



Machine Vision Recognition System for Iceberg Lettuce Health Condition on Raspberry Pi 4b: A Mobile Net SSD v2 Inference Approach

Alvin Sarraga Alon¹, Jesusimo L. Dioses, Jr.²

¹Technological Institute of the Philippines, Manila Philippines, aalon.cpe@tip.edu.ph

²Isabela State University, Echague Isabela Philippines, jdiosesjr@gmail.com

ABSTRACT

Lettuce provides vitamin C, calcium, potassium, and folate. Within iceberg lettuce, the nutrients will help you fulfill the normal daily requirements for many vitamins and minerals. It is most commonly cultivated as a vegetable leaf, but sometimes for its stem and seeds. Lettuce is most widely used for salads, but it can also be used in other foods, such as soups, sandwiches, and wraps; it can be grilled too. Many farmers produce lettuces on the farm. Producing lettuces isn't that an easy task it requires manpower and hard work. People who buy lettuce don't have the skill to determine if the lettuce is healthy or have a disease, they just based only on the color of the lettuce. The study developed a system project that focuses on lettuce health recognition. The system determines if the lettuce is healthy or disease. It is based on machine vision using deep learning, it is connected to a microcontroller raspberry pi 4b. Lettuce health recognition has been done with an overall testing accuracy of 97.59%.

Key words: Lettuce health recognition, machine vision, object detection, raspberry pi.

1. INTRODUCTION

Plant disorder, failure of the normal state of a plant that affects or changes its essential functions [1]. Lettuce plants experience disease, while each species is prone to characteristic diseases, these are relatively few, in each case [2]. The incidence and prevalence of plant diseases vary from season to season, depending on the nature of the pathogen, the conditions of the climate and the crops and varieties are grown [3]. Many varieties of plants are particularly prone to disease outbreaks while others are more immune to them [4].

Plant diseases and fungal attacks result in crop losses of up to 30 percent per year[5]. That's why the detection of plant diseases is very important. Although there is an existing way of detecting plant diseases, it is very expensive and time-consuming [6]. Local farmers need to be aware and informed about the different ways and techniques for handling and managing their crops [7]. Precise identification and diagnosis are very important for food security and the prevention of invasive pests and diseases [8].

There are several studies used image processing to recognize lettuce health. Image processing has been used by [9] to better diagnose plant diseases. Identification of a disease includes steps such as image gathering, image preprocessing, image segmentation, extraction and classification of features. The paper addressed the approaches used to diagnose diseases of plants using images of leaves. The paper also addressed several segmentation and extraction algorithms used for the identification of plant diseases. Among other studies used image processing are by [10]-[13].

There are other studies used machine learning and deep learning to recognize the health of lettuce. Among these are conducted by [14]-[16]. A review of these different approaches has been conducted by [17], The paper addresses the phases of the general plant disease detection process and comparative study of plant disease detection classification techniques in machine learning. Throughout the analysis, Convolutional Neural Network was found to have strong precision and to classify more specific crop diseases.

The objective of this study is to help the farmer's work efficiency. Providing them a device that can detect a lettuce disease which will help them lessen their work and effort since diagnosing plants nutrient deficiency can be a time consuming for the farmers. Also, this study aims to maximize a farmer's crop yield or plant quality.

The study only covers lettuce health conditions using raspberry pi and training an AI model. The software used here is TensorFlow to help us to implement our trained image which focuses on the lettuce diseases. But the study can only focus on whether the lettuce is healthy or it has a disease.

2. METHODOLOGY

The study as shown in Figure 1 the methodology for lettuce health recognition. First, the images of lettuce have been used as input on the system and then to be processed for training and testing. In the training process, individual lettuce images have been annotated and then the annotated datasets will undergo training and validation. For the process of testing the image, it directly goes in the model then it detects lettuce through the raspberry pi camera then the output will be display in the monitor.

