

Development and Testing of Embedded System for Smart Detection and Recognition of Witches' Broom Disease on Cassava Plants using Enhanced Viola-Jones and Template Matching Algorithm

Irma T. Plata¹, Allan C. Taracatac², Edward B. Panganiban³

¹Isabela State University, Philippines, irma.t.plata@isu.edu.ph

²Isabela State University, Philippines, allantaracatac@gmail.com

³Isabela State University, Philippines, edward.b.panganiban@isu.edu.ph



ABSTRACT

The Cassava Phytoplasma Disease (CPD) also known as Witches' Broom disease is presently manifesting in some cassava fields. The infection is named after the broom-like leaf spread at the top of cassava plants. This study centered on the development and testing of an embedded system for smart detection and recognition of Witches' Broom disease in natural environment with applied multiple layer validation procedures. The present technology in image processing applying enhanced Viola-Jones and template matching algorithm made the study possible.

The embedded system was developed using Python version 3, OpenCV version 3.2.0, and Raspbian Jessie OS. The hardware includes Raspberry Pi 3 Model B and, Raspberry Pi camera module v2 with 8-megapixel camera resolution. A custom cascade classifier model comprising of image acquisition, template selection, train cascade classifier, and real-time object detection and recognition was implemented. With the implementation of two custom trained cascade classifiers, the embedded system was able to lessen the false detection rate compared to the analysis based on general features of a CPD infected cassava leaves alone, thus, increasing its detection accuracy. During the template matching process, 2 voting schemes were used to compare key features from template versus the image of object of interest and used to evaluate the overall results with the 11 templates from the score gauge, thus, these schemes were able to recognize if cassava is CPD infected or not. Currently, the downside of the study is that with the implementation of multiple layers of validation in an attempt to minimize false detection puts too much toll in the micro processing unit which is critical in attempting the detection and recognition in real-time (or close to real-time). Although detection is real time, recognition has a delay in processing the feature of the cassava whether infected or not.

Key words : Cassava, Image processing, Template matching algorithm, Viola-Jones algorithm, Witches' broom.

1. INTRODUCTION

Cassava (*Manihot esculenta Crantz*) is locally known as "kamoteng-kahoy" or "balinghoy" in the Philippines. Cassava is a perennial shrub that sometimes reaches the size of a small tree, as high as 1 meter with a pale to dark-green leaves fan-shaped, with 5 to 9 lobes. Its stems vary in color from pale to dirty white to brown, marked by numerous nodes formed by scars left by fallen leaves [1]. In addition to rice and corn, cassava is one of the vital agricultural crops in the Philippines. It can be used as human food, ingredients for feeds, and numerous industrial uses that include starch, flour, and bio-ethanol.

The Philippine Statistics Authority (PSA) reported that Region 02 (Cagayan Valley) cassava volume of production between year 2013 to 2017 reached 77,723 metric tons (MT) in 2013 and 325,265 MT in 2017. The huge increase in volume can be attributed to areas harvested from 4,895 hectares in 2013 to 22,069 in 2017 [2]. Production growth came from the significant increase in areas harvested due to high demand and marketing particularly in Isabela, Cagayan Province, Quirino, Bukidnon, and Misamis Oriental. Cassava is planted each year in about 120,000 hectares of agricultural lands producing about 1.8 million tons of cassava [3], [4].

At present, a type of virus called Cassava Phytoplasma Disease (CPD) also known as Witches' Broom disease is manifesting in some cassava fields, which is a serious threat at present to the cassava industry. The disease is named after the broom-like leaf proliferation affecting the top of cassava plants. CPD infection shows yellowing or reddening of leaves, shrinking of the internode [5] resulting in stunted growth, excessive multiplication of shoots, and production of smaller leaves, among other symptoms, until the plant eventually dies. This disease reduces cassava root starch content, affecting yield value of the crop and farmers' income [5].

Phytoplasma interferes in plant development. Accordingly, if the manifestations or the symptoms appear as early as two to three months, there will be 100% loss as there will be no more

production of roots. If symptoms appear at later stages like five to six months or onward, there will be significant loss in yield of up to 40% [6]. In the study conducted by the Philippine Root Crop Research in 2015, CPD threatens cassava supply amidst its high demand for food, feeds, and industrial uses. Worse, it is also feared to decrease export volume to as much as 30 percent to the detriment of the country's 1.5-billion-dollar cassava industry [5]. This disease has already swept through Vietnam, Thailand, Cambodia, Laos, China, and the Philippines, resulting in dramatic reductions in cassava root starch content and in parts of Vietnam, a massive 80 percent yield losses [7].

The plant disease diagnosis is limited to human visual capabilities because most of the early symptoms are microscopic [8]. Technically, image processing [9] is the best technique that can be use in agricultural applications since it predicts plant disease from images of plants [8].

