5 algorithm to detect the leaves. The training data is first pre-processed (e.g., cropping images to a 1:1 ratio), then trained the model using the YOLOv5 algorithm. Static images and real-time video capture of leaves are used as inputs to process an output that provided the captured plant's information such as: what plant it is, its habitat, and its common uses and benefits. At the end of the study, the researchers concluded that using the said algorithm in object classification is highly accurate. The outcome presented is beneficial for the developers and will be beneficial to future researchers in line with the project in a way that they will have an idea of what YOLOv5 can do as an algorithm for object classification. The Web App can be used at this link: https://prodev-theron.github.io/dip-plantidentifier/

Keywords—object classification, image processing, plant identification, leaf recognition, YOLOv5

I. INTRODUCTION

1.1. *BACKGROUND OF THE STUDY*

Plants play an important role in nature. There are numerous plant species and types that are both known and unknown to humans [1]. There are thought to be 350,000 different plant species and some are widely used in the world of food, medicine, industrial products, and are extremely important for environmental protection [2, 3]. Plants also control the climate, and it can be a habitat for insects and other living organisms. The three main components of a plant are the root, stem, and leaves. Leaf structures play a very crucial role in determining the characteristics of a plant [4]. Several laboratory techniques have been used to study their identifications, and leaves are the primary way to recognize one [5]. The traditional plant identification method is difficult to use and not suited for widespread adoption and it was more complex and challenging to determine the leaf differences due to the broad morphological variations that had developed during the evolution. However, studies have recently shown how to distinguish between different leaf variations using genetic features, plant chemotaxonomy, and visual approaches [6]. Plant classification is the cornerstone of not only botany but also plant ecology, genetics, medicine, and life science. It plays a crucial part in the preservation, growth, and exploitation of plant resources. However, these categorization or identification techniques are exclusively accessible to researchers. Additionally, the accuracy of classification is impacted by the fact that these systems mostly rely on individuals' subjective assessments. As a result, it is challenging to satisfy the need for promptly identifying plants wherever they are.

The technology of target recognition based on image processing, such as face recognition, fingerprint recognition, etc., has already penetrated people's lives with the advent of digital image processing and pattern recognition. These methods offer sufficient theoretical support and technical setup for image-based plant recognition. As a result, research on plant recognition based on image processing has been quietly gaining popularity among scholars worldwide in recent years. This process is quick and does not rely on the subjectivity of the individual. It increases job productivity and plant recognition accuracy while successfully compensating for the shortcomings of artificial recognition methods. People can learn about the plant more quickly and thoroughly by using this quick and easy way.

1.2. *STATEMENT OF THE PROBLEM*

There are several algorithms to use for image classification such as Shadow algorithms, Minimum Mean Distance, K-Nearest Neighbor algorithms, Support Vector Machine, Decision Trees, K-means Clustering algorithms, and so on [7]. With these algorithms, developers can create image classification projects in various ways. However, choosing the suitable algorithm to use while considering numerous factors in the development such as the given timeframe, available resources, and so on, will be hard for most developers due to a lack of idea about these algorithms.

According to a study, surveys and experiments state that having more choices rarely makes consumers in a better position from the same source, another problem is that if we do not include plants in our discoveries and do not give sufficient emphasis on their life, our incomplete knowledge about them will be distorted [8].

To address the lack of ideas about these algorithms, tutorials and documentation are scattered online to educate the developers. However, learning these algorithms, not to mention the other algorithms that are not cited, one-by-one before making the most suitable choice will be time-consuming. Additionally, the development of mobile applications such as LeapSnap, Picture This, and PlantSnap have become the solution to the problem regarding the acquisition of knowledge at emphasis about plants' life.

Given the case, the researchers came up with the idea of designing an image classification project using one of the fastest algorithms, specifically YOLOv5 which is open source. In this way, the YOLOv5 algorithm's impact on the production of the output of the image classification project will be known by implementing it. By the end of this study, the readers will have a better understanding of YOLOv5 and will lessen the hurdle of choosing the best algorithm to use as this paper will make YOLOv5 one of their choices and lessen the options that they are lacking knowledge about. Regarding the development of applications, the researchers will support this solution by developing a web-based app which can be accessed in any device that can run browsers.

1.3. Objective of the Study

To respond to the problems cited above, the researchers composed objectives to serve as a guide in achieving the study's intent. Below are the general and specific objectives of the study.

1.3.1 General Objective

To develop a web application that can classify and provide information about plants through leaf image processing.

1.3.2 Specific Objective

1. To apply an algorithm that analyzes the physical characteristic of a leaf to have an accurate classification of plants.
2. To deliver information about the plant identified using leaf image processing (e.g., type of plant, plant kingdom, habitat, and way of planting).

1.4. Significance of the Study

The goal that is set to conduct this study is to primarily acquire a first-hand experience in creating a project to classify objects, in this time plants, with the use of YOLOv5. The expected output for this research will be beneficial and significant to the ff:

The Researchers who conducted this study. At the end of this study, the researchers will become more knowledgeable and gain a better understanding of the capabilities of YOLOv5 algorithm in conducting Digital Image Processing, particularly compound-scaled object detection.

The Developers conducting an image classification project. The research provides developers the opportunity to assess if they can integrate YOLOv5 in their own projects. The algorithm is not necessarily only limited to plant identification, but it can also be used for other object identification projects.

The Future Researchers who aim to conduct related studies. The outcome will provide helpful information that can guide the future researchers that have the same goal of conducting a study that is similar or related to the topic of plant classification with the use of image processing of leaves.

1.5. Scope and Limitations

1.5.1 Scope of the Study

This study will focus on implementing and/or developing an algorithm that will adhere to the objectives stated above. To do that, the researchers will need to use the algorithm present in YOLOv5 and train the algorithm to recognize 5 species of plant class such as Oregano, Mango, Croton, Golden Pothos, and Camias. Collaborative tools were also used during the development phase such as GitHub and Google Colaboratory.

1.5.2 Limitation of the Study

Limitations will also be set in place. Due to time constraints, the researchers will limit the training cycle of the algorithm to 100 images per plant class. The researchers will also not touch the subject of image quality, with factors such as weather, time of when it was captured and its brightness, being fed to the algorithm. Also, the condition of the leaf, such as being damaged, being moist, and obstructed by another object, will not be involved in the study. And the format of the images that were used in the study are all in JPEG/JPG. Other formats such as PNG or PDF, are not incorporated in the study.

II. Review of Related Literature

2.1. Plants

Plants have played a significant role in human history since the first studies of living processes, and they were at the heart of scientific inquiry in the late nineteenth and early twentieth centuries [9]. Plants are a crucial research tool used to better understand fundamental life processes and to address issues in agriculture, human health, and the environment [10]. 98% of the oxygen in the atmosphere is created by plants through photosynthesis and everything we eat originates from plants, either directly or indirectly [11]. Prescription medications made from plants or plant compounds account for 25% of all pharmaceuticals [11]. Humans study plants to better understand processes that are essential to both the survival of our species and the health of the planet, given the significance of plants in every part of our life. Beyond their obvious significance, plants have contributed significantly to numerous biological discoveries that have aided in our understanding of some of life's most intriguing mysteries. Research with plants has strongly influenced the development of biology and has contributed to many important scientific advances [9]. Through preventing erosive processes and water contamination, as well as by assisting in the reduction of air pollution, plants play a crucial role in preserving a healthy environment. From enclosed quarters to huge wilderness areas, they enhance the conditions for human activity everywhere.

2.1.1. *Identification of Plants*

Being able to assess a plant is a crucial part of the process in the field of natural products. According to Nakano [12], People around the world have learned to identify plants and their properties. This traditional knowledge, according to Nakano, allowed generations to thrive in diverse environments for thousands of years. However, according to Nakano, there are more than 300,000 species in the plant kingdom. Hence, confusion may inevitably ensue.

According to Marceau [13], he presented a case study where a retired man was almost apprehended due to suspected presence of marijuana in his garden only to find out that it was an okra, a comestible fruit. To an untrained eye, according to Marceau [13], okra and marijuana can be mistaken for each other as they have identical features that can hinder people from being able to distinguish each plant from each other.

2.1.2. *Information from Leaves*

Leaves are the most discernible and easily recognizable part of the natural world [14] and the reason for the earth's green decorations. It is also a major organ involved in light perception and conversion of solar energy into organic carbon [15]. Most leaves are constructed of three major parts: the petiole (the stalk that connects the leaf blade to the stem or meristem of the plant), the base (the region of the leaf blade that connects to the petiole), and the blade [14]. Even though there are only three parts, the leaves are unique for each plant. This is thanks to the enormous amount of diversity that exists in leaf shapes, textures, and even colors [14], to adapt to different natural habitats and maximize life strategies and propagation [15].

Leaves differ in various aspects. It can be large or small, symmetrical, or asymmetrical, have jagged or smooth edges, and appear glossy or rough. In some cases, it lacks a part like sessile leaves which lacks petiole. However, even though leaves vary uniquely, it is broadly classified into two types. The simple leaves are composed of a single blade (e.g., rose, mango, and banana), and the compound leaves are composed of multiple blades, which are called leaflets (e.g., acacia, coconut, and palm tree) [14]. Leaves don’t only vary physically, the arrangement of leaves on a stem also differs. It can be opposite, such as when two leaves grow directly across from each other, or alternating, when leaves alternate sides, one after the other [15]. Leaves don’t look the same all the time for each plant. Throughout their lives, leaves expand in size and may also change color [14]. It may be because of the season, or the mechanical properties of particular cells and tissues change for adaptation to the environment [15]. For example, in the fall, leaf cells accumulate more light-absorbing pigments called carotenoids. It is a yellow, orange, and red pigment that is present in plants but hidden by the chlorophyll of the plants [16].

Most of us know that leaves are multicellular organs [15] and the structure is perfectly made for photosynthesis [14] and for providing oxygen for other living organisms. Leaves share general characteristics, but some have taken on special abilities. Climbing plants like cucumbers have leaves that coil and form tendrils. This helps the plant attach to support as it climbs [14]. Other leaves impersonate petals to attract pollinators like bees to the flowers they are supporting or reduced themselves called cacti spines to protect the plant from predators with their sharp and pointed shapes [14]. These only states that leaves are hard to identify or classify.

2.1.3. *Recognizing Leaves to Identify Plants*

Workflows in plant ecology research heavily rely on the identification of plants [17]. Plant Identification demands extensive knowledge and understanding of complex terminologies, even professionals require significant amount of time in the field of their subject mastery. One of the most vital components of a plant is its leaf. They are essential for respiration, energy absorption, and defense. The classification of a plant's variety and family can be done with the use of its leaves. Plant species can be recognized by its leaf, flower, skin, fruit, and seed, etc., and relatively speaking, using leaf to recognize plant species is very simple and convenient, and many leaf based plant species recognition methods have been proposed [18]. Leaves possess useful characteristics for plant species identification—the shape of them and their floral organs are extremely important, with the leaves considered an especially useful identity of a plant [19]. Different leaf kinds can be distinguished by their form, shape, and other attributes. Identifying leaves requires looking at a variety of features [20]. The many leaf varieties each serve a specific function and have adaptations that enable the plant to flourish in its natural habitat. It is crucial to examine the entire leaf, paying particular attention to the Base, Margins, Tip, Veins, if present, Petiole, and the Midrib. Leaves are generally arranged either alternately along the stem or opposite each other [21].