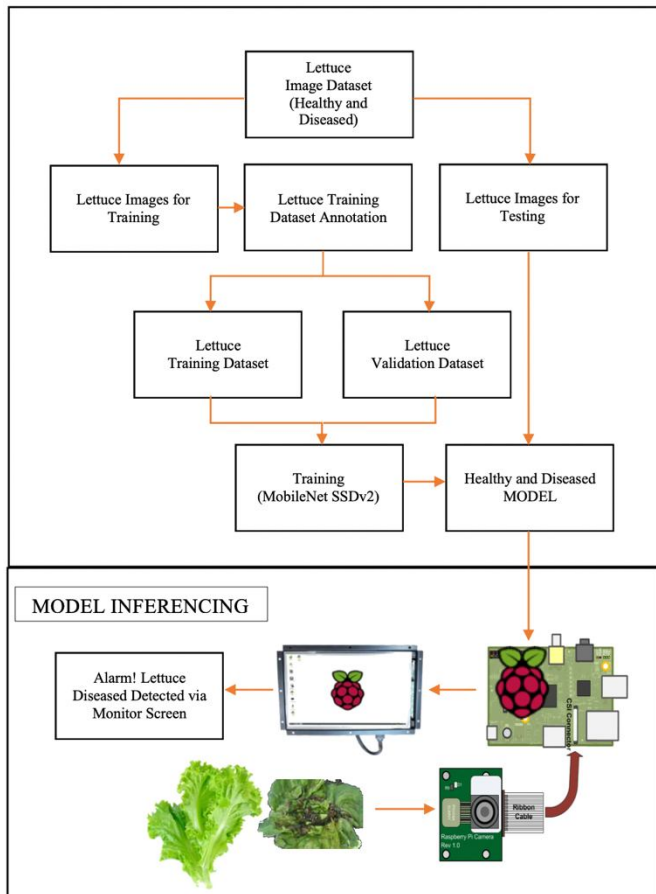


Figure 1: Methodology of Lettuce Health Detection

2.1 Lettuce Images

The study capture images using different phone cameras of a specific plant called “lettuce” that has a condition of healthy and diseased from a different perspective and store it for our datasets needed for training and recognition as shown in Figure 2.



Figure 2: Lettuce Images

2.2 Lettuce Images for Training

After gathering all the datasets, the study used 80% images of lettuce 80% for training. $202 * 80\% = 161$ images were used for training (not annotated yet) as shown in Figure 3.

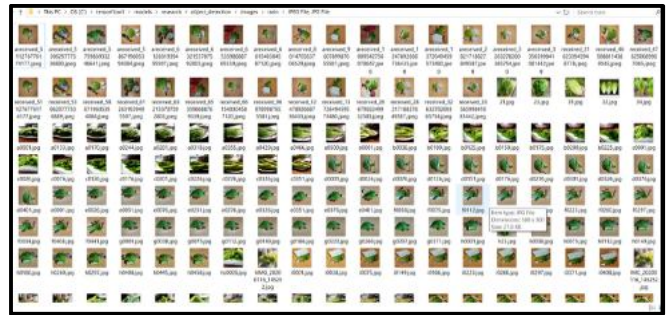


Figure 3: Lettuce Images for Training

2.3 Lettuce Images for Testing

The remaining 20 percent of the datasets were used for validation. The study made sure the \test and \train folders included a variety of images. See Figure 4.



Figure 4: Lettuce Images for Testing

2.4 Annotated Images of a Lettuce using Labelling

The study used a graphical image annotation tool in python called “labelImg” to annotate the lettuce. Generally, it was done by selecting a “zone” of the data (which is the lettuce), and adding a label to this specific zone (labeled it as lettuce) then save it as .xml file. These.xml files were used to build TFRecords, one of the inputs for the TensorFlow trainer. The folders \test and \train comprise an xml file for every image. See Figure 5.

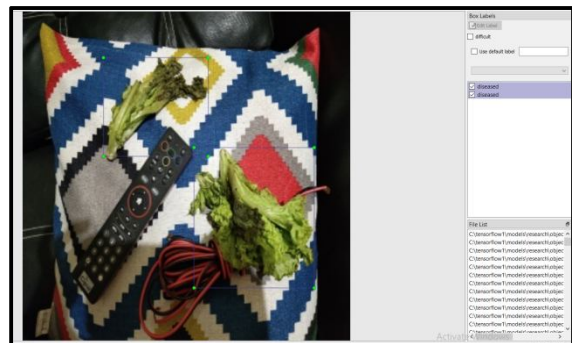


Figure 5: LabelImg Sample Window

2.5 Training Lettuce Dataset

After labeling all the images which serve as input data for the TensorFlow training model, the study generated the

TFRecords. The xml image data is used to create csv files which contain both train and image test data. The following command in the Anaconda command prompt was used from the \object detection folder, as seen in Figure 6, the file .py contains the code for converting .xml files to the .csv file. See Figure 6.

```

Anaconda Prompt (Anaconda3)
(tensorflow) C:\tensorflow\models\research\object_detection>python xml_to_csv.py
successfully converted xml to csv.
successfully converted xml to csv.
    
```

Figure 6: Training Lettuce Dataset

2.6 Testing Lettuce Dataset

It is the same as the training lettuce dataset. The study used the .py file containing the code for converting .xml to .csv file that has been used to train the annotated datasets gathered. See Figure 6.