Strengthening cassava pest and disease response systems, as an approach is increasingly recognized as a key priority. According to Hoat, a strategy to address the CPD includes tools and techniques to improve their early detection [7]. Toward that direction, this study is fixated on the development and testing of an embedded system design to detect and recognize CPD through image processing by cutting the time in the detection process of plant disease symptoms by only analyzing those potential object of interest and set criteria to increase accuracy rate.

2. METHODOLOGY

2.1 Conceptual Framework

2.1.1 The Viola-Jones Object Detection Framework

Viola-Jones object detection framework is considered as the first object detection framework to provide competitive object detection rates in real-time. It has several characteristics which make it a good detection algorithm which includes: (1) robust, high detection rate (true-positive rate) and low false-positive rate always; (2) real-time, for practical applications with at least 2 frames per second must be processed; and (3) face detection only (not recognition), the goal is to distinguish faces from non-faces (detection is the first step in the recognition process) [10]. The algorithm has four stages: Haar feature selection; creating an integral image; AdaBoost training; and cascading classifiers [11].

2.1.2 The Template Matching Algorithm

Template Matching algorithm is the classification of unknown samples by comparing them to known prototypes or templates using digitized images [12]. The difficulty of this algorithm is the matching of identical features and its processing will cause a big toll on the processor if this will be implemented frame by frame from a live camera feed which has several hundreds of frames per minute [12].

2.1.3 The Enhanced Viola-Jones and Template Matching Algorithm

The detection and recognition of CPD using image processing was made possible through the enhanced Viola-Jones and template matching algorithm by filling in the feature-gap of both algorithms. As shown in Figure 1, the block diagram illustrates how the custom cascade classifier model was prepared adapted from Viola-Jones object detection framework and collection of least common visual appearances that will be used in template matching based from template matching algorithm for CPD infected cassava that intends to work in parallel [13]. The process involves acquisition of images and template selection, then train cascade classifier, and the real-time object detection and post-processing for recognition.

The enhanced algorithm cuts the time in the detection process by only analyzing those potential objects of interest and set criteria to increase recognition accuracy rate. The Viola-Jones is responsible for locating potential objects of interest using its detection algorithm's robust real-time processing. On the other hand, the template matching algorithm is responsible for object recognition by comparing the acquired image (*potential CPD infected cassava*) with CPD infected cassava image template that was carefully hand-picked from the least common visual appearance of severe CPD infected cassava.

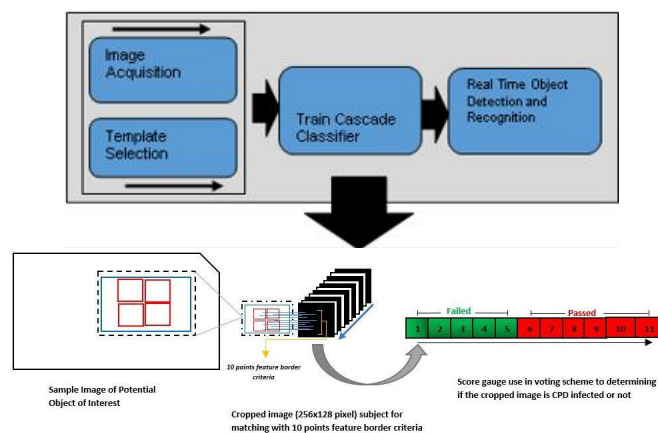


Figure 1: Cascade Classifier Model based on Enhanced Viola-Jones and Template Matching Algorithm

2.2 Pre-Implementation

Initially, the researchers visited an infected plantation located at Santos, San Agustin, Isabela with the assistance of an agricultural technologist in the locality. Present method used to distinguish and collect data on infected cassava plantations via visual referencing (*naked eye*) was employed. Only KU-50, Lakan 1, and Golden Yellow cassava variety were considered as subject of study focusing on leaf features including textures and edges which are consistent on most CPD infected cassavas. Samples collected were carefully examined and analyzed by taking photos of those severe CPD infected cassava and randomly hand-picked twenty (20) stalk

with leaf samples for further analysis. Testing was done in a controlled-environment.

2.3 Implementation

In the study implementation, the embedded system was developed and tested based on the custom cascade classifier model based on enhanced Viola-Jones and Template Matching Algorithm.

2.3.1 Development of the Embedded System

The actual diagram of the embedded system is shown in Figure 2. It includes the Raspberry Pi 3 Model B, a microcontroller [14], that runs the python scripts for the detection and recognition of CPD infected cassavas. The enclosed Raspberry Pi 3 Model B together with the Raspberry Pi camera module v2 with an 8-megapixel camera resolution, HDMI connector that delivers crisp high definition display, 3.5” touchscreen display, a power bank, micro USB power cable, and 16 Giga Byte micro SD card. The total weight of the system is around 700 g. The software used in the development include Raspbian Jessie OS, OpenCV ver 3.2.0; and Python version 3.

