IJRAR.ORG E-ISSN: 2348-1269, P-ISSN: 2349-5138

INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS (IJRAR) | IJRAR.ORG

An International Open Access, Peer-reviewed, Refereed Journal

Efficient Precision Agriculture with Python-based Raspberry Pi Image Processing for Real-Time Plant Target Identification

Mrutyunjay Padhiary[1*], Nidhi Rani[2], Debapam Saha[3], Javed Akhtar Barbhuiya[4], L. N. Sethi[5]

[1]Assistant Professor, [2, 4]M.Tech, [5]Professor, Department of Agricultural Engineering, Assam University Silchar, India – 788011

[3]M.Tech, Agriculture and Food Engineering Department, IIT Kharagpur, India – 721302

Abstract : Precision agriculture has emerged as a crucial solution to feed the rapidly growing global population, considering the limited resources available for traditional farming practices. Advancements in farm machinery have transformed agriculture into a more scientific endeavor. In this context, we propose an innovative image processing algorithm that utilizes a Raspberry Pi, a camera, and a power source to recognize and locate plant targets in real-time to enhance yield and quality, subjected to various abiotic and biotic factors. By capturing images through the camera, this system minimizes the risk of crop damage caused by careless handling. This research presents a visual assessment of computer vision, image processing and machine learning algorithms for plant target detection. A laptop was used to control the Raspberry Pi, which is mounted on a fixed frame and equipped with a camera for image capturing. A unique dataset for various plant target was created from images captured from plantations in Assam University's campus. By leveraging tensor flow object detection API, an image processing system of plant targets have been developed with Python programming. The parameters of the proposed system were optimized for the identification task and achieved an accuracy of 91% with a delayed timing of 167ms. Precision agriculture, which covers a sizable portion of the agricultural industry, will be highly benefitted from automated plant target detection technique for harvesting, pruning, cutting, grading and spraying etc. Hence, this cutting-edge technology will ensure sustainable growth and lower the risks associated with farm labor and food security.

Index Terms: **Precision agriculture, Image processing, Raspberry Pi, Machine learning, Real-time identification.**

I. INTRODUCTION

Recent ameliorations in the agriculture sector have led to the emergence of precision agriculture, revolutionizing farming practices by enhancing fertility rates, increasing yield production, optimizing time management, and minimizing water and chemical wastage. To meet the food demands of the world's 7.9 billion population and avert a global food crisis, technological interventions are essential to achieve higher crop yields. While traditional agriculture remains deeply rooted in our history, solely relying on it hinders the possibility of another green revolution. In the past, farmers relied on their own observational research to assess crops, but this approach lacks precision, especially for inexperienced farmers. Identifying the exact source of crop damage through visual observation can be challenging, leading to significant losses in yield, time, and resources. However, the integration of machine learning, computer vision, and neural networks into precision agriculture has introduced a transformative approach, providing accurate plant condition diagnosis and enabling timely protection for vast acres of land, even in cases of asymptomatic infections. This new perspective revitalizes the outdated farming methods and sets the stage for a more sustainable and productive future in agriculture [1]. In early farm machinery, progress was often rudimentary and experimental, relying on a trial-and-error approach. However, in modern times, farm machinery has evolved into a more scientific and precise field. This transformation is particularly crucial in emerging nations like India, where agriculture plays a significant role. To address the challenges faced by farmers and prevent crop losses, there is a growing demand for automated systems capable of identifying plant illnesses accurately. Such systems can empower farmers by providing timely and informed actions to safeguard their crops and improve agricultural productivity.

In the realm of image processing systems, the latest technology leverages high-performance computers and application-oriented software. This shift is a response to the vast amounts of data that require processing and the time-intensive nature of real-time data analysis. In the context of mechanizing the agriculture sector, significant research efforts have been dedicated to developing diverse crop image classifications. Python, MATLAB, LABVIEW vision, and various advanced techniques such as SOM (Self Organizing Map) neural network, zooming algorithms, K-Means clustering method, Fuzzy logic SVMRBF, SVMPOLY, and Back propagation method have all contributed to this progress. [2]. A survey on different image processing techniques used to classify images for disease detection, which was firstly done by [3], who proposed an android intelligent device for diagnosing wheat disease.