2.7 Training Using TensorFlowLite

After creating the CSV file, from the \object_detection directory. Each training phase shows the loss. It starts high and gets smaller and lower as training progresses. Trained the model until the loss decreases steadily below 0.05, which requires about 40,000 steps or 2 hours (depending on how efficient the CPU and GPU are). The percentages of losses are different by using a different model. MobileNet-SSD begins with about 20 losses and will be trained before the loss is regularly under 2. Training stopped after the loss fell steadily below 0.05 because it might lead the model to overfit. The study used TensorBoard to show the development of the training job. The TensorBoard service offers facts and diagrams demonstrating how the training progresses. One essential graph is the Loss index, which displays the classifier's total loss over time. See Figure 7.

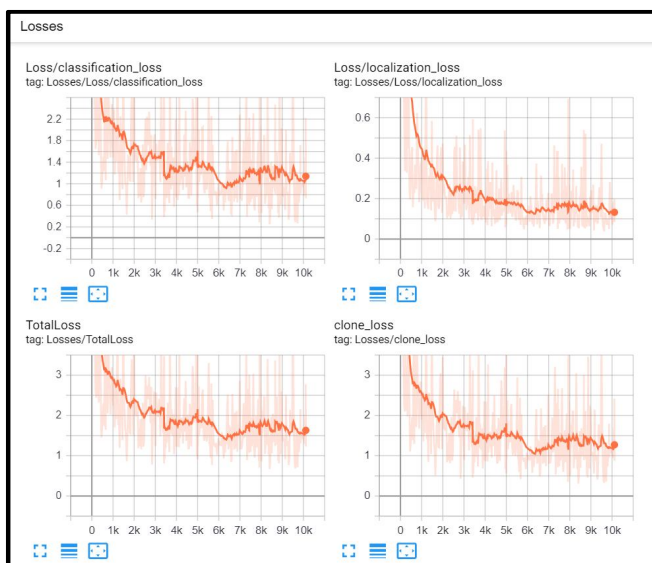


Figure 7: Training Using TensorFlowLite

2.8 Lettuce Model

The last step would be to create the frozen inference graph (.pb file) after the training is complete. The study issues the following command from the folder \object detection, where "XXXX" will be substituted by the highest numbered.ckpt file in the training folder in "model.ckpt-XXXX" "python export_inference_graph.py --input_type image_tensor --pipeline_config_path training/fast_er_rcnn_inception_v2_pets.config --trained_checkpoint_prefix training/model.ckpt-XXXX --output_directory inference_graph". That created a frozen inference_graph.pb file in the folder \object detection\inference graph. The .pb file includes the classifier for object detection and was used to identify the lettuce.

2.9 Diseased Lettuce Alarm System

A USB webcam was used to check the object detector and pointing it to the lettuce (using Object detection webcam.py). The Object_detection_webcam.py contains code that opens the webcam and is used to detect the lettuce. See Figure 8.

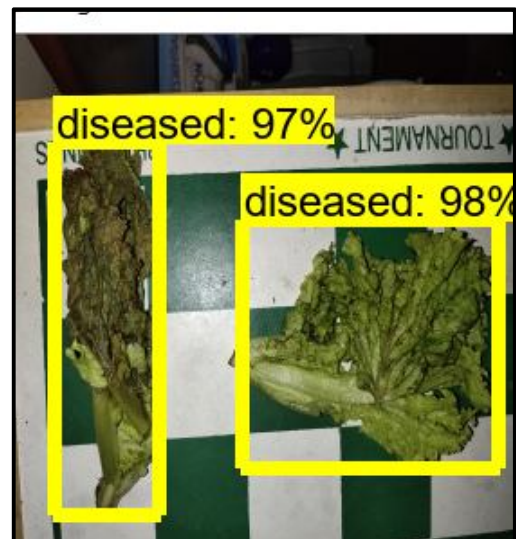


Figure 8: Detect Plant through Raspberry Pi Camera

2.10 Lettuce Condition Health Recognition

The study created a model that can detect whether the lettuce is healthy or not. So, after opening the .py file that can detect lettuce. It detects if the lettuce is healthy or not. It appears on the monitor screen after pointing the lettuce in the camera. The lettuce health condition is displayed on the monitor with the result and its accuracy detection percentage. It shows the percentage of health and diseased lettuce. Detection of lettuce health condition is important to people especially on farmers because not all farmers have the capability to detect if the lettuce is in a health condition or diseased.