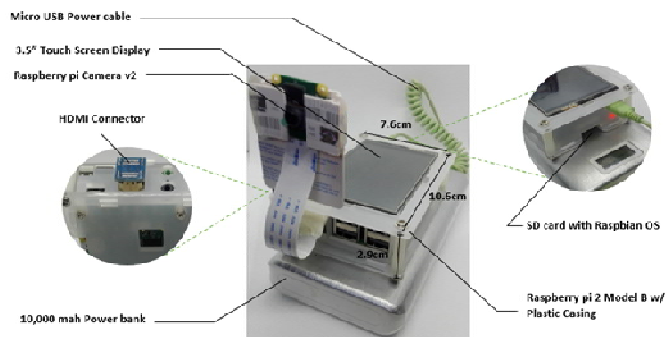


Figure 2: The Hardware Components of the Embedded System

2.3.2 Feature matching + Homography

In addition to template matching procedure, an OpenCV process called “Feature Matching plus Homography” was introduced to find the location of some parts of an object (*query image*) in another cluttered image (*train image*). This information is used to locate the object on the train image. With this technique, `cv2.findHomography()` a function from `calib3d` module, was used. Furthermore, passing the set of points from both images, it will find the perspective transformation of that object. Then using `cv2.perspectiveTransform()` function from OpenCV to find the object. It needs at least four (4) correct points to find the transformation. There can be some possible errors while matching by just using this procedure which may affect the result. To solve this problem, the algorithm uses RANSAC or LEAST_MEDIAN (*which can be decided by the flags*). In this section a condition was set that at least 10 matches (defined by `MIN_MATCH_COUNT`) are to be there to find the object. Good matches which provide correct estimation are called inliers and the remaining are called outliers.

`Cv2.findHomography()` returns a mask which specifies the inlier and outlier points [15].

2.3.3 System Process Flow

The system process flow shown in Figure 3. Using its embedded camera, the system scans real-time footage which undergoes image classification frame by frame using the trained cascade classifiers. If it hits potential CPD infected cassava the system is triggered to record a short video clip (2-3 seconds); otherwise, it will continue to analyze each frame of the real-time camera feed. The recorded video clip is converted into images each frame, and each converted image undergoes object-of-interest mapping. The biggest object of interest from the image is cropped-out and then processed by the template matching procedure using the pre-defined templates. Matched features score is weighted and serve as the deciding mechanism for recognition. If the crop image scores high enough in the score gauge, the original image where the crop image came from is retained; otherwise, it will be disregarded and the next image in line is processed.

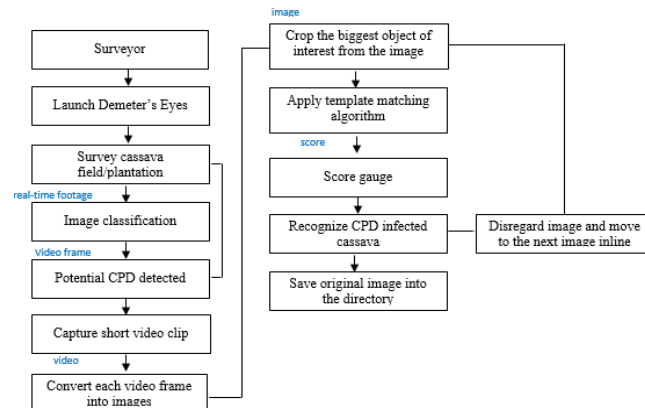


Figure 3: System Process Flow

2.3.4 Implementation of the Custom Cascade Classifier Model

2.3.4.1 Image Acquisition

Image acquisition involves collecting image samples of both CPD infected cassavas and non-infected (*and other background images*). As shown in Figure 4, the acquisition of positive images consists of CPD infected cassavas. At least 20 stalks (*with leaves*) were randomly collected then placed into a pot (*just to hold the stalk upright*) taken with an approximate distance of within 1 to 2 meters from different angles with changing canvas as background.



Figure 4: Sample Positive Images (CPD infected cassava leaves - overhead view)

On the other hand, negative images include images that do not contain the object of interest as shown in Figure 5. Negative images were taken from the actual background where it will soon be implemented so that the classifier can easily adapt with the environment and minimize false positive detection.



Figure 5: Sample of Negative Images (*non-infected cassava leaves and background*)

2.3.4.2 Template Selection

Template selection is looking for the 11 least common visual appearance of CPD infected cassava among all the positive images. The Raspberry Pi camera v2 was used in taking photos since template matching is sensitive enough when it comes to color and texture.