Additionally, [4] implemented a video image processing algorithm on Raspberry Pi 3 to detect and track vehicles on roads. In precision farming, machine learning, particularly using convolutional neural networks (CNNs), has proven effective in identifying diseases in various plants like apple, tomato, maize, and grape, as demonstrated by [5] with an 86% accuracy on a dataset of 15,210 leaf images belonging to 10 classes. Moreover, [6] proposed a VGG16 (CNN) with transfer learning for rice plant disease identification, achieving an accuracy of 92.4%. There has been disease detection in coffee leaf using techniques such as fuzzy logic, radial basis function neural network, CNN with data augmentation, and transfer learning [7]. Employing a dedicated CNN hardware block-equipped Intel Movidius Neural Compute Stick model, [8] achieved an 88.46-93.5% accuracy on a Raspberry Pi 3 for identifying diseased leaves. Image processing plays a crucial role in identifying crop requirements and ensuring timeliness, especially with real-time image capture using USB or Raspberry Pi cameras [9]. Furthermore, [10] successfully identified plants through leaf vein image processing using Raspberry Pi with an 84.29% accuracy rate. These advancements contribute to sustainable growth, reduce risks associated with farm labor, and enhance food security.

In this research, we explore the use of artificial neural network training and deep learning methods for plant target detection through visual assessment. The adoption of above-mentioned systems in the agriculture sector is limited due to their high cost and complexity. To address this challenge, we propose the development of a novel mobile system that allows remote desktops or mobile devices to access real-time data efficiently.

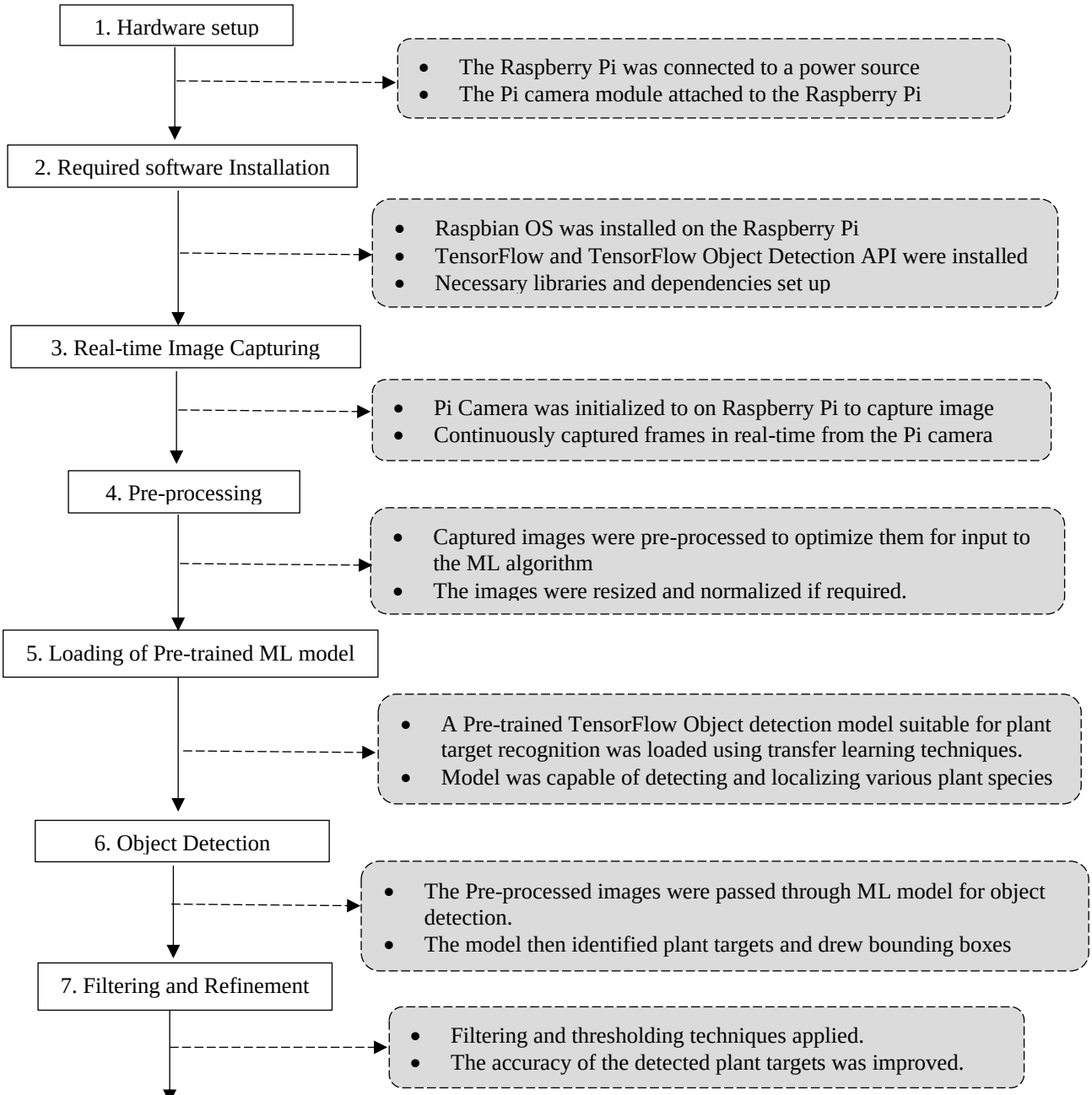
A. Objective:

1) The aim of this study is to design an image processing system based on Raspberry Pi for precision agriculture applications.
2) To create a Python-based program that provides intelligent decision support for different agricultural systems and assess its performance in real-field conditions.

B. Proposed system:

1) The suggested system model utilizes a Raspberry Pi connected via USB or a Pi camera.
2) The Raspberry Pi allows for view selection and can be viewed by laptops, Android devices, or distant desktops.

II. RESEARCH METHODOLOGY



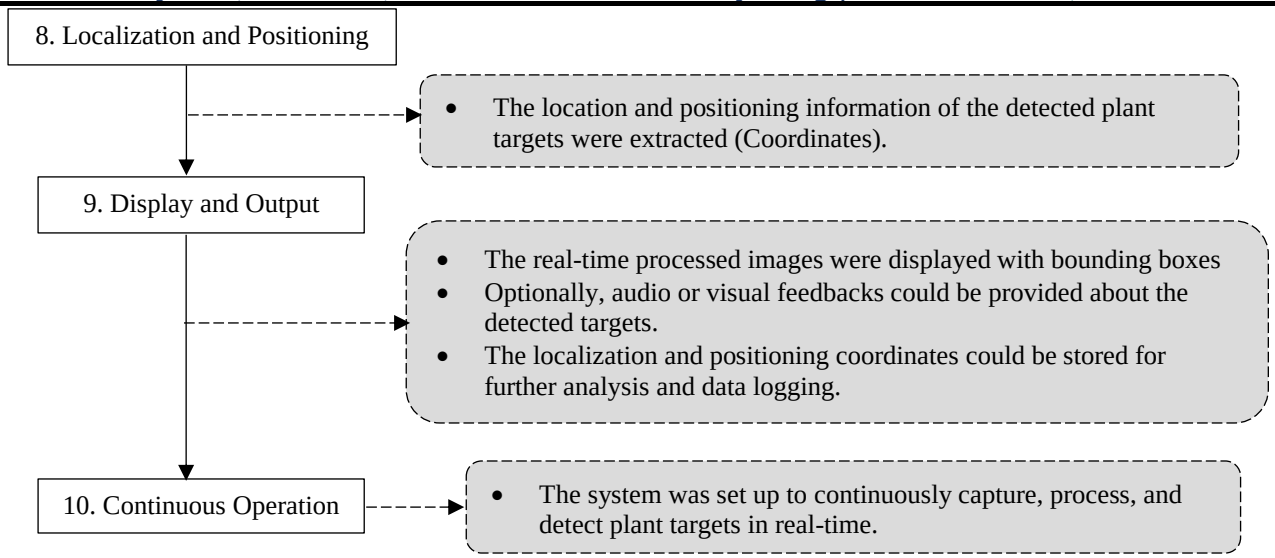


Fig.1 Block diagram of current research work

III. HARDWARE REQUIREMENTS

1) Camera Sensors: A camera forms the indispensable component of the computer vision system. The camera captures the image or video which is further transferred to central processing system for identification. A Pi camera with MIPI interface was selected considering the most flexible way to control the Camera Module or HQ Camera using Python. Pi camera captures higher-resolution images (2592 x1944) than the standard Camera Module with 60-90 fps and supports 1080p, 720p and 480p. The camera dimensions are of the order 32*26*27.5. The camera was connected to the raspberry pi through 2 lane MIPI CSI camera port.