3. HARDWARE DESIGN

The study used a box that holds the raspberry pi with a camera. The camera needs to be compact so that detecting the lettuce is stable. On the camera, there is a case with a mount that can adjust the angle of the camera. Figure 9 shows the actual prototype of the study.



Figure 9: Actual Prototype

4. RESULTS AND DISCUSSIONS

4.1 Training Results

Loss functions for classification are computationally feasible loss functions in machine learning and mathematical optimization which reflect the inaccuracy of predictions in the classification problems of the system.

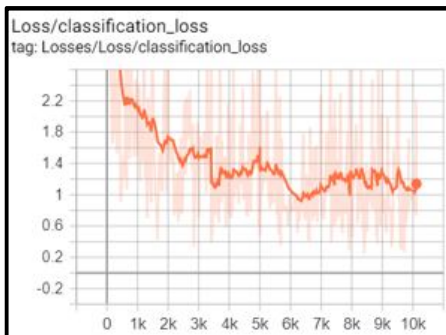


Figure 10: Classification Loss

The X-Axis values are 10k while the Y-Axis represents the classification loss. The red line represents the training and the white line represents the test as shown in Figure 10.

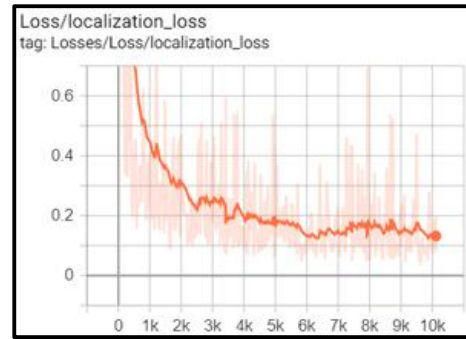


Figure 11: Localization Loss

Figure 11 shows the Localization loss of the training. The X-Axis value is 10k while the Y-Axis represents the number of values in localization

Classification/localization loss values are the product of the loss functions and represent the "price paid for the imprecision of the predictions" in the classification/localization problems. The loss value is the sum of the loss of classification and the loss of position. Such loss values are attempted by the optimization algorithms before the researchers are satisfied with the results.

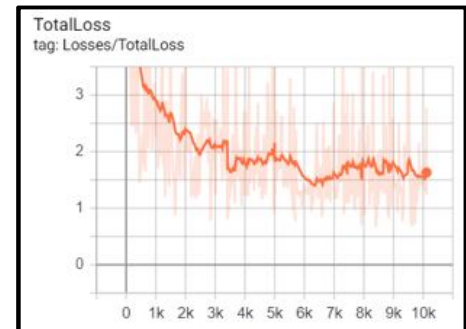


Figure 12: Total Loss

Figure 12 shows the total loss of the training. The X-Axis value is 10k while the Y-axis represents the total loss value.

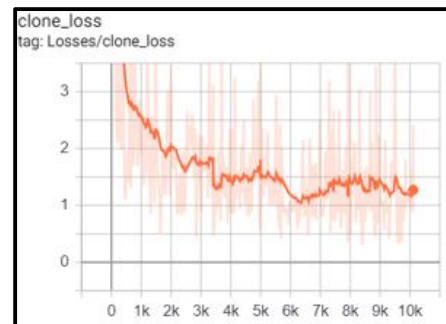


Figure 13: Clone Loss

Figure 13 shows the clone loss of the training. The X-Axis value is 10k while the Y-Axis represents the clone loss value