2.2. *Digital Image Processing*

2.2.1. *Web-Based Application*

A web-based application is essentially a computer program that is kept on a remote server and makes use of the internet. The utilization of web browsers and various technologies (such as Flash, Silverlight, JavaScript, HTML, and CSS) to provide one or more functions via a network for the end user using a browser client. It can be as basic as Google's search engine, or as complex as a SaaS word processor that allows users to store information and download the document to their personal hard drive. [22]

There are various advantages that web-based applications can provide to their users. This type of application allows users to have a wider range of accessibility, in the sense that the web app can be accessed on either desktop or mobile devices, so long as an internet connection is present. Another benefit of using web applications is the ease of customization. Customization of web apps allows flexibility in changing or modifying to meet user satisfaction accordingly. Due to web applications being a part of SaaS, its implementation across the organization is easy. Once the program has been installed on the host server, access is given. Without having to upgrade each machine, this can be applied each time through the host server. It allows for simpler upgrades and maintenance. Then, enhancing the application's ability to scale with your organization is as simple to do as making upgrades. Your web-based software can enable it as you need more processes to happen at once. Furthermore, where problems arise, servers can be totally changed without having an impact on the overall operating system. As a result, any downtime you might otherwise have been reduced. [23]

The study conducted will be implemented as a web-based application due to the various advantages that it is able to provide compared to desktop or mobile devices, even with the type of electronic devices that is used. Web apps are truly flexible and are much simpler in various aspects that provide users their needs and satisfaction, which is why for more accessibility, the researchers lean towards utilizing web-based apps.

2.2.2. *Leaf Image Processing to Identify Plants*

Digital image processing is the use of a digital computer to process digital images through an algorithm [24]. It can be used for a variety of different applications, including pattern and object recognition, feature extraction, and data compression. The most successful algorithm to classify plants using images of leaves in 2019 is by using Support Vector Machines (SVM) [25]. Support Vector Machine is a supervised learning algorithm that can be used to classify images into groups based on their similarities or dissimilarities with other images in those groups. The idea behind this method is that it can be used to learn from data and make predictions on unseen data.

III. Theoretical Framework

The research is anchored on the study namely, “What is a Leaf?” by E. Mendelson, C. Zumajo-Cardona, and B. Ambrose, which stated that leaves are unique and vary in different aspects. It differs in size, shape, color, and more, and the habitat and weather of the plant affect its leaves [14]. This explains that leaves are possible to be classified.

Additionally, the study is also supported by the article of R. Gonzalez entitled, “Digital Image Processing”. It declares that digital image processing (DIP) can be used for pattern and object recognition, feature extraction, and data compression, using an algorithm like Support Vector Machines (SVM) and You Only Look Once (YOLO) [24]. This means that through DIP, the researchers can develop a program that recognizes plants through leaves.

Moreover, this research is on the study called, “You Only Look Once: Unified, Real-Time Object Detection” by J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. The paper presented the YOLO Algorithm as a single neural network that predicts bounding boxes and class probabilities directly from full images in one evaluation. It also states that the base model of YOLO, processes images in real-time at 45 frames per second (fps), and the Fast YOLO, processes an astounding 155 fps. Additionally, the paper said that YOLO makes more localization errors but is far less likely to predict false detections where nothing exists and it outperforms all other detection methods, including DPM and R-CNN [26]. This proves that using the YOLO algorithm, the program that the researchers will develop has the possibility to be accurate.

IV. Methodology

4.1. *Experimental Materials*

Five leaf images were collected from a plantation of one of the researchers located in Cavite, Philippines. One-Hundred Leaf images were captured during August 11, 2022, with an average temperature of 26.4°C (79.52°F). A Xiaomi Note 10s camera was used to photograph the leaves at different angles, backgrounds, shades, and positions, with an image resolution of 3472 x 3472 pixels. To ensure the representativeness of the image set, they were collected under natural light conditions. Below is the date and the time range the leaf was captured:

Table 4.1 Time and Date of Leaf Captured

Date	Time	Leaf

August 11, 2022	16:25-16:42	Kamias
	16:44-16:52	Oregano
	17:02-17:13	Mango
	17:14-17:28	Golden Pothos
	17:34-17:41	Croton

There was a total of 500 leaf images employed in the data set including these five images in the figure below:

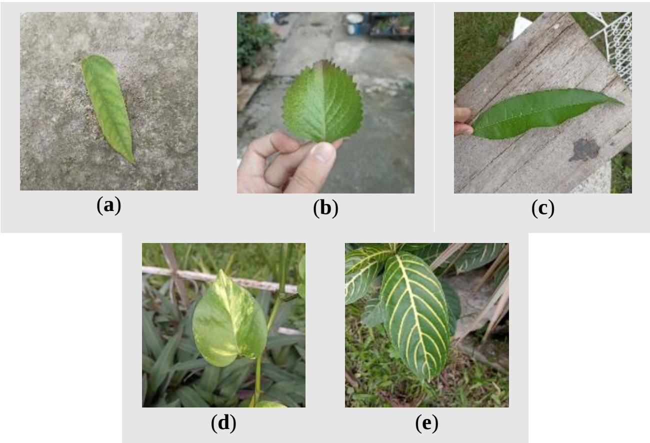


Figure 4.1 Captured Leaf Images. (**a**) Kamias leaf image; (**b**) Oregano leaf image; (**c**) Mango leaf image; (**d**) Golden Pothos leaf image; (**e**) Croton leaf image.

Additionally, five background images were added resulting to a total of 505 images employed in the data set. It was used to perform a more accurate data processing. Images added are show in the figure below:

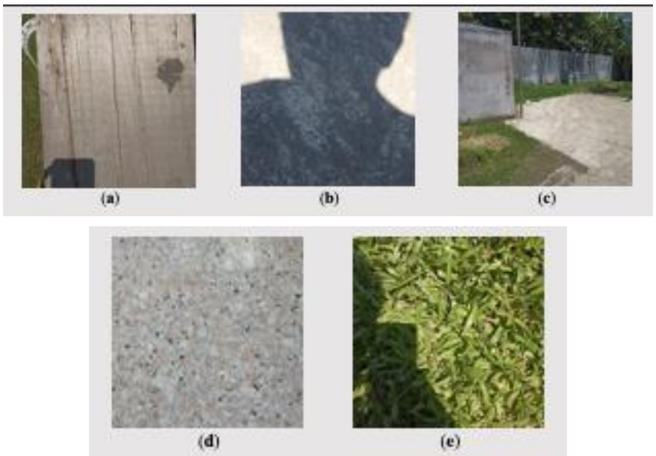


Figure 4.2 Captured Background Images. (**a**) Background image 1; (**b**) Background image 2; (**c**) Background image 3; (**d**) Background image 4; (**e**) Background image 5.

4.2. *DATA IMAGE PROCESSING*

Several image sizes were used in the YOLOv5 network algorithm/model, including the following: original size with 3472x3472 px; half of the original size (1,736x1,736 px); 1280x1280 px; and 640x640 p. All images were rescaled and processed with train.py (note that train python was used to improve the detection of the small targets). The original size 3472x3472 px and half of the original size results in a runtime error while 1280x1280 px and 640x640 px results in successful image processing. Both procedures produced varying levels of accuracy, with 1280x1280 px producing results that were far more accurate than 640x640 p. However, 1280x1280 px was executed at a slower rate of runtime, so we researchers rescaled the image to 1080x1080 px and obtained an accuracy value similar to that of the 1280x1280 px image but at a rate that was significantly faster. The processing results are shown in Figure 3.

Figure 4.3 Image Processing Result. (**a**) Image Processing Result 1; (**b**) Image Processing Result 2; (**c**) Image Processing Result 3.

4.3. *EXPERIMENTAL EQUIPMENT*

A desktop computer was used as the processing platform, and the algorithm/model used was the YOLOv5. The program was written in Python 3.8 using Google Colaboratory and the hosting platform used to collaborate with was GitHub. Since Google Colaboratory was used for the image processing, no local processor or graphics card computing was required because Google offers the usage of its free GPU for data processing. The specific configurations are provided in table 2.

Table 4.2 Test environment settings

Parameter	Configuration
Image Processing Algorithm/Model	YOLOv5
Programming Language	Python 3.8
Hosting Collaboration Tool	GitHub
Collaboration Tool	Google Colaboratory
Processing Platform	Desktop Computer

4.4. *Experimental Process*

First, the manual labeling method was used to mark each leaf images to obtain training label images. The training set was inputted into the YOLOv5 network of different structures for training. The training process was divided into 4 batches. Subsequently, the performance of the network model was determined using the test set. The test set will also determine the data analysis and post-processing of the experiment. The test process is shown in Figure 4.

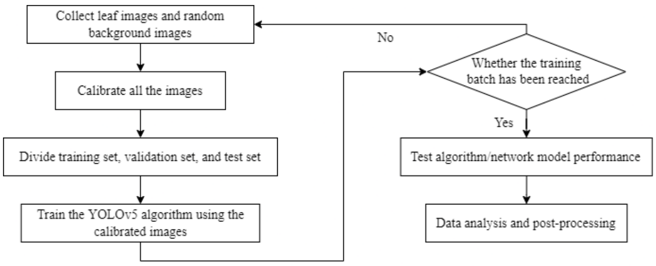


Figure 4.4 Experimental Process Flow Chart

V. Results and Discussion

In this study, we first annotated the data. The process of annotating data entails marking a specific item (leaf) on the photographs of the dataset. The researchers split the information into three parts. 30% of the data is used for validation, 10% for testing, and 60% is used for training. The four models in YOLOv5 are YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x. The improved YOLOv5 model is trained on a cloud environment using Google Colaboratory and Python 3.8, invoking libraries such as CUDA, PyTorch, OpenCV, and others to test and train the plant identification, detection, and classification model.

5.1. *Confusion Matrix*

Figure 5.1 Confusion Matrix Graph

A confusion matrix is a tabular way of visualizing the prediction model's performance. Each entry in a confusion matrix denotes the number of predictions. There is almost no class confusion at 90% confidence level. A confidence level of 0.04 or 4% results in the vast majority of detections being False Positives, while a confidence level at 25% results in higher confusion, but not necessarily between classes.

True Positive (TP) refers to the number of predictions where the model correctly predicts the positive class as positive.

True Negative (TN) refers to the number of predictions where the model correctly predicts the negative class as negative.

False Positive (FP) refers to the number of predictions where the model incorrectly predicts the negative class as positive.

False Negative (FN) refers to the number of predictions where the model incorrectly predicts the positive class as negative.