2) Raspberry Pi: Raspbery pi is a single board microcontroller decvice having RAM, I/O Pins, CPU/GPU and USB hub. This low-cost computer is utilised for writing using the Python programming language along with a variety of other tasks. The specifications include: 64-bit quad-core ARM Cortex-A72 at 1.5GHz CPU, VideoCore VI at 500MHz GPU, 2GB RAM, Gigabit Ethernet, dual-band 802.11ac and Bluetooth 5.0 for connectivity. Further it is equipped with Camera Serial Interface and Display Serial Interface (DSI) with MicroSD (up to 512GB) storage. For A/V Outputs, it has 3.5 mm analogue AV jack and 2 × micro-HDMI 2.0. It has GPIO pins that are a component of a GPIO port, Each GPIO pin has a name and a numerical ID to identify and the total number of GPIO pins mounted on Raspberry Pi is 40.

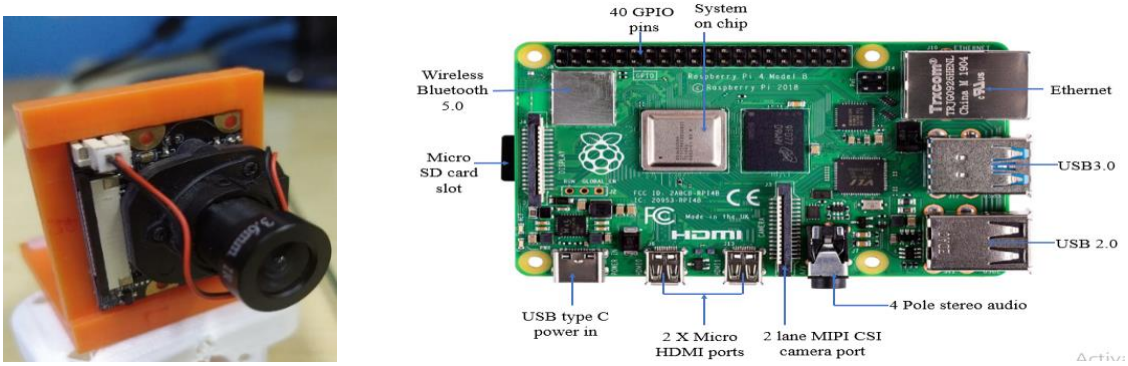


Fig.2 Pi camera and Raspberry Pi-4 microcontroller board

3) Other accessories: LCD screen and connecting wires. The LCD is powered through one of the USB ports and the video signal is provided through mini-HDMI port connected by HDMI cable. The key board and mouse are connected through remaining empty USB port.