2.3.4.3 Train Cascade Classifier

The train cascade classifier is the process of training the cascade classifier to distinguish CPD infected cassavas and disregard those are not based on HAAR features from positive images versus negative images acquired.

2.3.4.4 Real-time Object Detection and Recognition

The real-time object detection and post-processing recognition is where the trained classifier is implemented in the detection of object of interest and taking photos. This activity signifies Viola-Jones object detection framework working in parallel with Template Matching algorithm to detect and recognize CPD infected cassavas.

2.3.5 Experimentation

2.3.5.1 Sample Images of Potential Object of Interest using HAAR

The implementation of template matching algorithm compares features of a template image into a complex image. Taking photos/images of object of interest is the initial step. The captured photo/image features were matched with the 11 templates identified in the template selection. If the image taken has enough matched features with the templates, then it will be retained; otherwise it will be disregarded.

The first custom trained cascade classifier will detect object of interest based from general features of a CPD infected cassava (*this includes the leaf formation as a whole*). Once an object of interest is spotted a bounded box will mark the target, represented by blue bounded box as shown in Figure 6a. Then the second custom trained cascade classifier will verify it by scanning each leaf and if it hits positive a bounded box will mark each leaf as shown in Figure 6b represented by the red

bounded box. This will then trigger the camera to record a short video clip with at least 2 to 3 seconds duration and convert the video clip frames into images of the object of interest.

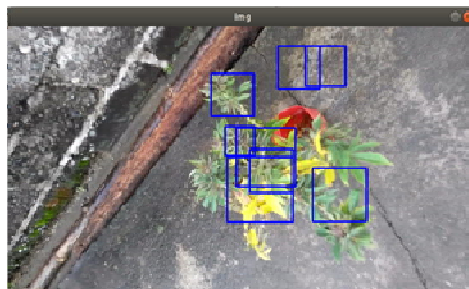


Figure 6a: Applying the First Custom Trained Cascade Classifier to Detect Potential Object of Interest as Test Image



Figure 6b: Applying the Second Custom Trained Cascade Classifier to Verify Potential Object of Interest as Test Image

2.3.5.2 Cropped Image Subject for Matching with 10-points Feature Border Criteria

After capturing the image and saving it on the potential CPD infected cassava directory, the template matching procedure will start using the same custom trained cascade classifier (for general features) from the detection process to locate the object of interest on the input image. Then all detected object of interest are cropped from the input image but only retains the biggest one (this is to make sure that the template matching process has bigger view of the object and higher chance to recognize the target object). Then place the cropped image on the matching procedure with the predefined 11 templates of the least common visual appearance of a CPD infected cassava. The 11 predefined templates are cropped in roughly 150 x 150 pixels. The cropped image from the potential CPD infected cassava will then compare with the 11 templates using 10 points feature border criteria in filtering key features from the cropped image as shown in Figure 7.

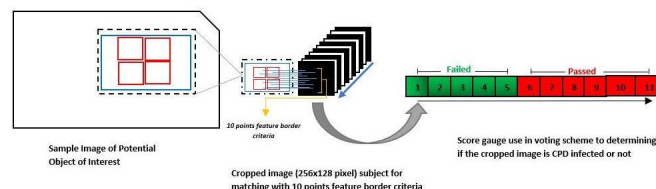


Figure 7: Cropped Image Subject for Matching with 10 Points Feature Border Criteria

If enough key features are found from the cropped image and reached 10 points border criteria, this will mark on the score gauge. The same argument will loop to other templates and whenever it reaches the 10 points border criteria, the counter on the score gauge will increment by 1 until it reaches the last template. After the matching process with 11 templates, the total score will then be evaluated. If the counter in the score gauge is 5 and below then this marks the image as failed (*negative CPD infected cassava*) if the score is 6 and up then this means that the potential image is now recognized as a positive CPD infected cassava and saved on the positive CPD directory.

In the first attempt of training a custom cascade classifier, the researchers gathered 2,325 samples/images of non-CPD infected cassava (*including other images*) with different backgrounds using different camera resolution and 4,046 samples/images of cropped CPD infected cassava taken from different backgrounds and angle which manifest severe CPD infection using a 5-megapixel camera.

The HAAR cascade training process completed 20 stages of training. The training uses Gentle AdaBoost with minimal hit rate of 0.995, maximal false alarm rate of 0.500, maximal trim rate of 0.950, maximal depth weak tree of 1.000 and, maximal weak trees of 100. These settings can be changed depending on the desired depth of training. The deeper the training, the more accurate the cascade is. However, it is still dependent on the sample datasets used in training.