5.1.1. *Confusion Matrix Interpretation*

Expressing confidence level as percentage chance:
Prediction Chance = Confidence Level * 100

A. *Kamyas Prediction = 0.78 * 100 = 78*

The image classification model has a 78% chance of predicting the Kamyas leaf correctly, while has a 22% chance of mistakenly predicting it as a Mango leaf.

B. *Oregano Prediction = 0.96 * 100 = 96*

The image classification model has a 96% chance of predicting the Oregano leaf correctly, while has a 11% chance of mistakenly predicting it as a Kamyas leaf, and 44% chance of predicting it as background FP.

C. *Mango Prediction = 0.67 * 100 = 67*

The image classification model has a 67% chance of predicting the Mango leaf correctly, while has a 4% chance of mistakenly predicting it as a Croton leaf, and 22% chance of predicting it as background FP.

*D. Golden Pothos Prediction = 0.5 * 100 = 50*

The image classification model has a 50% chance of predicting the Golden Pothos leaf correctly, while has a 4% chance of mistakenly predicting it as an Oregano leaf and 38% chance as a Croton leaf, and 33% chance of predicting it as background FP.

*E. Croton Prediction = 0.19 * 100 = 19*

The image classification model has a 19% chance of predicting the Croton leaf correctly.

*F. Background FN Prediction = 0.5 * 100 = 50*

The image classification model has a 11% chance of mistakenly predicting the Background FN (False Negative) as a Kamyas leaf and Mango Leaf, 50% chance as a Golden Pothos Leaf, and 38% chance as a Mango leaf.

5.2. RECALL-CONFIDENCE CURVE

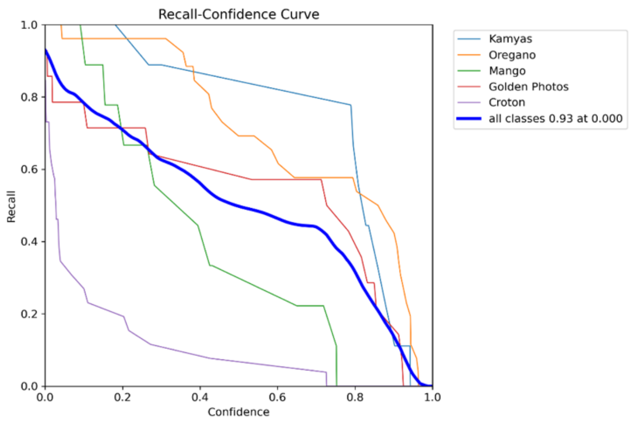


Figure 5.2 Recall-Confidence Curve Graph

Recall is the ratio of the number true positives to the total number of actual objects. The diagrams show the relation of recall values to the confidence level of the model. For all classes, the curve is going down as it approaches the 100% confidence level, this means that as the confidence level increases, the number of true positives decreases.

5.2.1. RECALL-CONFIDENCE CURVE INTERPRETATION

Confidence Percentage = Confidence Value * 100
Recall Percentage = Recall Value * 100

A. Kamyas

From 0% up to less than 20% of confidence, the recall rate is at 100%. However, as it approaches to less than 30% of the confidence rate, the recall percentage dipped down into almost 90%. From that point, up to almost 80% of confidence percentage, the rate of recall only goes down up to 80%. However, from that moment on, it dips down up to 0% of recall percentage as it approaches to 100% of confidence rate.

B. Oregano

At 0% up to almost 30% of confidence percentage, the rate of recall is at 100% to 90%. However, from that point, the percentage of recall dipped down into less than 60% as it approaches to 70% confidence. It stayed at that recall percentage until 80% confidence until it dipped down when it approaches to 100% confidence.

C. Mango

At 0% up to almost 10% of confidence rate, the percentage of recall is at 100%. However, from that point, the recall percentage dipped down into less than 40% at nearly 40% confidence. The slope of the curve from that point is not that large wherein it just dropped to 20% of recall rate as it approaches to 70% confidence. Unfortunately, from that point, it just dropped to 0% of recall percentage as it moves just a little from 70% confidence.

D. Golden Pothos

From 0% up to almost 5% of confidence percentage, the recall rate has already dipped down into less than 80%. Fortunately, from that point, the percentage of recall only dipped into almost 60% as it approaches to 80% confidence. However, the recall percentage just dropped quickly to 0% as it approaches to 90%.

E. Croton

From 0% up to less than 10% of confidence percentage, the recall rate has already dropped to almost 30% and from that point the recall percentage just gradually dropped to less than 10% as it approaches to almost 70% confidence. At the end point, it just quickly dropped to 0% recall percentage as it moves just a little in the confidence level.

5.3. PRECSION-CONFIDENCE CURVE

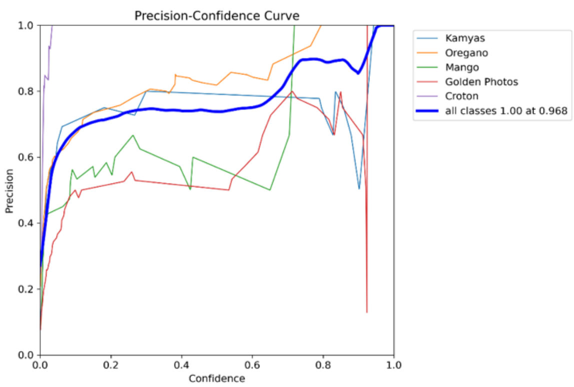


Figure 5.3 Precision-Confidence Curve Graph

The diagram shows the relation between the precision and confidence level of the model trained. The precision refers to the number of true positives divided by the total number of positive predictions. For all classes, at 10% confidence level up to 80%, 60% to 80% of the total positive predictions are true positives.

5.3.1. PRECISION-CONFIDENCE CURVE INTERPRETATION

Confidence Percentage = Confidence Value * 100
Precision Percentage = Precision Value * 100

A. *Kamyas*

At 0% up to less than 20% of confidence, the precision rate is at around 0% to below 80%. However, as it approaches to less than 30% of the confidence rate, the recall percentage dipped down slightly below 70%. From 20% to 40% confidence level, the precision rate reached to almost 80%, and at that point, it stayed consistent until confidence level almost reached 80%. At 80% confidence level to 100% level, the precision levels are starting to go up and down, in between the range of below 80% to below 60%, until it ends at 100% precision level.

B. *Oregano*

At 0% up to less than 20% of confidence, the precision rate reached to below 80%. From 20% to 40% confidence level, the precision rate reached over 80%. At 40% to 60% confidence level, the precision level is steadily above 80%. Then, in between 60% to 80% confidence levels, the precision level continued to rise to 100%, and it has steadily been there until the end.

C. *Mango*

From 0% up to less than 20% of confidence, the precision rate reached to above 40%. From 20% to 40% confidence level, the precision rate reached above 60%. At 40% to 60% confidence level, the precision level dipped below 60%. Then, in between 60% to 80% confidence levels, just like Oregano, the precision level continued to rise to 100%.

D. *Golden Pothos*

From 0% up to less than 20% of confidence, the precision rate reached to a level in between 40% to 60%. It remained steady at that precision level when the confidence level is at 20% to 40%. When it is at 40% to 80% confidence level, the precision level remained steady, until it almost reaches the 60% mark, where precision rate rises at almost 80%. Then, in between 80% to 100% confidence levels, the precision level greatly fell to below 20%, but quickly rose up to 100%.

E. *Croton*

Croton quickly rose to 100% precision level. The confidence level has not even reached over 10%, but the precision rate rises to above 100%.

5.4. PRECISION-CONFIDENCE CURVE

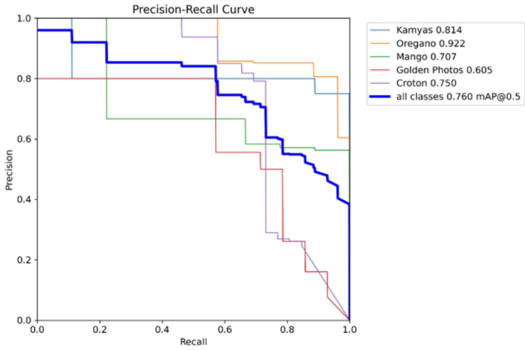


Figure 5.4 Precision-Recall Curve Graph

As the name implies, precision-recall curve shows the measurement of consistency of the result throughout tests. The mean average precision is the area under the curve. This means that the more the algorithm gets correct answer, the recall curve rises, but the precision curve falls. With this, the researchers conclude that the more the correct answer the algorithm gets, the more the precision lowers since it shows inconsistency compared to its current result.

5.4.1. PRECISION-RECALL CURVE INTERPRETATION

Precision Percentage = Precision Value * 100
Recall Percentage = Recall Value * 100

A. *Kamyas*

At 0% up to less than 20% of recall level, the precision rate dropped from 100% to 80%. From 20% to less than 80% recall level, the precision rate stabilized until 80% to 100%. At 80% to 100% of recall level, the precision rate slightly dropped below 80% until the end.

B. *Oregano*

From 0% up to less than 20% of confidence, the precision rate reached to above 40%. From 20% to 40% confidence level, the precision rate reached above 60%. At 40% to 60% confidence level, the precision level dipped below 60%. Then, in between 60% to 80% confidence levels, just like Oregano, the precision level continued to rise to 100%.

C. *Mango*

From 0% up to less than 20% of confidence, the precision rate reached to above 40%. From 20% to 40% confidence level, the precision rate reached above 60%. At 40% to 60% confidence level, the precision level dipped below 60%. Then, in between 60% to 80% confidence levels, just like Oregano, the precision level continued to rise to 100%.

D. *Golden Pothos*

At 0% up to less than below 60% of recall level, the precision rate rises and stabilizes at 80%. When recall rate nearly reaches 60%, its precision rate dropped to below 60%. From 60% to 80% recall level, the precision rate dropped consecutively, reaching as low as below 30%. Then, when recall level reached 80%, the precision continued to dip to as low as 10%. After reaching 10%

precision level, it continued to drop to 0%, when confidence level is near 100%.

E. Croton

At 0% up to around 50% recall rate, the precision rate is consistent at a 100%. Then from 50% to 60% recall rate, the precision dropped below 100% level and nearing the 80% mark. When the recall rate reaches 60% to 80%, there are consecutive drops in the precision rate, with one long drop, reaching below 30%. After that, from 80% to 100% recall rate, the precision level has small consecutive dips, before finally dropping to 0.

VI. Conclusions and Recommendations

Given the data presented in the previous section, it clearly shows that the Oregano leaf has the highest chance of getting predicted correctly by the model. It has the best grade in the confusion matrix, it approaches to 80% confidence at 60% recall, it is 100% precise from 80% confidence and it reached 80% precision rate at 90% recall which is a high recall percentage. The class that the model found it hard to predict is the Croton. It is the lowest in confusion matrix, the recall rate quickly dropped below 40% at the recall-confidence curve, and the precision rate just dropped from 80% to 30% at 70% recall.