IV. SOFTWARE REQUIREMENTS

1) Model architecture for transfer learning: The transfer learning leverages the knowledge gained by the model while solving one problem and applying it to a dissimilar but associated problem. Building a model for image classification from scratch requires colossal number of images besides time in training and GPU for storage and processing. The TF techniques saves time without compromising with the accuracy of the trained model. In computer vision applications, transfer learning is usually expressed through pre trained models. The pre trained models such as VGG, Inception and MobileNet are trained on large standard dataset i.e., COCO dataset. SSD MobileNet V2 FPNLite 320x320 was selected for final implemention with raspberry pi.

2) Preparation of custom dataset: A distinctive database comprising images of diverse plant targets was compiled using photographs taken from plantations in Assam University's campus. The captured images were first resized to 500 x 500 pixels and then preprocessed to remove background noise from the image so that the network could learn specific feature. For the training of models, the processed image was labelled using LABELING tool. The main function of the tool was to create bonding box around the target location in the image and provide unique name.

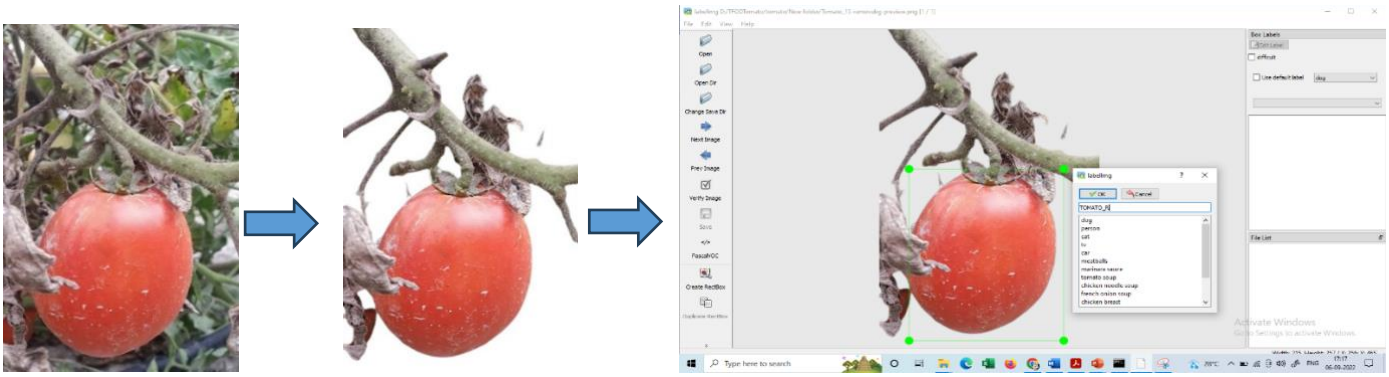


Fig.3 Real time image capturing and Pre-processing

All processed images were manually labelled and saved in a file. With 168 images of tomatoes with different maturing stage, a unique customized database was created for various plant targets. The final file contains two objects i.e., the image and its corresponding labelled file in xml format as shown in Fig.4. The XML file contains information about the class of the object, label name, size of the bonding box and location of the file. The labelled dataset was further fragmented into two sections, viz. train and test sections containing 152 and 16 images respectively.

Fig.4 View of file containing images and corresponding labels

3) Coding and folders: The coding for the development of the real time object detection was written in Jupyter Notebook, an open-source package under anaconda environment. After the creation and activation of the virtual environment, a folder system was made for easy understanding and access to results after the training. The base folder contains subfolders such as virtual environment folder and the training folder with python code. In training folder, another set of 4 subfolders were created. The model's folder contained the TensorFlow models for object detection. The protoc and scripts folder contained protocol buffers and generated scripts respectively. The workspace folder has four subfolders within it. The annotation folder contained the pbtxt file, test and train record file. The pbtxt file contained the class label name with a unique id. The train and test record file were used in training the neural network. The images folder had test and train folder within it. These folders contained images with their corresponding labels. The model's folder stored the result of the training in the form of checkpoints which was used to load the trained model for real time detection.