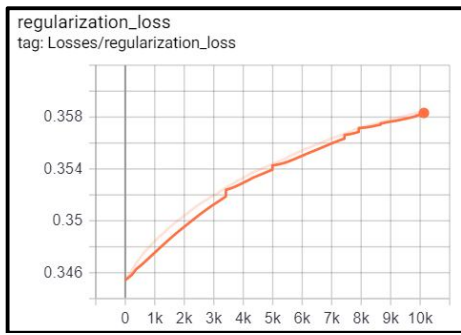


Figure 14: Regularization Loss



Figure 15: Images of Detected Condition of Lettuce

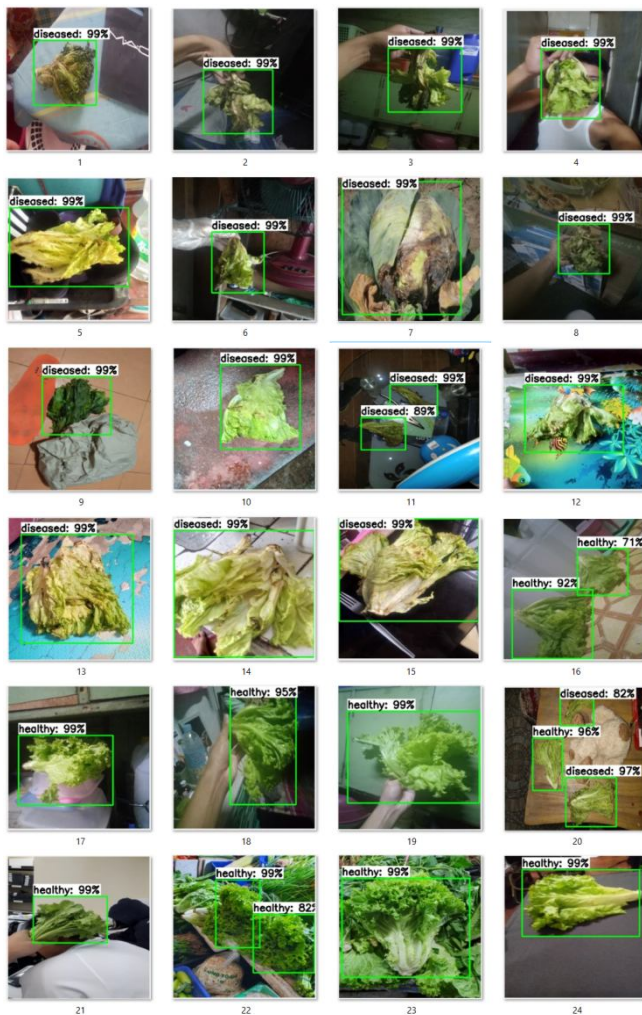
The X-Axis and Y-Axis are nearly equal in value. Regularization techniques are used to further boost the generalization of the optimization process as shown in Figure 14. The X-axis is the number of steps while the Y-axis is the extent of the absence of regularization.

4.2 Testing Results

In getting the mean average, the study used the test case outputs from Figure 15 to get the percentage of the detections. The mean average result is getting the total summation of all the percentages and dividing it to the total number of test cases, the study gets the 97.59% testing result of the system.

Table 1: Actual Testing Results

Test #	Test Case	Expected Output	Actual Output (%Detection)
1	diseased	diseased	99%
2	diseased	diseased	99%
3	diseased	diseased	99%
4	diseased	diseased	99%
5	diseased	diseased	99%
6	diseased	diseased	99%
7	diseased	diseased	99%
8	diseased	diseased	99%
9	diseased	diseased	99%
10	diseased	diseased	99%
11	diseased, diseased	diseased, diseased	94%
12	diseased	diseased	99%
13	diseased	diseased	99%
14	diseased	diseased	99%
15	diseased, healthy	diseased, healthy	99%
16	diseased, diseased	diseased, diseased	81.5%
17	healthy	healthy	99%
18	healthy	healthy	95%
19	healthy, healthy	healthy, healthy	99%
20	healthy, diseased	healthy, diseased	91.67%
21	healthy	healthy	99%
22	healthy, healthy	healthy, healthy	90.5%
23	healthy	healthy	99%
24	healthy	healthy	99%
25	healthy	healthy	99%
26	healthy	healthy	99%
27	healthy	healthy	99%
28	healthy	healthy	99%
29	healthy	healthy	99%
30	healthy	healthy	99%
mean Average			97.59%



5. CONCLUSION AND FUTURE WORKS

This study would improve the way farmers harvest their crops, the efficient harvest will lead to more produces thus more income for the farmers and affordable market price for the consumers. Therefore, with furthermore development of the project, the future of the agriculture section would be smart and thriving. The system proved to be effective as it gained an overall testing accuracy of 97.59%.

The new tool is to use technology to help people understand lettuce diseases even better. Some people struggle to find out if the lettuce is safe or not. Nowadays the use of technology helps people ease their jobs. In conducting this study, it lets

them decide much more quickly whether the lettuce is safe or sick. In conducting this study, it lets them decide much more quickly whether the lettuce is safe or sick.