As explained in the previous section the researchers conclude that by utilizing the YOLOv5 algorithm, digital image processing can be used to identify plants through the images of their leaves. Supported by the results, the study shows that with the implementation of the algorithm to the image classification project, it can be observed that the accuracy of the system is high.

As explained in Chapter 1, Scope and Limitations, a lot of considerations were factored-out such as the image format including PNG, GIF, BMP, and the likes, the image quality, the time the image was taken, etc. of the research due to time constraints. With this, the researchers recommend to future developers and researchers whose projects are in line with what are discussed in the paper to take into account such factors to fully determine the accuracy and the effectiveness of the proposed model and finished product.

Acknowledgment

First and foremost, praises and thanks to the **God**, the Almighty, for His continuous blessings throughout our research work to complete the research successfully.

We would also like to express our deep and sincere gratitude to **our families** who continuously supported us, the researchers, emotionally and financially. Their support is what kept us going through this research.

We would also like to acknowledge the efforts and patience of **everyone in our group.** Without their contributions, we highly doubt that this research will be a success.

Last but not the least, we would like to extend our gratitude to **Engr. Eufemia A. Garcia** for her continuous guidance to not only our group, but to all other groups within Block 2. Her thoroughness inspired us to be thorough in our research as well.

Any attempts to pursue this research would not be possible without the help and support of every people stated above. Hence, we, as a group, are humbled and sincerely give our thanks to them.

References

[1] I. Kharir, V. Z. Swu, and D. J. Bora, "Identification of Different Plants through Image Processing Using Different Machine Learning Algorithms," vol. 43, no. 3, Dec. 2020.

[2] F. Fida, M. R. Talib, F. Fida, N. Iqbal, M. Aqeel, M. N. Iqbal, and A. Noman, "Leaf Image Recognition Based Identification of Plants: Supportive Framework for Plant Systematics," vol 3, no. 3, pp. 125-131, Aug. 2018.

[3] J. D. S. Selda, R. M. R. Ellera, L. C Cajayon II, and N. B. Lisangan, "Plant Identification by Image Processing of Leaf Veins," *Proceedings of the International Conference on Imaging, Signal Processing and Communication - ICISPC*, Jul. 2017. doi:10.1145/3132300.3132315

[4] S. Pramanik, S. K. Bandyopadhyay, D. Bhattacharyya, and T. Kim, "Identification of Plant Using Leaf Image Analysis," *Signal Processing and Multimedia*, pp. 291-303, Aug. 2010.

[5] K. Bahadar, M. Fatima, A. Noman, S. Abbas, A. Raza, T. Saleem, R. Sarfraz, K. Khan, and M. Zaynab, "The Anatomical Examination of Leaf Epidermis of Genus Plantago L. (Ispaghol) from Pakistan," vol. 3, no. 2, pp. 57-60, May 2018.

[6] Z. Wang, H. Li, Y. Zhu, and T. Xu, "Review of Plant Identification Based on Image Processing," *Arch Computat Methods Eng.*, vol 24, no 3, pp. 637-654, Jul. 2016.

[7] C. Fernandez, "The Paradox of Choice: Why More is Less," *The Journal for Decision Makers*, vol. 42, no. 4, pp. 265-267, Jun. 2019.

[8] National Research Council (US), "Plant Biology Research and Training for the 21st Century," *Committee on Examination of Plant Science Research Programs in the United States*, Available from: https://www.ncbi.nlm.nih.gov/books/NBK234814. [Accessed: Aug. 2, 2022].

[9] National Research Council (US), "Plant Biology Research and Training for the 21st Century," *Committee on Examination of Plant Science Research Programs in the United States*. Available from, https://www.ncbi.nlm.nih.gov/books/NBK234814. [Accessed: Aug. 1, 2022].

[10] University of Washington, "Plant Research Biosafety," *Biological Research Safety*. Available from, https://ehs.washington.edu/biological/biological-research-safety/plant-research-biosafety. [Accessed: Aug. 1, 2022].

[11] L. M. Kelly, B. A. Ambrose, and D. W. Stevenson, "Why Study Plants?," Available from, https://www.nybg.org/blogs/science-talk/2014/06/why-study-plants.[Accessed: Aug. 1, 2022].

[12] M. Nakano, "Introduction to Plant Identification," *Red Seal Landscape Horticulturist Identify Plants and Plant Requirements (F2 - 1&2)*, 2020.

[13] H. Marceau, "The Importance of Plant Identification – Part 1," *Popularization*, Oct. 2014. Available from: https://phytochemia.com/en/2014/10/20/the-importance-of-plant-identification-part-1. [Accessed: Aug. 3, 2022].

[14] E. Mendelson, C. Zumajo-Cardona, and B. Ambrose, "What is a Leaf?", *Frontiers for Young Minds*, vol. 10, Mar 2022.

[15] R. Malinowski, "Understanding of Leaf Development—the Science of Complexity," *Plants*, vol. 2, no. 3, pp. 396–415, Jun. 2013. doi: 10.3390/plants2030396.

[16] "*Carotenoids*", Linus Pauling Institute. [Accessed August 6, 2022].

[17] R. Ali, R. Hardie, and A. Essa, "A Leaf Recognition Approach to Plant Classification Using Machine Learning," *NAECON 2018 - IEEE National Aerospace and Electronics Conference*, Jul. 2018.

[18] S. Zhang, W. Huang, Y. Huang, and C. Zhang, "Plant species recognition methods using leaf image: Overview," 2020. Available from: https://www.sciencedirect.com/science/article/abs/pii/S0925231220304902.

[19] C. Gwo and C. Wei, "Plant identification through images: Using feature extraction of key points on leaf contours 1," *Botanical Society of America,* Sep. 2013. Available from: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4103455.

[20] B. L. Grant, "Leaf Identification – Learn About Different Leaf Types In Plants," Apr. 2022.

[21] "Identifying Native Plants," Available from: https://www.sciencelearn.org.nz/resources/2634-identifying-native-plants. [Accessed: Aug. 1, 2022].

[22] R. Sturm and J. Craig, "Chapter 7 - Managing Web-Based Applications," *Application Performance Management (APM) in the Digital Enterprise*, C. Pollard , Ed. pp. 83–93, 2015.

[23] P. Khamooshi, "The benefits of using web-based applications," *Geeks Insights.* [Online]. Available from: https://www.geeks.ltd.uk/insights/the-benefits-of-using-web-based-applications. [Accessed: Aug. 14, 2022].

[24] R. Gonzalez, "Digital image processing," *New York, NY: Pearson*, 2018.

[25] E. Yigit, K. Sabanci, A. Toktas, and A. Kayabasi, "A Study on Visual Features of Leaves in Plant Identification Using Artificial Intelligence Techniques," *Computers and Electronics in Agriculture*, 156, pp. 369–377. doi:10.1016/j.compag.2018.11.036.

[26] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," Computer Vision and Pattern Recognition, Jun. 2015. https://doi.org/10.48550/arXiv.1506.02640