4) Object Detection API: The TensorFlow Object Detection API was installed on the local machine. The pre trained model SSD MobileNet V2 FPNLite 320x320 was downloaded in the pre-trained-models folder inside workspace. A labelmap.pbtxt file had been created and stored in annotation folder and pipeline configuration was updated. the training of the pretrained model was conducted in 1500, 2000, 2500 and 3000 steps. Then evaluation was done and trained model was converted into TFlite and stored in model folder as detect.tflite. The created tflite file was transferred to folder in raspberry pi operating system. The folder contains a python file with code for object detection and a labels.txt file.

5) Thonny Python IDE: An integrated development environment (IDE) that combines many tools required to create or develop software into a single user interface or environment. It contains functional Toolbar, script area and Python shell.

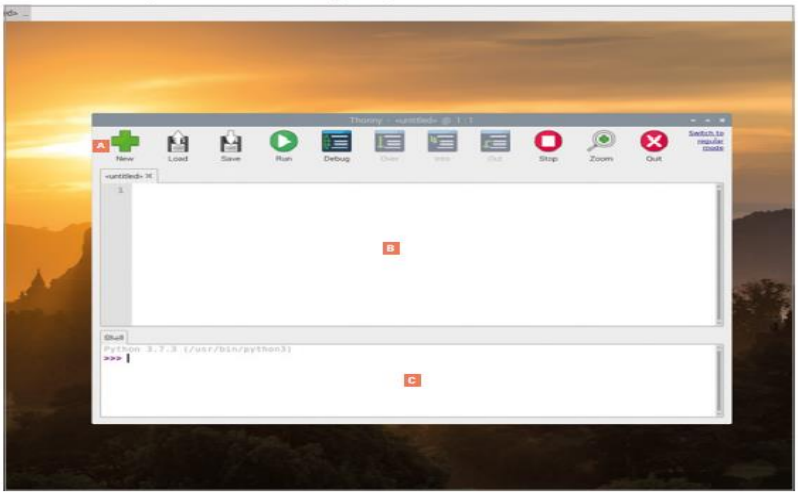


Fig.5 Interface of Thonny Python IDE

V. EVALUATION

There were 168 instances in the tomato dataset collection The network was fed the output of the suggested method using 320x320-pixel input images. To attain accuracy levels above 96%, the maximum epoch of network hyper-parameters such learning rate is modified throughout the training phase. The learning rate's optimized value was .0004. A lightweight model for real-time data inference was then developed using feature visualization. The validation process used a total of 168 photos from the database. The degree of classification accuracy was achieved by various machine learning techniques following TensorFlow API model preprocessing. The outcome was what a machine learning system would have produced had the tomato picture input dataset not been preprocessed.

VI. EXPERIMENTATION RESULTS

1) Result of Model Evaluation: The result of model evaluation was recorded from tensor board in the form of total loss while training and mean average precision while testing. The training was carried out in 1k, 2k, 25k, 30k and 35k steps. The total loss and mean average precision are tabulated in Table 1. The best result was obtained with training on preprocessed dataset in 2000 steps which corresponds to total loss of 0.24 i.e., minimum among the tested group and mean average precision of 0.69. The performance of the model can be improved by adding more images in the custom dataset. The bar graph shown in Fig.6 clearly indicates that the total loss is minimum and mean average precision is maximum for 2000 training steps as compared to other training steps.

Table 1: Total loss and mean average precision in various training steps

Training steps	Total loss	mAP
1000	0.34	0.40
1500	0.33	0.52
2000	0.24	0.69
2500	0.27	0.55
3000	0.28	0.64
3500	0.29	0.45