2.4 Testing of the embedded system in terms of its accuracy of detection and recognition

The detection and recognition accuracy of the embedded system was tested based on detection capability and the recognition capability. To evaluate, dozens of sample images (*positive and negative*) and pre-recorded video clips are fed into the system to test the accuracy of the two cascade classifiers together with the template matching procedure. Final result of the test are calculated using a simple mathematical formula adapted from the work of Wen Zhu [16] wherein:

$$Accuracy = \frac{(TN + TP)}{(TN + TP + FN + FP)} = \frac{(Number\ of\ Correct\ Assessments)}{(Number\ of\ All\ Assessments)} \quad (1)$$

where: TP – True Positive, TN – True Negative, FN – False Negative, FP – False Positive

3 RESULTS AND DISCUSSION

3.1 Pre-test to determine speed and accuracy

During the development (programming) of the second python script, a pre-test was performed to determine its speed and accuracy in the template matching process across different image resolutions.

There were 2 sets of test images. First is the positive list in high definition resolution of 3240 x 2464 taken from an 8-megapixel camera and second, the same positive list but resized into 256 x 128 pixels which contain one (1) CPD infected cassava. Template matching analysis using the image original resolution taken from an 8-megapixel camera took around 29 to 33 seconds per template in 1 image and a total of around 5 to 6 minutes for the whole cycle (comparing all 11 templates into 1 image). On the resized resolution of 256 x 128 pixels, it only took 5 to 8 seconds per template in 1 image and total of around 55 seconds to 1 minute and 27 seconds for the whole cycle. Other image resolution lower than 256 x 128 showed inconsistency in the results of succeeding tests.

With the test results, few things are notable, such as, several images are disregarded which supposedly should not have been. Reasons for this could be any of the following: 1) the 10 points border criteria were too high; 2) the distance where the target object was taken may be too far; or (3) there were not sufficient recognizable feature in the subject.

3.2 Experimental Results

In the template matching, eleven (11) image samples were carefully hand-picked from severe stages of a CPD infected cassava and these samples served as the template image that will be compared to those images acquired from the surveying unit wherein those images are suspected to have the object of interest.

The 11-template implementation was based on the positive images taken in the training of the cascade classifier. These are the least common visual appearances of a CPD infected cassava. Template images are taken using a 5-megapixel and an 8-megapixel camera. Images of potential CPD infected cassava that are saved in the potential CPD directory are then subjected to the template matching procedure based on key features existing on the 11 templates. The potential CPD image was scanned using the general classifier for leaf formation. The same classifier is used in the detection process, and this is to locate the object of interest in the image.

After locating all object of interest, the images will then be cropped but only the biggest one is retained (this is to make sure that the template matching process has bigger view of the object and higher chance to recognize the target object) and place the cropped image on the matching procedure with the predefined 11 templates. Using the templates, the cross examination starts and each image is analyzed. Each template will be compared on the acquired image. First is finding the key points and descriptors with Scale-Invariant Feature Transform (SIFT)[17] features in the image and store all the good matches as per Lowe's ratio test. Set a condition that at least 10 matches that are to be there to considered as a positive hit. At the end of image analysis per template, the system will give a result of either passed or failed; meaning that all 11 template images will be compared to each acquired images

from the surveying unit and each template comparison will give an output if it passed or failed. Each acquired image will have a total of 11 results (coming from 11 templates) and the majority of the results will be the final conclusion. Figure 8 are sample template images while Figure 9 are acquired images.

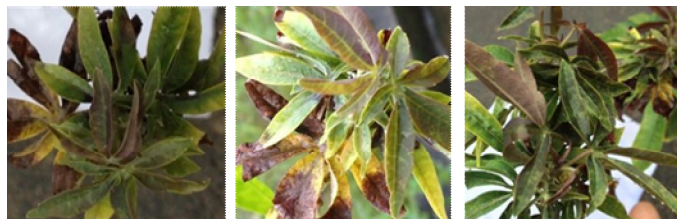


Figure 8: Sample Template Images



Figure 9: Acquired Images

Finally, the Template Matching Algorithm found enough matches on the images taken, marking a positive hit as seen on Figure 10. However, on Figure 11, the Template Matching Algorithm found “no match” on the image taken using the prepared template, where in fact the image taken is positively identified as infected cassava via naked eyes. This is the reason why there should be at least 11 template image of infected cassava taken from severe CPD infection at different angles for broader view on the subject to minimize false negative results.