Appendix A: Train Batches, Labels and predictions using the YOLOv5 Model

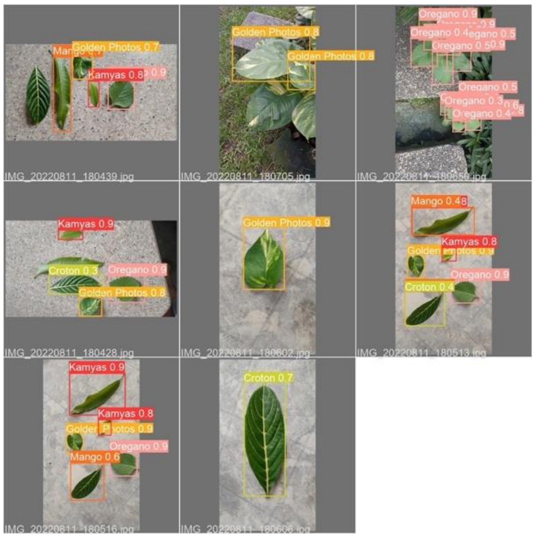
A1. Train Batches

A2. Labels and Predictions

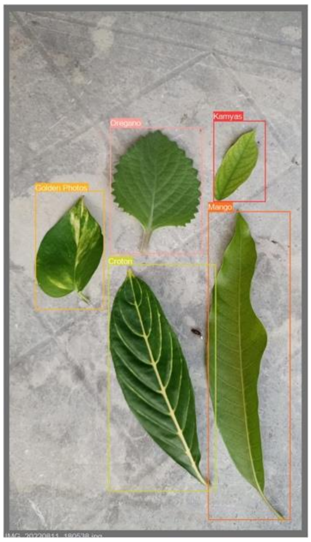
Label 1

Prediction 1

Labels 2



Predictions 2

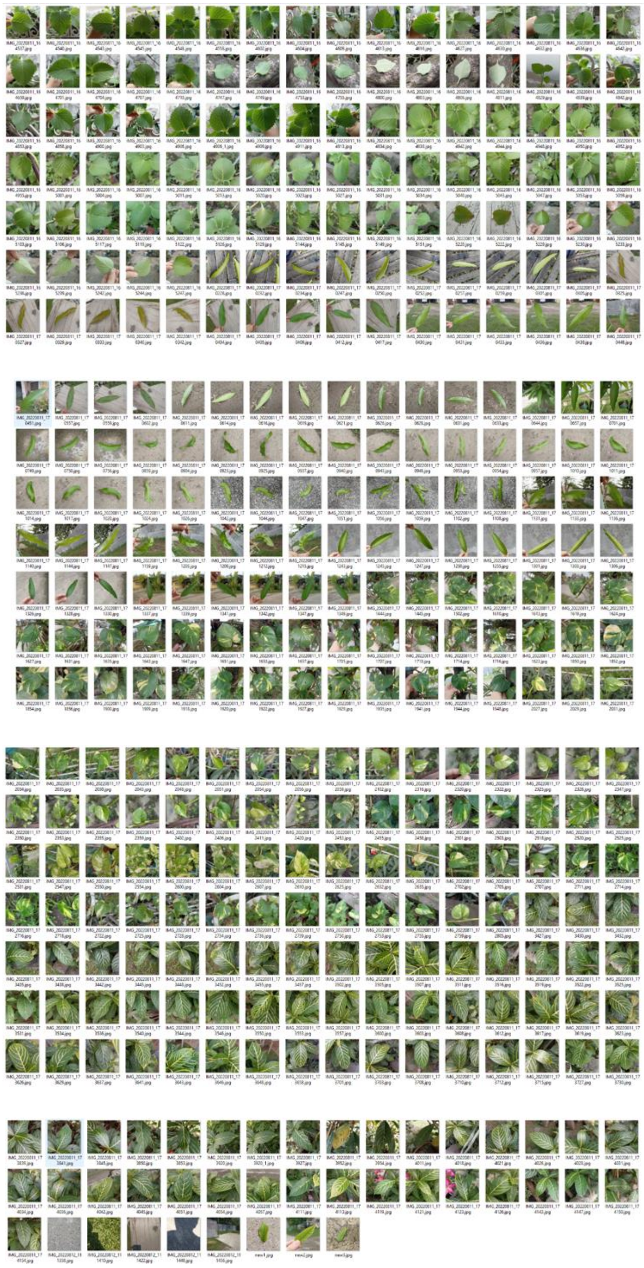


Label 3

Prediction 3

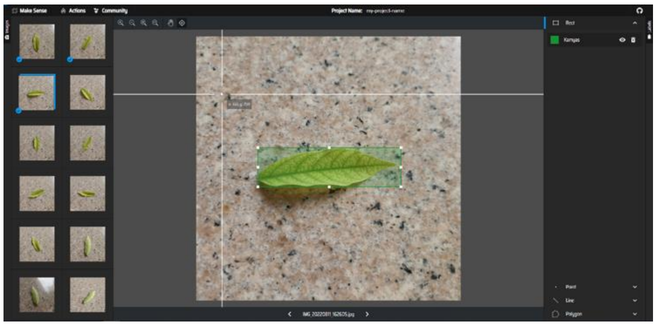
APPENDIX B: YOLOV5 TRAINING

B1. TRAIN DATASET

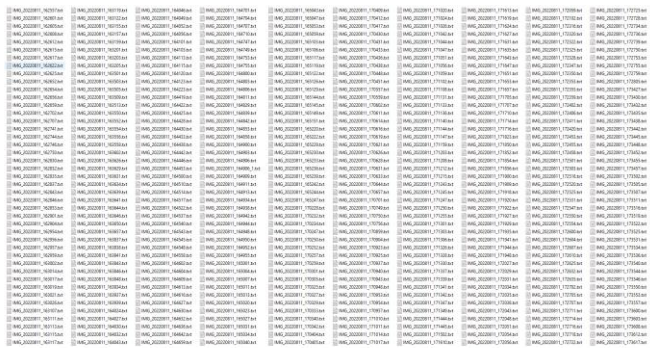


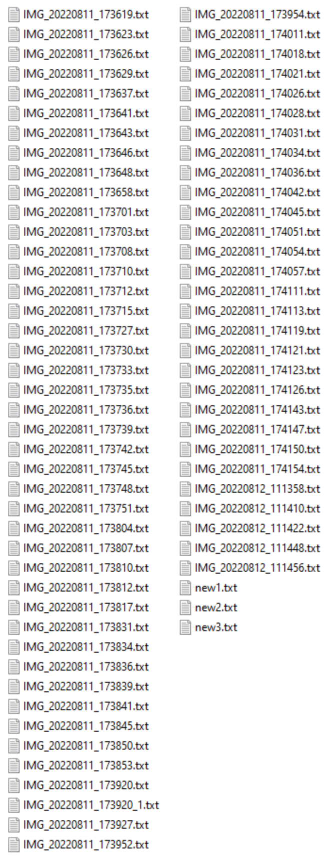
B2. VALIDATION DATASET

B3. ANNOTATION



B4. TRAIN LABELS

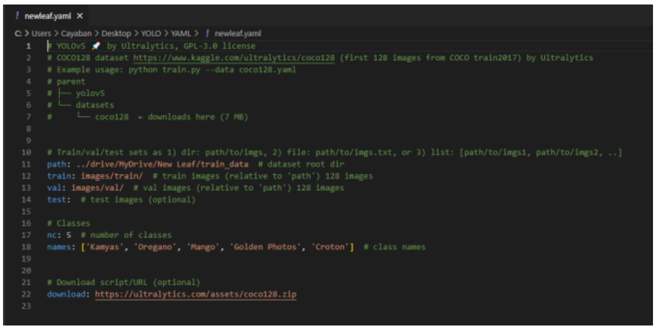




B5. Validation Labels

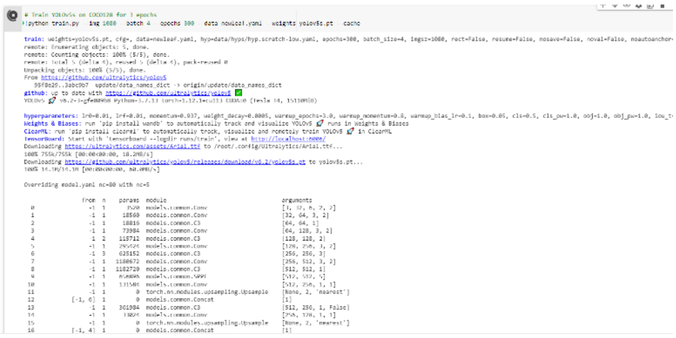
IMG_20220811_180418.txt	11/08/2022 3:54 pm	Text Document	1 KB
IMG_20220811_180428.txt	11/08/2022 3:54 pm	Text Document	1 KB
IMG_20220811_180439.txt	11/08/2022 3:54 pm	Text Document	1 KB
IMG_20220811_180451.txt	11/08/2022 3:54 pm	Text Document	1 KB
IMG_20220811_180454.txt	11/08/2022 3:54 pm	Text Document	1 KB
IMG_20220811_180513.txt	11/08/2022 3:54 pm	Text Document	1 KB
IMG_20220811_180516.txt	11/08/2022 3:54 pm	Text Document	1 KB
IMG_20220811_180522.txt	11/08/2022 3:54 pm	Text Document	1 KB
IMG_20220811_180538.txt	11/08/2022 3:54 pm	Text Document	1 KB
IMG_20220811_180602.txt	11/08/2022 3:54 pm	Text Document	1 KB
IMG_20220811_180606.txt	11/08/2022 3:54 pm	Text Document	1 KB
IMG_20220811_180610.txt	11/08/2022 3:54 pm	Text Document	1 KB
IMG_20220811_180614.txt	11/08/2022 3:54 pm	Text Document	1 KB
IMG_20220811_180619.txt	11/08/2022 3:54 pm	Text Document	1 KB
IMG_20220811_180650.txt	11/08/2022 3:54 pm	Text Document	1 KB
IMG_20220811_180705.txt	11/08/2022 3:54 pm	Text Document	1 KB
IMG_20220811_180719.txt	11/08/2022 3:54 pm	Text Document	1 KB

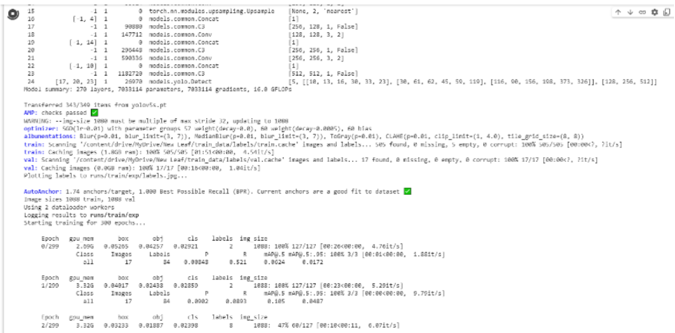
B6. YAML Files



```
newleaf.yaml
C: > Users > Cayaban > Desktop > YOLO > YAML > newleaf.yaml
# YOLOv5 🚀 by Ultralytics, GPL-3.0 license
# COCO128 dataset https://www.kaggle.com/ultralytics/coco128 (first 128 images from COCO train2017) by Ultralytics
# Example usage: python train.py --data coco128.yaml
# parent
# ├── yolov5
# └── datasets
#     └── coco128  ← downloads here (7 MB)


# Train/val/test sets as 1) dir: path/to/imgs, 2) file: path/to/imgs.txt, or 3) list: [path/to/imgs1, path/to/imgs2, ..]
path: ../drive/MyDrive/New Leaf/train_data  # dataset root dir
train: images/train/  # train images (relative to 'path') 128 images
val: images/val/  # val images (relative to 'path') 128 images
test:  # test images (optional)

# Classes
nc: 5  # number of classes
names: ['Kamyas', 'Oregano', 'Mango', 'Golden Photos', 'Croton']  # class names


# Download script/URL (optional)
download: https://ultralytics.com/assets/coco128.zip
```

B7. YOLOv5 Application

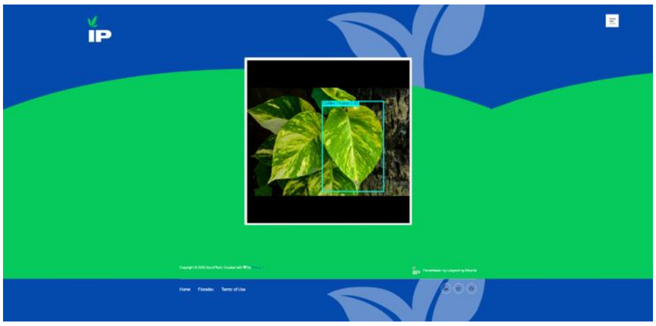




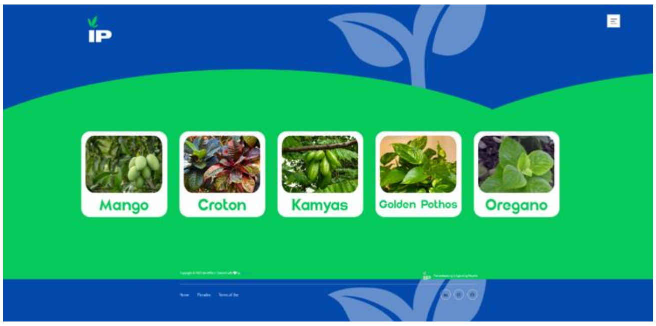
```
Model summary: 213 layers, 7023610 parameters, 0 gradients, 15.8 GFLOPs
                 Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 3/3 [00:00<00:00,  3.22it/s]
                   all         17         84      0.703      0.757       0.76      0.546
                Kamyas         17          9      0.727          1      0.814      0.584
               Oregano         17         26      0.723      0.962      0.922      0.701
                 Mango         17          9      0.559      0.889      0.707      0.426
         Golden Photos         17         14      0.504      0.714      0.605      0.491
                Croton         17         26          1      0.221       0.75      0.529
Results saved to runs/train/exp2
```

Appendix C: Identiplant User Interface

C1. Home Page



C2. Floradex



C3. Plants Page

Appendix D: Raw Code for Web Application

D1. Floradex Page HTML

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
```

```
  <meta name="viewport" content="width=device-width,
initial-scale=1, shrink-to-fit=no">
  <meta name="theme-color" content="#000000">
  <title>IdentiPlant</title>
  <!-- GOOGLE FONT -->
  <link
href="https://fonts.googleapis.com/css?family=Roboto+Co
ndensed:300,300i,400,400i,700,700i" rel="stylesheet">
  <link
href="https://fonts.googleapis.com/css?family=Dosis:200,3
00,400,500,600,700" rel="stylesheet">
  <!-- CSS LIBRARY -->
  <link rel="stylesheet" type="text/css" href="css/font-
awesome.min.css">
  <link rel="stylesheet" type="text/css"
href="css/ionicons.min.css">
  <link rel="stylesheet" type="text/css" href="css/Pe-icon-7-
stroke.min.css">
  <link rel="stylesheet" type="text/css" href="css/slick-
theme.css">
  <link rel="stylesheet" type="text/css" href="css/slick.css">
  <link rel="stylesheet" type="text/css"
href="css/owl.carousel.min.css">
  <link rel="stylesheet" type="text/css"
href="css/bootstrap-select.min.css">
  <link rel="stylesheet" type="text/css"
href="css/animation.css">
  <!-- MAIN STYLE -->
  <link rel="stylesheet" href="css/floradex.css">
  <link rel="stylesheet" href="css/styles.css">
  <!--
      manifest.json provides metadata used when your web
app is added to the
      homescreen on Android. See
https://developers.google.com/web/fundamentals/engage
-and-retain/web-app-manifest/
    -->
  <link rel="manifest"
href="%PUBLIC_URL%/manifest.json">
  <link rel="shortcut icon"
href="%PUBLIC_URL%/favicon.ico">
  <!--
      Notice the use of %PUBLIC_URL% in the tags above.
      It will be replaced with the URL of the `public` folder
during the build.
      Only files inside the `public` folder can be referenced
from the HTML.