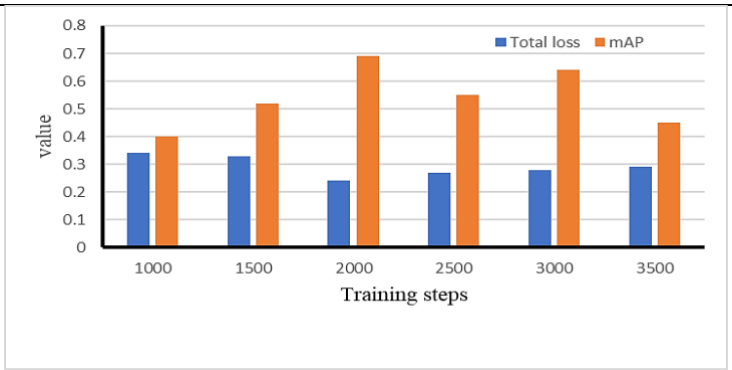


Fig.6 Variation in total loss and mAP with respect to training steps

2) Performance on laptop through webcam: The developed real time detection model was tested for its accuracy in laboratory condition for detecting ripen tomato and unripen tomato. The observation in recorded in Table 2. The evaluation parameters such as overall accuracy, misclassification rate, precision and recall for ripen and unripen tomato were compared. The overall accuracy of the model was 79% with misclassification rate of 21%. The model was able to assign correct label to true class for 79% of times in case of ripen and unripen tomato. The misclassification rate was because of the inability of the model to detect objects from distance. The objects were needed to bring closer to the camera for more accurate predictions. The model was able to precisely predict TOMATO_R and TOMATO_G, 78% and 80% of times, respectively. The precision in the case of unripen tomato was higher as compared to ripen tomato because the custom dataset contained more images of unripen tomato as compared to ripen. The recall rate was 86% for unripen tomato and 70% for ripen tomato.

Table 2: Matrix from lab observation

Predicted class		**True class**	
		Ripen tomato	**Unripen tomato**
	TOMATO_R	7	2
	TOMATO_G	3	12

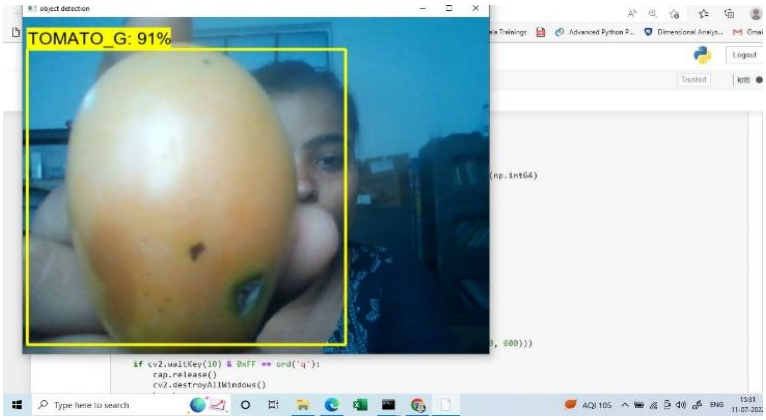


Fig.7 Real time detection of sample in lab

Table 3: Values of evaluation parameters

Parameters	**Ripen tomato (TOMATO_R)**	**Unripen tomato (TOMATO_G)**
True positive (TP)	7	12
True Negative (TN)	12	7
False negative (FN)	3	2
False Positive (FP)	2	3
Overall accuracy	0.79	0.79
Misclassification rate	0.21	0.21
Precision	0.78	0.80
Recall	0.70	0.86

(TP: The number of real targets that were successfully identified; TN: The number of false targets that were successfully identified; FN: The number of undetected targets; FP: The number of undetected real targets)

3) Performance on raspberry pi: The developed real time detection model was tested for its accuracy in laboratory condition on raspberry pi with pi camera for detecting ripen tomato and unripen tomato. The observation in recorded in Table 4. Here, the overall accuracy of the model was 58% with misclassification rate of 42%. The model was able to assign correct label to true class for 58% of times in case of ripen and unripen tomato. The misclassification rate is because of the inability of the model to detect the object. The objects were needed to bring closer to the camera for more accurate predictions. The model was able to precisely predict TOMATO_R and TOMATO_G, 45% and 69% of times, respectively. The precision in the case of unripen tomato was higher as compared to ripen tomato because the custom dataset contained more images of unripen tomato as compared to ripen. The recall rate was 56% for unripen tomato and 60% for ripen tomato.