ACKNOWLEDGEMENT

The author would like to thank particularly his Microprocessor's Class of 2nd SEM SY19-20. Additionally, to MR-SUAVE Lab, and Microprocessors Lab of the Technological Institute of the Philippines.

REFERENCES

1. J. Ryan, **Biological Threat to Agriculture**, *Biosecurity and Bioterrorism*, pp. 185-216, 2016. <https://doi.org/10.1016/B978-0-12-802029-6.00008-6>
2. A. Moreno and A. Fereres, **Virus Diseases in Lettuce in the Mediterranean Basin**, *Advances in Virus Research*, pp. 247-288, 2012.
3. D. HE, J. ZHAN and L. XIE, **Problems, challenges and future of plant disease management: from an ecological point of view**, *Journal of Integrative Agriculture*, vol. 15, no. 4, pp. 705-715, 2016. [https://doi.org/10.1016/S2095-3119\(15\)61300-4](https://doi.org/10.1016/S2095-3119(15)61300-4)
4. D. Salomon and G. Sessa, **Biotechnological strategies for engineering plants with durable resistance to fungal and bacterial pathogens**, *Plant Biotechnology and Agriculture*, pp. 329-342, 2012.
5. P. Moffett, **Mechanisms of Recognition in Dominant R Gene Mediated Resistance**, *Advances in Virus Research*, pp. 1-229, 2009.
6. L. Shcherbakova, **Advanced methods of plant pathogen diagnostics**, *Comprehensive and Molecular Phytopathology*, pp. 75-116, 2007.
7. J. Legg et al., **Cassava Virus Diseases**, *Advances in Virus Research*, pp. 85-142, 2015. <https://doi.org/10.1016/bs.aivir.2014.10.001>
8. U. Mc Carthy, I. Uysal, R. Badia-Melis, S. Mercier, C. O'Donnell and A. Ktenioudaki, **Global food security – Issues, challenges and technological solutions**, *Trends in Food Science & Technology*, vol. 77, pp. 11-20, 2018.
9. S. D. Khirade and A. B. Patil, **Plant Disease Detection Using Image Processing**, 2015 International Conference on Computing Communication Control and Automation, Pune, 2015, pp. 768-771.
10. A. Devaraj, K. Rathan, S. Jaahnavi and K. Indira, **Identification of Plant Disease using Image Processing Technique**, 2019 International Conference on Communication and Signal Processing (ICCSP), Chennai, India, 2019, pp. 0749-0753. <https://doi.org/10.1109/ICCSP.2019.8698056>
11. J. F. Molina, R. Gil, C. Bojacá, G. Díaz and H. Franco, **Color and size image dataset normalization protocol for natural image classification: A case study in tomato crop pathologies**, Symposium of Signals, Images and Artificial Vision - 2013: STSIVA - 2013, Bogota, 2013, pp. 1-5.
12. F. J. P. Montalbo and A. A. Hernandez, **Classification of Fish Species with Augmented Data using Deep Convolutional Neural Network**, 2019 IEEE 9th International Conference on System Engineering and Technology (ICSET), 2019.
13. A. S. Alon, **A YOLOv3 Inference Approach for Student Attendance Face Recognition System**, *International Journal of Emerging Trends in Engineering Research*, vol. 8, no. 2, pp. 384-390, 2020. <https://doi.org/10.30534/ijeter/2020/24822020>
14. M. Sardogan, A. Tuncer and Y. Ozen, **Plant Leaf Disease Detection and Classification Based on CNN with LVQ Algorithm**, 2018 3rd International Conference on Computer Science and Engineering (UBMK), Sarajevo, 2018, pp. 382-385. <https://doi.org/10.1109/UBMK.2018.8566635>
15. N. K. Gattim, **Plant Leaf Disease Detection Using SVM Technique**, *International Journal of Emerging Trends in Engineering Research*, vol. 7, no. 11, pp. 634-637, 2019 <https://doi.org/10.30534/ijeter/2019/367112019>
16. N. S. C, **Heuristics of Machine Learning on Embedded Intelligence Device for Multiple Object Detection**, *International Journal of Emerging Trends in Engineering Research*, vol. 8, no. 3, pp. 787-791, 2020. <https://doi.org/10.30534/ijeter/2020/29832020>
17. U. Shruthi, V. Nagaveni and B. K. Raghavendra, **A Review on Machine Learning Classification Techniques for Plant Disease Detection**, 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS), Coimbatore, India, 2019, pp. 281-284. <https://doi.org/10.1109/ICACCS.2019.8728415>