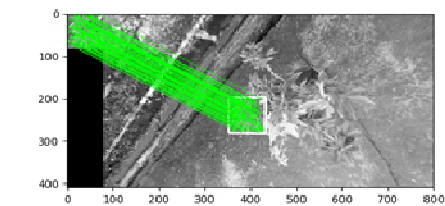


Figure 10: Test Image using Template Matching Algorithm makes it as a positive hit (Matched)

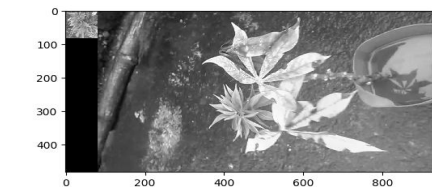


Figure 11: Test Image using Template Matching Algorithm makes it as a negative hit (mismatched)

3.3 Testing in Terms of Accuracy Detection

To generate a simplified test log, we choose 2 extracted frame from the video clip as shown in Figure 12 and a python script (part of the processing unit) to test the accuracy detection of all trained cascade classifiers for general features with the same parameters for the detection scale, the number of test images and, minimum neighbors. The total number of true negative (TN) objects that can be seen via naked eye on both images is 10, based from manual counting. This includes grasses and other plants present.



Figure 12: Actual Image Used in the Accuracy Test

Based on Table 1, the first and third attempt were not able to detect any object of interest using the specified detection scale and shows multiple false detection, yielding 50 percent accuracy rate. Although the second attempt was able to detect one (1) true positive object (TP) on the test images, yet still yields 50 percent accuracy rate due to the increase of false positive (FP) detection.

Table 1: General Classifier – Accuracy Test Log using 80 x 80 pixels’ detection scale

GENERAL CLASSIFIER – ACCURACY TEST								
Atte mpt (s)	Min Detection Scale (pixels)	No of Test Image	Min Neig hbor	TP	TN	FP	FN	Accura cy
1	80 x 80	2	5	0	7	3	4	50%
2	80 x 80	2	5	1	6	4	3	50%
3	80 x 80	2	5	0	7	3	4	50%

Where: TP – means correct detection of an object of interest
 TN – means correctly not detecting something that is not the object of interest
 FP – means incorrect detecting something else as the object of interest
 FN – means missing object of interest

In Table 2, the first and second attempt in training a custom cascade classifier shows no improvement in terms of detecting true positive objects. The accuracy rate of the first and second attempt increased by 21 percent due to zero (0) detections in both true positive (TP) negative (TP) and false positive (FP).

In the third attempt, this is where the adjustments made on the training samples paid off. The accuracy rate of the trained cascade classifier rose to 79 percent; this is because of the increase in detecting true positive (TP) objects and lesser false negative (FN) detection.

Table 2: General Classifier – Accuracy Test Log using 200 x 200 pixels’ detection scale

GENERAL CLASSIFIER – ACCURACY TEST								
Atte mpt (s)	Min Detection Scale (pixels)	No of Test Image	Min Neig hbors	TP	TN	FP	FN	Accura cy
1	200 x 200	2	5	0	10	0	4	71%
2	200 x 200	2	5	0	10	0	4	71%
3	200 x 200	2	5	2	9	1	2	79%

There is a remarkable improvement in training the custom cascade classifier for CPD infected leaf detection that was used to cross examine the detected object of the general feature cascade classifier. This is an attempt to lessen the false detection. Instead of simply training the classifier with sets of leaf formation and general features, the classifier was trained to analyze each leaf using the same parameters when training with the general cascade classifier but only focuses on the 2 major leaf appearance of a CPD infected cassava which is the yellowing or reddening of leaves.

As illustrated on Figure 13, the detected potential objects of interest are enclosed with thin red lines (*the red bounded box seen on the test result are only used for testing purposes, in actual implementation the blue bounded box signify the general feature(s) found and the red bounded box signifies the individual leaf detection*). Also, note that the classifier used in Figure 13 can be improved by acquiring lots of sets of good samples and by carefully preparing the positive images.

As observed with the custom trained cascade classifiers, even if there are numbers of false detection using the trained classifiers, there is still a second algorithm working in parallel with the first algorithm that will validate the images taken if it will be accepted as positive hit or not, using the template matching algorithm.



Figure 13: Actual Test Result of the Third attempt in Training Cascade Classifier

3.4 Evaluation of the Embedded System in terms of its Accuracy and Speed of Detection and Recognition

During the training stage of the general cascade classifier for CPD infected cassavas, there were three attempts made as shown in Table 3: using the same training parameters in a number of training stages, gentle AdaBoost with minimal hit rate, maximal false alarm rate, maximal trim rate, maximal depth weak tree and maximal weak trees.