      Unlike "/favicon.ico" or "favicon.ico",
"%PUBLIC_URL%/favicon.ico" will
      work correctly both with client-side routing and a non-
root public URL.
      Learn how to configure a non-root public URL by
running `npm run build`.
    -->
  <title>React App</title>
</head>
<body class="blog-v2">
  <noscript>
    You need to enable JavaScript to run this app.
  </noscript>
  <!--
      This HTML file is a template.
      If you open it directly in the browser, you will see an
empty page.

      You can add webfonts, meta tags, or analytics to this
file.
      The build step will place the bundled scripts into the
<body> tag.

      To begin the development, run `npm start` or `yarn
start`.
      To create a production bundle, use `npm run build` or
`yarn build`.
    -->
  <header id="main-header" class="header-v6 hidden-sm
hidden-xs">
    <div class="container-fluid">
      <div class="inner row">
        <div class="logo col-lg-2 col-md-2">
          <a href="index.html" title="logo"><img alt="logo-
theme" src="images/logo.png" class="img-responsive"
/></a>
        </div>
        <div class="header-right col-lg-10 col-md-10">
          <div class="main-menu">
            <div id="menu-popup">
              <div class="burger-menu"><span></span></div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </header>

  <header id="header_mobile" class="header-mobile-
default hidden-lg hidden-md">
    <div class="header-top">
      <div class="container">
        <div class="logo text-center">
          <a href="index.html" title="logo"><img alt="logo-
theme" src="images/logo.png" class="img-responsive"
/></a>
          <a href="Floradex.html" title="logo"><img alt="logo-
theme" src="images/FloraDex/Floradex.png"
              class="img-responsive" /></a>
        </div>
      </div>
    </div>
    <div class="header-bottom">
      <div class="container">
        <div class="inner">
          <div class="header-main">
            <div class="main-left">
```

```html
            <button data-toggle="offcanvas" class="btn btn-
offcanvas btn-toggle-canvas offcanvas" type="button">
              <i class="ion ion-android-menu"></i>
            </button>
          </div>
          <div class="main-right">

          </div>
        </div>
      </div>
    </div>
  </div>
</header>

<section class="blog-zigzag ">
  <div id="plantcontainer">
    <a href="Mango.html">
      <img src="images/FloraDex/Mango.png"
alt="mango.png" id="plants">
    </a>
    <a href="Croton.html">
      <img src="images/FloraDex/Croton.png"
alt="croton.png" id="plants">
    </a>
    <a href="Kamyas.html">
      <img src="images/FloraDex/Kamyas.png"
alt="kamyas.png" id="plants">
    </a>
    <a href="Golden-Pothos.html">
      <img src="images/FloraDex/Golden Pothos.png"
alt="goldenphotos.png" id="plants">
    </a>
    <a href="Oregano.html">
      <img src="images/FloraDex/Oregano.png"
alt="oregano.png" id="plants">
    </a>
  </div>
</section>
<div id="content_menu_popup" class="content-menu-
popup hidden-sm hidden-xs">
  <div class="menu-popup-body">
    <ul class="navbar-menu">
      <li class="items">
        <a href="index.html"><span>Home</span></a>
      </li>
      <li class="items">
        <a href="Floradex.html"><span>Floradex</span></a>
      </li>
      <li class="items">
        <a href="/"><span>Contact Us</span></a>
      </li>
    </ul>
  </div>
</div>

<div id="pbr-off-canvas" class="pbr-off-canvas sidebar-
offcanvas hidden-lg hidden-md">
  <div class="pbr-off-canvas-body">
      <div class="offcanvas-head">
        <button type="button" class="btn btn-close btn-
toggle-canvas" data-toggle="offcanvas">
          <i class="pe-7s-close-circle"></i>
        </button>
        <span>Menu</span>
      </div>
      <nav class="navbar navbar-offcanvas navbar-static">
        <ul class="nav navbar-nav">
          <li class="level1 active hassub">
            <a href="index.html">Home</a>
          </li>
          <li class="level1 hassub">
            <a href="Floradex.html">Floradex </a>
          </li>
        </ul>
      </nav>
    </div>
  </div>
  <footer class="footer">
    <div class="container">
      <div class="footer-bottom">
        <div class="wrap-copyright">
          <div class="copyright" id="coppy">
            Copyright © 2022 IdentiPlant. Created with <i
class="fa fa-heart"></i> by<a href="#">Group 4</a>.
          </div>
          <div class="countries" id="coppy">
            <img src="images/logo-white.png" alt="logo"
/>Pamantasan ng Lungsod ng Maynila
          </div>
        </div>
        <div class="row">
          <div class="col-xs-12 col-sm-12 col-md-6 col-lg-8">
            <div class="menu-footer">
              <ul>
                <li><a href="index.html"
id="coppy">Home</a></li>
                <li><a href="Floradex.html"
id="coppy">Floradex</a></li>
                <li><a href="/" id="coppy"> Terms of Use
</a></li>
              </ul>
            </div>
          </div>
          <div class="col-xs-12 col-sm-12 col-md-12 col-lg-4">
            <div class="social">
              <a href="#" title="linkedin">
                <i class="fa fa-linkedin"></i>
              </a>
              <a href="#" title="instagram">
                <i class="    fa fa-instagram"></i>
              </a>
              <a href="https://github.com/proDev-Theron/dip-
plantidentifier" title="GitHub">
                <i class="fa fa-github"></i>
              </a>
            </div>
```

```
        </div>
      </div>
    </div>
  </div>
</footer>

<a href="#" title="sroll" class="scrollToTop"><i class="fa
fa-angle-up"></i></a>

<script type="text/javascript" src="js/jquery-
1.12.4.min.js"></script>
<script type="text/javascript"
src="js/owl.carousel.min.js"></script>
<script type="text/javascript"
src="js/bootstrap.min.js"></script>
<script type="text/javascript"
src="js/slick.min.js"></script>
<script type="text/javascript" src="js/vit-
gallery.js"></script>
<script type="text/javascript"
src="js/jquery.countTo.js"></script>
<script type="text/javascript"
src="js/jquery.appear.min.js"></script>
<script type="text/javascript"
src="js/isotope.pkgd.min.js"></script>
<script type="text/javascript" src="js/bootstrap-
select.js"></script>
<script type="text/javascript"
src="js/slick.min.js"></script>
<script type="text/javascript"
src="js/jquery.littlelightbox.js"></script>