The performance of the model decreased on the raspberry pi because of the limited computational and processing power of the system. The deep learning model cannot be trained on the pi platform or alternatives. The training has to be done on local pc and then transferred to pi platform. The threshold value has to be kept low so that the model can detect the object. If the threshold value is very high, the model on pi platform even will not detect the object.

Table 4: Matrix from lab observation on Raspberry Pi

		True class	
		Ripen tomato	Unripen tomato
Predicted class	TOMATO_R	5	6
	TOMATO_G	4	9

Table 5: Values of evaluation parameters on Raspberry pi

Parameters	Ripen tomato (TOMATO_R)	Unripen tomato (TOMATO_G)
True positive (TP)	5.00	9.00
True Negative (TN)	9.00	5.00
False negative (FN)	4.00	6.00
False Positive (FP)	6.00	4.00
Overall accuracy	0.58	0.58
Misclassification rate	0.42	0.42
Precision	0.45	0.69
Recall	0.56	0.60

VII. CONCLUSION

The developed methodology combining a Raspberry Pi, camera, and transfer learning with TensorFlow Object Detection API has demonstrated successful real-time recognition and localization of plant targets. The system's ability to identify various plant targets and their locations has significant applications in agriculture and environmental monitoring. By leveraging transfer learning, the model achieved high accuracy despite limited computational resources on the Raspberry Pi, making it a cost-effective solution for practical deployments. Furthermore, the work can be extended by refining the object detection model through continuous optimization, expanding the dataset to improve generalization, and exploring cloud integration for scalability. With these advancements, the technology holds promising potential for precision agriculture, biodiversity studies, and sustainable environmental management.

REFERENCES

1. Padma, U., S. Jagadish, and M.K. Singh, *Recognition of plant's leaf infection by image processing approach.* Materials Today: Proceedings, 2022. **51**: p. 914-917.
2. Raut, S. and A. Fulsunge, *Plant disease detection in image processing using MATLAB.* International journal of innovative Research in science, Engineering and Technology, 2017. **6**(6): p. 10373-10381.
3. Xia, Y., Y. Li, and C. Li, *Intelligent diagnose system of wheat diseases based on android phone.* JOURNAL OF INFORMATION &COMPUTATIONAL SCIENCE, 2015. **12**(18): p. 6845-6852.
4. Anandhalli, M. and V.P. Baligar, *A novel approach in real-time vehicle detection and tracking using Raspberry Pi.* Alexandria engineering journal, 2018. **57**(3): p. 1597-1607.
5. Beyene, H., D.N.A. Joshi, and D.K. Kotecha, *Plant diseases prediction using image processing and machine learning techniques: survey.* Int J Comput Appl, 2018. **1**(8): p. 226-37.
6. Ghosal, S. and K. Sarkar. *Rice leaf diseases classification using CNN with transfer learning*. in *2020 IEEE Calcutta Conference (CALCON)*. 2020. IEEE.
7. Kumar, M., P. Gupta, and P. Madhav. *Disease detection in coffee plants using convolutional neural network*. in *2020 5th International Conference on Communication and Electronics Systems (ICCES)*. 2020. IEEE.
8. Mishra, S., R. Sachan, and D. Rajpal, *Deep convolutional neural network based detection system for real-time corn plant disease recognition.* Procedia Computer Science, 2020. **167**: p. 2003-2010.
9. Shilpashree, K., H. Lokesha, and H. Shivkumar, *Implementation of image processing on Raspberry Pi.* International journal of advanced research in computer and communication engineering, 2015. **4**(5): p. 199-202.
10. Selda, J.D.S., et al. *Plant identification by image processing of leaf veins*. in *Proceedings of the International Conference on Imaging, Signal Processing and Communication*. 2017.