Table 3: General Classifier Training Complete Log

Attempt (s)	Training Sample Image		Camera Type use to take sample images	Remarks	GENERAL CLASSIFIER- TRAINING LOG				
	Positive Image Count	Negative Image Count			No of Training Stages	Gentle AdaBoost w/ Min Hit Rate	Maximal False Alarm Rate	Maximal Trim Rate	Maximal Depth Weak Tree
1	4,046	2,325	5 megapixels	Leave too much background frame on Pos. images	20	0.9950	0.5000	0.9500	1,0000
2	2,174	2,000	5.8 megapixels	Pos. images has repeated background frame	20	0.9950	0.5000	0.9500	1,0000
3	822	531	5.8 megapixels	Reduced Pos. image background frame and avoid constant background	20	0.9950	0.5000	0.9500	1,0000

As illustrated in Table 4 and Figure 14, there is a dramatic reduction of false alarm rate (FA) as training progresses. As observed in the log results across all attempts in training the custom cascade classifier, it is noticeable that in the first attempt the FA result pattern from stage 1 up to stage 20 is low-low-high-low-low. The high remark is when the FA reached 0.997419 from 0.498495 previously. We do not know for sure what is the exact reason behind this, but the end product (*the cascade classifier produced in the first attempt*) shows low detection accuracy rate since it was not able to detect the object of interest using 80 x 80 pixels and 200 x 200 pixels’ detection scale.

The second attempt maintains FA result pattern from stage 1 to 20 of low-low-low-low-low, since it did not go up higher than 0.4. But the end result is almost the same as the first attempt, the only difference is that it was able to identify one correct true positive (TP) with 80 x 80 pixels’ detection compared to the first attempt which was not able to detect any of the objects of interest across the set detection scale during the test.

In the third attempt, as the second attempt, the FA result pattern of low-low-low-low-low was maintained but lower in terms of numbers. With this result pattern, the end product shows remarkable improvement from the first two attempts. Based on the test, it was able to detect two (2) true positive (TP) object using the detection scale of 200 x 200 pixels.

Table 4: General Classifier Training Log – Summary (per stage)

GENERAL CLASSIFIER- SUMMARY OF TRAINING LOG					
A	1 st Stage	5 th Stage	10 th Stage	15 th Stage	20 th Stage
1	0.458065	0.498495	0.997419	0.490323	0.48086
2	0.48	0.488	0.4865	0.4685	0.498
3	0.478343	0.403013	0.442561	0.463277	0.489642

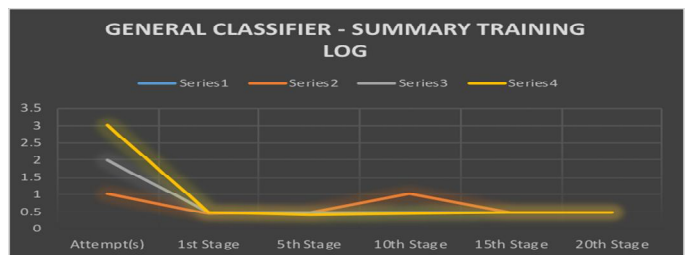


Figure 14: Graphical General Classifier Training Log – Summary (per stage)

In the actual detection procedure where the two custom trained cascade classifiers were implemented, resolutions of proximately 30 frames per second at 640 x 480 resolution with raspberry pi 3 model B and raspberry pi camera v2 were reached. This marks a good performance and near to real-time detection. This output can still be enhanced using threading to achieve higher frame rates. Considering that Raspberry Pi 3 Model B has only 1 gigabyte of RAM and 1.2 GHz CPU, running the camera module together with the template matching script pushes the system unit to its limit.

4 CONCLUSION

The embedded system can detect potential CPD infected cassava in natural environment with the implementation of multiple layer validation procedure to recognize if a cassava manifests visual symptom of CPD infection. The present technology in image processing applying enhanced Viola-Jones and template matching algorithm made the study possible.

With the implementation of two custom trained cascade classifiers, the researchers were able to refine the detection capabilities of the embedded system from the way of seeing it as one object (in general) with the help of the first custom trained cascade classifier that was specifically trained to detect potential CPD infected cassava as a whole (this includes the leaf formation and features in general) and narrowing down the detection into its leafs with the use of the second custom trained cascade classifier that is specifically trained to analyze each leaf. By doing so, the system was able to lessen the false detection rate compared to the analysis based on general features of a CPD infected cassava leaves alone. With better data sets for training the custom cascade classifiers, the detection accuracy can be greatly increased.

Also, in the template matching procedure, the use of two (2) voting scheme; (1) compare key features from the template versus the image of object of interest (2) evaluate the overall result with the 11 templates from the score gauge. These schemes were able to recognize if a subject cassava plant is CPD infected or not.