<script type="text/javascript"
src="js/function.js"></script>
</body>

</html>
```

D2. PLANTS PAGE HTML

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1, shrink-to-fit=no">
  <meta name="theme-color" content="#000000">
  <title>Mango</title>
  <!-- GOOGLE FONT -->
  <link
href="https://fonts.googleapis.com/css?family=Roboto+Co
ndensed:300,300i,400,400i,700,700i" rel="stylesheet">
  <link
href="https://fonts.googleapis.com/css?family=Dosis:200,3
00,400,500,600,700" rel="stylesheet">
  <!-- CSS LIBRARY -->
  <link rel="stylesheet" type="text/css" href="css/font-
awesome.min.css">
  <link rel="stylesheet" type="text/css"
href="css/ionicons.min.css">
  <link rel="stylesheet" type="text/css" href="css/Pe-icon-7-
stroke.min.css">
  <link rel="stylesheet" type="text/css" href="css/slick-
theme.css">
  <link rel="stylesheet" type="text/css" href="css/slick.css">
  <link rel="stylesheet" type="text/css"
href="css/owl.carousel.min.css">
  <link rel="stylesheet" type="text/css"
href="css/bootstrap-select.min.css">
  <link rel="stylesheet" type="text/css"
href="css/animation.css">
  <!-- MAIN STYLE -->
  <link rel="stylesheet" href="css/styles.css">
  <link rel="stylesheet" href="css/fruits.css">
  <!--
      manifest.json provides metadata used when your web
app is added to the
      homescreen on Android. See
https://developers.google.com/web/fundamentals/engage
-and-retain/web-app-manifest/
    -->
  <link rel="manifest"
href="%PUBLIC_URL%/manifest.json">
  <link rel="shortcut icon"
href="%PUBLIC_URL%/favicon.ico">
  <!--
      Notice the use of %PUBLIC_URL% in the tags above.
      It will be replaced with the URL of the `public` folder
during the build.
      Only files inside the `public` folder can be referenced
from the HTML.

      Unlike "/favicon.ico" or "favicon.ico",
"%PUBLIC_URL%/favicon.ico" will
      work correctly both with client-side routing and a non-
root public URL.
      Learn how to configure a non-root public URL by
running `npm run build`.
    -->
  <title>React App</title>
</head>
<body class="blog-v2">
    <noscript>
      You need to enable JavaScript to run this app.
    </noscript>
    <!--
        This HTML file is a template.
        If you open it directly in the browser, you will see an
empty page.

        You can add webfonts, meta tags, or analytics to this
file.
        The build step will place the bundled scripts into the
<body> tag.
```

```
      To begin the development, run `npm start` or `yarn start`.
      To create a production bundle, use `npm run build` or `yarn build`.
    -->
  <header id="main-header" class="header-v6 hidden-sm hidden-xs">
    <div class="container-fluid">
      <div class="inner row">
        <img src="images/Mango/mangbanner.png" id="bgimage">
        <div id="head">
          <a href="index.html">
            <img src="images/ipblue.png" alt="IP" id="home">
          </a>
          <img src="images/Mango/mangotitle.png" alt="mango" id="banner">
        </div>
       <div class="header-right col-lg-10 col-md-10">
        <div class="main-menu">
         <div id="menu-popup">
          <div class="burger-menu" id="menuses"><span></span></div>
         </div>
        </div>
       </div>
      </div>
    </div>
  </header>

  <header id="header_mobile" class="header-mobile-default hidden-lg hidden-md">
    <div class="header-top">
      <div class="container">
       <div class="logo text-center">
        <img src="images/Mango/mangbanner.png" id="bgimage">
        <a href="index.html" title="logo"><img alt="logo-theme" src="images/logo.png" class="img-responsive" /></a>
        <img src="images/Mango/mangotitle.png" alt="mango" id="banner">
       </div>
      </div>
    </div>
    <div class="header-bottom">
      <div class="container">
       <div class="inner">
        <div class="header-main">
         <div class="main-left">
          <button data-toggle="offcanvas" class="btn btn-offcanvas btn-toggle-canvas offcanvas" type="button">
           <i class="ion ion-android-menu"></i>
          </button>
         </div>
         <div class="main-right">
          </div>
         </div>
        </div>
       </div>
      </div>
  </header>
    <div id="bodycontainer">
      <div class="tabs" onclick="overview()">
        Overview
      </div>
      <div class="tabs" onclick="habitat()">
        Habitat
      </div>
      <div class="tabs" id="domestication" onclick="benefits()">
        Benefits
      </div>
    </div>
    <hr>
    <div id="content">
      Mangoes are sweet, creamy fruits that have a range of possible health benefits. They are highly popular around the world.
      The mango is a tropical stone fruit and member of the drupe family. This is a type of plant food with a fleshy outer section that surrounds a shell, or pit. This pit contains a seed.
      Other members of the drupe family include olives, dates, and coconuts.
      There are many different kinds of mango. They vary in color, shape, flavor, and seed size. Although mango skin can be green, red, yellow, or orange, its inner flesh is mostly golden yellow.
      This feature is part of a collection of articles on the health benefits of popular foods. In this article, we explore the many benefits of mangoes, explain their nutritional breakdown, and provide a few healthy recipe ideas.
      <br>
      A mango is an edible stone fruit produced by the tropical tree Mangifera indica which is believed to have originated in the region between
      northwestern Myanmar, Bangladesh, and northeastern India. M. indica has been cultivated in South and Southeast Asia since ancient times resulting in two
      types of modern mango cultivars: the "Indian type" and the "Southeast Asian type". Other species in the genus Mangifera also produce edible fruits that
      are also called "mangoes", the majority of which are found in the Malesian ecoregion. Worldwide, there are several hundred cultivars of mango. Depending on the
      cultivar, mango fruit varies in size, shape, sweetness, skin color, and flesh color which may be pale yellow, gold, green, or orange. Mango is the national fruit of
      India, Pakistan and the Philippines, while the mango tree is the national tree of Bangladesh.
    </div>
```

```
<div id="content_menu_popup" class="content-menu-popup hidden-sm hidden-xs">
  <div class="menu-popup-body">
    <ul class="navbar-menu">
      <li class="items">
        <a href="index.html"><span>Home</span></a>
      </li>
      <li class="items">
        <a href="Floradex.html"><span>Floradex</span></a>
      </li>
      <li class="items">
        <a href="/"><span>Contact Us</span></a>
      </li>
    </ul>
  </div>
</div>

<div id="pbr-off-canvas" class="pbr-off-canvas sidebar-offcanvas hidden-lg hidden-md">
  <div class="pbr-off-canvas-body">
    <div class="offcanvas-head">
      <button type="button" class="btn btn-close btn-toggle-canvas" data-toggle="offcanvas">
        <i class="pe-7s-close-circle"></i>
      </button>
      <span>Menu</span>
    </div>
    <nav class="navbar navbar-offcanvas navbar-static">
      <ul class="nav navbar-nav">
        <li class="level1 active hassub">
          <a href="index.html">Home</a>
        </li>
        <li class="level1 hassub">
          <a href="Floradex.html">Floradex </a>
        </li>
      </ul>
    </nav>
  </div>
</div>
<footer class="footer">
  <div class="container">
    <div class="footer-bottom">
      <div class="wrap-copyright">
        <div class="copyright" id="coppy">
          Copyright © 2022 IdentiPlant. Created with <i class="fa fa-heart"></i> by<a href="#">Group 4</a>.
        </div>
        <div class="countries" id="coppy">
          <img src="images/logo-white.png" alt="logo" />Pamantasan ng Lungsod ng Maynila
        </div>
      </div>
      <div class="row">
        <div class="col-xs-12 col-sm-12 col-md-6 col-lg-8">
          <div class="menu-footer">
            <ul>
              <li><a href="index.html" id="coppy">Home</a></li>
              <li><a href="Floradex.html" id="coppy">Floradex</a></li>
              <li><a href="/" id="coppy"> Terms of Use </a></li>
            </ul>
          </div>
        </div>
        <div class="col-xs-12 col-sm-12 col-md-12 col-lg-4">
          <div class="social">
            <a href="#" title="linkedin">
              <i class="fa fa-linkedin"></i>
            </a>
            <a href="#" title="instagram">
              <i class="    fa fa-instagram"></i>
            </a>
            <a href="https://github.com/proDev-Theron/dip-plantidentifier" title="GitHub">
              <i class="fa fa-github"></i>
            </a>
          </div>
        </div>
      </div>
    </div>
  </div>
</footer>

<a href="#" title="sroll" class="scrollToTop"><i class="fa fa-angle-up"></i></a>

<script type="text/javascript" src="js/jquery-1.12.4.min.js"></script>
<script type="text/javascript" src="js/owl.carousel.min.js"></script>
<script type="text/javascript" src="js/bootstrap.min.js"></script>
<script type="text/javascript" src="js/slick.min.js"></script>
<script type="text/javascript" src="js/vit-gallery.js"></script>
<script type="text/javascript" src="js/jquery.countTo.js"></script>
<script type="text/javascript" src="js/jquery.appear.min.js"></script>
<script type="text/javascript" src="js/isotope.pkgd.min.js"></script>
<script type="text/javascript" src="js/bootstrap-select.js"></script>
<script type="text/javascript" src="js/slick.min.js"></script>
<script type="text/javascript" src="js/jquery.littlelightbox.js"></script>

<script type="text/javascript" src="js/function.js"></script>
```

```
        <script>
         function overview() {
            document.getElementById("content").innerHTML
=
             "Mangoes are sweet, creamy fruits that have a range of possible health benefits. They are highly popular around the world. The mango is a tropical stone fruit and member of the drupe family. This is a type of plant food with a fleshy outer section that surrounds a shell, or pit. This pit contains a seed. Other members of the drupe family include olives, dates, and coconuts. There are many different kinds of mango. They vary in color, shape, flavor, and seed size. Although mango skin can be green, red, yellow, or orange, its inner flesh is mostly golden yellow. This feature is part of a collection of articles on the health benefits of popular foods. In this article, we explore the many benefits of mangoes, explain their nutritional breakdown, and provide a few healthy recipe ideas. <br> A mango is an edible stone fruit produced by the tropical tree Mangifera indica which is believed to have originated in the region between northwestern Myanmar, Bangladesh, and northeastern India. M. indica has been cultivated in South and Southeast Asia since ancient times resulting in two types of modern mango cultivars: the ‘Indian type’ and the ‘Southeast Asian type’ .Other species in the genus Mangifera also produce edible fruits that are also called ‘mangoes’, the majority of which are found in the Malesian ecoregion. <br> Worldwide, there are several hundred cultivars of mango. Depending on the cultivar, mango fruit varies in size, shape, sweetness, skin color, and flesh color which may be pale yellow, gold, green, or orange.[1] Mango is the national fruit of India, Pakistan and the Philippines, while the mango tree is the national tree of Bangladesh.";
         }
         function habitat() {
            document.getElementById("content").innerHTML
=
             "The mango is now cultivated in most frost-free tropical and warmer subtropical climates. It is cultivated extensively in South Asia, Southeast Asia, East and West Africa, the tropical and subtropical Americas, and the Caribbean. Mangoes are also grown in Andalusia, Spain (mainly in Málaga province), as its coastal subtropical climate is one of the few places in mainland Europe that permits the growth of tropical plants and fruit trees. The Canary Islands are another notable Spanish producer of the fruit. Other minor cultivators include North America (in South Florida and the California Coachella Valley), Hawai'i, and Australia. <br> Many commercial cultivars are grafted on to the cold-hardy rootstock of Gomera-1 mango cultivar, originally from Cuba. Its root system is well adapted to a coastal Mediterranean climate. Many of the 1,000+ mango cultivars are easily cultivated using grafted saplings, ranging from the 'turpentine mango' (named for its strong taste of turpentine) to the Bullock's Heart. Dwarf or semidwarf varieties serve as ornamental plants and can be grown in containers. A wide variety of diseases can afflict mangoes. <br> Mango* production – 2020 An important breakthrough in mango cultivation is the use of potassium nitrate and ethrel to induce flowering in mangoes. The discovery was made by Filipino horticulturist Ramon Barba in 1974 and was developed from the unique traditional method of inducing mango flowering using smoke in the Philippines. It allowed mango plantations to induce regular flowering and fruiting year-round. Previously, mangoes were seasonal, because they only flowered every 16 to 18 months. The method is now used in most mango-producing countries.";
         }
         function benefits() {
            document.getElementById("content").innerHTML
= "Mangos have been an important crop in India for millennia. Today, these colorful, sweet fruits are a mainstay of Indian cuisine and are popular throughout the world. Mangos can weigh anywhere from a few ounces to more than five pounds each, depending on the variety. Regardless of the type of mango you buy, these fruits offer some impressive health benefits. While mangos were historically only available at the end of the dry season, today they can be found in grocery stores all year long. <br> The vitamins, minerals, and antioxidants in mangos can provide important health benefits. For example, vitamin K helps your blood clot effectively and helps prevent anemia. It also plays an important role in helping strengthen your bones. <br> Mangos are also rich in vitamin C, which is important for forming blood vessels and healthy collagen, as well as helping you heal. <br> In addition, mangos can provide other health benefits like: <br> Lower Risk of Cancer <br> Lower Blood Pressure <br> Reduce Inflammation of the Heart";
         }
       </script>

    </body>
</html>
```

D3. INDEX PAGE HTML

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <meta name="theme-color" content="#000000">
  <!-- GOOGLE FONT -->
  <link
href="https://fonts.googleapis.com/css?family=Roboto+Condensed:300,300i,400,400i,700,700i" rel="stylesheet">
  <link
href="https://fonts.googleapis.com/css?family=Dosis:200,300,400,500,600,700" rel="stylesheet">
```

```
  <!-- CSS LIBRARY -->
  <link rel="stylesheet" type="text/css" href="css/font-awesome.min.css">
  <link rel="stylesheet" type="text/css" href="css/ionicons.min.css">
  <link rel="stylesheet" type="text/css" href="css/Pe-icon-7-stroke.min.css">
  <link rel="stylesheet" type="text/css" href="css/slick-theme.css">
  <link rel="stylesheet" type="text/css" href="css/slick.css">
  <link rel="stylesheet" type="text/css" href="css/owl.carousel.min.css">
  <link rel="stylesheet" type="text/css" href="css/bootstrap-select.min.css">
  <link rel="stylesheet" type="text/css" href="css/animation.css">
  <!-- MAIN STYLE -->
  <link rel="stylesheet" href="css/styles.css">
  <link rel="stylesheet" href="css/floradex.css">
  <!--
      manifest.json provides metadata used when your web app is added to the
      homescreen on Android. See https://developers.google.com/web/fundamentals/engage-and-retain/web-app-manifest/
    -->
  <link rel="manifest" href="%PUBLIC_URL%/manifest.json">
  <link rel="shortcut icon" href="%PUBLIC_URL%/favicon.png">
  <!--
      Notice the use of %PUBLIC_URL% in the tags above.
      It will be replaced with the URL of the `public` folder during the build.
      Only files inside the `public` folder can be referenced from the HTML.

      Unlike "/favicon.png" or "favicon.png", "%PUBLIC_URL%/favicon.png" will
      work correctly both with client-side routing and a non-root public URL.
      Learn how to configure a non-root public URL by running `npm run build`.
    -->
  <title>React App</title>
</head>

<body class="home7">
  <noscript>
    You need to enable JavaScript to run this app.
  </noscript>
  <div id="root"></div>
  <!--
      This HTML file is a template.
      If you open it directly in the browser, you will see an empty page.

      You can add webfonts, meta tags, or analytics to this file.
      The build step will place the bundled scripts into the <body> tag.

      To begin the development, run `npm start` or `yarn start`.
      To create a production bundle, use `npm run build` or `yarn build`.
    -->


  <a href="#" title="sroll" class="scrollToTop"><i class="fa fa-angle-up"></i></a>

  <script type="text/javascript" src="js/jquery-1.12.4.min.js"></script>
  <script type="text/javascript" src="js/owl.carousel.min.js"></script>
  <script type="text/javascript" src="js/bootstrap.min.js"></script>
  <script type="text/javascript" src="js/slick.min.js"></script>
  <script type="text/javascript" src="js/vit-gallery.js"></script>
  <script type="text/javascript" src="js/jquery.countTo.js"></script>
  <script type="text/javascript" src="js/jquery.appear.min.js"></script>
  <script type="text/javascript" src="js/isotope.pkgd.min.js"></script>
  <script type="text/javascript" src="js/bootstrap-select.js"></script>
  <script type="text/javascript" src="js/slick.min.js"></script>
  <script type="text/javascript" src="js/jquery.littlelightbox.js"></script>

  <script type="text/javascript" src="js/function.js"></script>
</body>

</html>
```

D3. FLORADEX CSS

```
body {
    background-image: url('../images/FloraDex/Background.png');
    background-size:auto;
    background-color: #06cb5c;
}

a {
    padding: 0%;
    margin:0%;
    border: 0%;
```

```
}
#head {
   display:inline;
   margin-right:5%;
}
#cont {
   display: inline;
}

#home {
   width:10%;
   height:10%;
}

#title {
   width:20%;
   height:20%;
   text-align: center;
}

#plantcontainer {
   display: block;
   margin-top:10%;
   margin-left:12%;

}

#plants {
   width:15%;
   height:15%;
   margin-right:2%;
}

#coppy {
   color:white;
}
```

D4. FRUITS CSS

```
body {
   background-color: #044aad;
   margin: 0%;
}

a {
   padding: 0%;
   margin:0%;
   border: 0%;
}
#head {
   padding-left:2%;
   padding-top:2%;
}

#bgimage {
   position: absolute;
   z-index: -1;
   width: 100%;
}

#home {
   width:7%;
   margin-right:30%;
}

#banner {
   width: 17%;
}

#goldy {
   width: 22%;
}

#menuses {
   border-top: 5%;
}

.tabs {
   font-family:'Franklin Gothic Medium', 'Arial Narrow',
Arial, sans-serif;
   font-size: 150%;
   color: white;
   cursor:pointer;
   margin-right: 30%;
   display: inline;
}

#domestication {
   margin:0%;
}

#title {
   width:20%;
   height:20%;
   text-align: center;
}

#bodycontainer {
   margin-top:21%;
   margin-left:12%;
   margin-bottom:2%;

}

#content {
   margin-top: 2%;
   margin-left: 7%;
   margin-right: 7%;
   margin-bottom: 2%;
   font-family:Arial, Helvetica, sans-serif;
   line-height: 300%;
   text-align: justify;
   font-size: 100%;
   color: white;
```

```
}

#coppy {
  color:white;
}
```

D5. Package CSS

```
{
  "name": "dip-plantidentifier",
  "version": "0.1.0",
  "description": "",
  "keywords": [],
  "main": "src/index.js",
  "homepage": "https://proDev-Theron.github.io/dip-
plantidentifier",
  "dependencies": {
    "@tensorflow/tfjs": "3.9.0",
    "gh-pages": "^3.2.3",
    "react": "16.5.2",
    "react-dom": "16.5.2",
    "react-magic-dropzone": "1.0.1",
    "react-scripts": "2.0.3"
  },
  "devDependencies": {},
  "scripts": {
    "start": "react-scripts start",
    "build": "react-scripts build",
    "test": "react-scripts test --env=jsdom",
    "eject": "react-scripts eject",
    "predeploy": "npm run build",
    "deploy": "gh-pages -d build"
  },
  "browserslist": [
    ">0.2%",
    "not dead",
    "not ie <= 11",
    "not op_mini all"
  ]
}
```

D6. Readme.md

~~~~
### Local Test
After exporting the tfjs model, clone this repo:
```
git clone https://github.com/proDev-Theron/dip-
plantidentifier.git
cd dip-plantidentifier
```
Install packages with npm:
```
npm install
```
Link YOLOv5 weights directory into the 'public' folder:
```
ln -s ../../yolov5/yolov5s_web_model public/web_model
```
Run the react app with:
```
npm start
```

Run
```
npm run deploy
```

PS: <del> This repo assumes the model input resolution is
640x640. </del>
If you change the `--img` value in exporting `*.pb`, change
`modelWidth` and `modelHeight` in `src/index.js`
accordingly.
~~~~

D7. index.js

```
import React from "react";
import ReactDOM from "react-dom";
import Layout from "./Layout/Layout";
import Footer from "./Layout/Footer"
import MagicDropzone from "react-magic-dropzone";

import "./styles.css";
const tf = require('@tensorflow/tfjs');

//Has 404 error in production. Added GitHub repository
name /dip-plantidentifier/
const weights = '/dip-
plantidentifier/web_model/model.json';

const names = ['Kamyas', 'Oregano', 'Mango', 'Golden
Photos', 'Croton', 'person']

class App extends React.Component {
  state = {
    model: null,
    preview: "",
    predictions: []
  };

  componentDidMount() {
    document.title = "Digital Image Processing Plant
Identifier"
    tf.loadGraphModel(weights).then(model => {
      this.setState({
        model: model
      });
    });
  }

  onDrop = (accepted, rejected, links) => {
    this.setState({ preview: accepted[0].preview || links[0] });
  };

  cropToCanvas = (image, canvas, ctx) => {
    const naturalWidth = image.naturalWidth;
    const naturalHeight = image.naturalHeight;

    // canvas.width = image.width;
```

```
    // canvas.height = image.height;

    ctx.clearRect(0, 0, ctx.canvas.width, ctx.canvas.height);
    ctx.fillStyle = "#000000";
    ctx.fillRect(0, 0, canvas.width, canvas.height);
    const ratio = Math.min(canvas.width /
image.naturalWidth, canvas.height / image.naturalHeight);
    const newWidth = Math.round(naturalWidth * ratio);
    const newHeight = Math.round(naturalHeight * ratio);
    ctx.drawImage(
      image,
      0,
      0,
      naturalWidth,
      naturalHeight,
      (canvas.width - newWidth) / 2,
      (canvas.height - newHeight) / 2,
      newWidth,
      newHeight,
    );

  };

  onImageChange = e => {
    const c = document.getElementById("canvas");
    const ctx = c.getContext("2d");
    this.cropToCanvas(e.target, c, ctx);
    let [modelWidth, modelHeight] =
this.state.model.inputs[0].shape.slice(1, 3);
    const input = tf.tidy(() => {
      return tf.image.resizeBilinear(tf.browser.fromPixels(c),
[modelWidth, modelHeight])
        .div(255.0).expandDims(0);
    });
    this.state.model.executeAsync(input).then(res => {
      // Font options.
      const font = "16px sans-serif";
      ctx.font = font;
      ctx.textBaseline = "top";

      const [boxes, scores, classes, valid_detections] = res;
      const boxes_data = boxes.dataSync();
      const scores_data = scores.dataSync();
      const classes_data = classes.dataSync();
      const valid_detections_data =
valid_detections.dataSync()[0];

      tf.dispose(res)

      var i;
      for (i = 0; i < valid_detections_data; ++i){
        let [x1, y1, x2, y2] = boxes_data.slice(i * 4, (i + 1) * 4);
        x1 *= c.width;
        x2 *= c.width;
        y1 *= c.height;
        y2 *= c.height;
        const width = x2 - x1;
        const height = y2 - y1;
        const klass = names[classes_data[i]];
        const score = scores_data[i].toFixed(2);

        // Draw the bounding box.
        ctx.strokeStyle = "#00FFFF";
        ctx.lineWidth = 4;
        ctx.strokeRect(x1, y1, width, height);

        // Draw the label background.
        ctx.fillStyle = "#00FFFF";
        const textWidth = ctx.measureText(klass + ":" +
score).width;
        const textHeight = parseInt(font, 10); // base 10
        ctx.fillRect(x1, y1, textWidth + 4, textHeight + 4);

      }
      for (i = 0; i < valid_detections_data; ++i){
        let [x1, y1, , ] = boxes_data.slice(i * 4, (i + 1) * 4);
        x1 *= c.width;
        y1 *= c.height;
        const klass = names[classes_data[i]];
        const score = scores_data[i].toFixed(2);

        // Draw the text last to ensure it's on top.
        ctx.fillStyle = "#000000";
        ctx.fillText(klass + ":" + score, x1, y1);

      }
    });
  };

  render() {
    return (
      <>
      <Layout/>
      <div className="Dropzone-page">
        {this.state.model ? (
          <MagicDropzone
            className="Dropzone"
            accept="image/jpeg, image/png, .jpg, .jpeg, .png"
            multiple={false}
            onDrop={this.onDrop}
          >
            {this.state.preview ? (
              <img
                alt="upload preview"
                onLoad={this.onImageChange}
                className="Dropzone-img"
                src={this.state.preview}
              />
            ) : (
              "Choose or drop a file."
            )}
            <canvas id="canvas" width="640" height="640" />
          </MagicDropzone>
        ) : (
          <div className="Dropzone">Loading model...</div>
        )}
      </div>

      <Footer />
      </>
    );
  }
}
```

```
const rootElement = document.getElementById("root");        ReactDOM.render(<App />, rootElement);
```