Currently, the embedded system was able to detect and recognize CPD infected cassava but the downside is that with the implementation of multiple layers of validation in an attempt to minimize false detection, the capability of the micro-processing unit is pushed to its limits. Nevertheless, the embedded system was able to achieve CPD detection in almost real-time (the detection script runs on live camera feed with more or less 35ms latency only) by lowering the camera resolution to 640 x 480. When recording video clips for image extraction, the camera stays at high resolution then goes back to 640 x 480 resolution after extracting frames for template matching.

Finally, the results obtained from experimentation showed that this study will have high potentials in reducing disastrous effect on cassava plantation caused by CPD infestation. Through immediate identification of affected cassava plants from a large plantation, infected plants can be isolated and treated in a timely manner.

REFERENCES

- [1] K. Nordin, N. Hamzah, M. A. Jamaluddin, and S. A. Bahari, "Physical properties of cassava (*Manihot esculenta*) stem," vol. 4, no. 3, pp. 115–118, 2016.
- [2] Philippine Statistics Authority. "Crops Statistics of the Philippines 2013-2017," ISBN: 2012-0478. 2018. <https://psa.gov.ph/content/crops-statistics-philippines-national-and-regional>
- [3] J. L. Bacusmo, "Status and potentials of the Philippines cassava industry," Cassava's Potential Asia 21st Century Present Situat. Futur. Res. Dev. Needs. Proc. 6th Reg. Work. held Ho Chi Minh City, Vietnam. Feb, pp. 21–25, 2001.
- [4] R.H. Howeler; S.L. Tan, "Cassava's potential in Asia in the 21st Century: Present situation and future research and development needs," Proc. Sixth Reg. Work. held Ho Chi Minh City, Vietnam, pp. 314–332, 2000.
- [5] R. Argana, "Addressing the threats of Cassava Phytoplasma Disease," DOST-PCAARRD S T Media Serv., pp. 1–2, 2016. <http://www.pcaarrd.dost.gov.ph/home/portal/index.php/quick-information-dispatch/2761-addressing-the-threats-of-cassava-phytoplasma-disease>
- [6] R. Dela Cruz, "Fighting cassava witches' broom disease," Philipp. Star Glob., p. 1, 2016.
- [7] T. X. Hoat, "New agriculturist: Witches' broom - a curse on cassava," pp. 1–2, 2014.
- [8] R. C. R. K. P. P. S. Desai, "Detection and measurement of paddy leaf disease symptoms using image processing," Int. J. Eng. Sci. Comput., vol. 7, no. 5, pp. 11964–11967, 2017.
- [9] E. Panganiban, A. Paglinawan, W. Chung, and G.L. Paa, "A Novel Technique in in Classifying Heart Diseases based on Electrocardiogram (ECG) Signals using Deep Learning and Spectrogram Image Analysis", International Journal of Advanced Trends in Computer Science and Engineering, vol. 8, no. 4, pp. 1734-1740, 2019. <https://doi.org/10.30534/ijatcse/2019/102842019>
- [10] Q. Chen, K. Kotani, and F. Lee, "Face recognition using multiple histogram features in spatial and frequency domains," 2016 12th Int. Conf. Signal-Image Technol. Internet-Based Syst., pp. 204–208, 2016. <https://doi.org/10.1109/SITIS.2016.40>

- [11] P. Viola and M. M. J. Jones, “**Robust real-time face detection,**” *Int. J. Comput. Vis.*, vol. 57, no. 2, pp. 137–154, 2004.
<https://doi.org/10.1023/B:VISI.0000013087.49260.fb>
- [12] G. Cox, “**Template matching and measures of match in image processing,**” *Dep. Comput. Sci. Univ. Cape*, 1995.
- [13] C. P. Papageorgiou, M. Oren, and T. Poggio, “**A general framework for object detection,**” *Sixth Int. Conf. Comput. Vis. (IEEE Cat. No.98CH36271)*, pp. 555–562, 2018.
- [14] E. Panganiban, “**Automated Hazardous Gas Detecting Robot using Wireless Sensor Networks with GSM-SMS Alert and Fire Control System for Households**”, *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 8, no. 3, pp. 804-809, 2019.
<https://doi.org/10.30534/ijatcse/2019/72832019>
- [15] A. Mordvintsev and K. Abid, “**OpenCV-Python Tutorials Documentation,**” 2017.
- [16] W. Zhu, N. Zeng, and N. Wang, “**Sensitivity, specificity, accuracy, associated confidence interval and ROC analysis with practical SAS® implementations.,**” *Northeast SAS Users Gr. 2010 Heal. Care Life Sci.*, pp. 1–9, 2010.
- [17] Yakkali S, Nara V, Tikone N and Ingle D “**Robust object detection and tracking using SIFT algorithm,**” *Int. J. Adv. Res. Comput. Sci. Softw. Eng.* **5** pp. 683–8, 2015.