Fruit Categorization And Disease Detection Using Ml Raspberry Pi Based Fruit Categorization And Quality Maintenance With Disease Detection Using Ai And Machine Learning

Saleem Ulla Shariff , M G Guru Basavanna

Abstract: In this paper we discuss about the fruit categorization and quality maintenance using raspberry pi board by leveraging the machine learning concepts. India is predominantly agro based economy with Agriculture as the main source of income for the farmers who are termed as the backbone of the country. Our farmers are known to use typical old methods and due to lack of education, they are still far from incorporating the advanced technical tools in agriculture. We are proposing a low cost yet powerful fruit quality maintenance device which can be helpful for our fruit merchants and farmers. Fruit Detection classification and categorization has been implemented in this paper using Machine learning and embedded concepts. We have selected Apple, Banana, Orange, Papaya etc. fruits for the demonstration. We studied the fruit detection by methods such Haar cascade classifier and tensor flow classifier. We trained the fruit classifier by using machine learning concepts and obtained the trained classifier to detect & categorize the fruits with quality. The electronic components used here are Raspberry Pi. Instead of raspberry pi, the laptop with Linux operating system (Ubuntu) can be used. Through image processing & machine learning algorithms we identify the type of fruit and its quality. An audio acknowledgment is given about the identification of the type of fruit while processing the fruit for packaging. In further enhancement we can develop a robot which can be used to separate the raw and ripe fruits with the help of detection algorithm used in this project.

Keywords: Raspberry Pi; Fruits; Quality; Disease; Machine Learning; Aritificial Intelligence

1. INTRODUCTION

Tensor flow is one of the powerful machine learning framework used nowadays and with the availability of Tensor Flow Lite Micro in the Arduino Library Manager, Arduino can made machine learning easy enough for anyone to use [2]. A project by Arduino on fruit categorization has been an inspiration for us. Tensor Flow Lite Micro library and the Arduino Nano 33 BLE Sense's microcontroller which has colorimeter and proximity sensor can be used to classify the objects and to achieve this small neural network on the board itself needs to be run. When the object is brought close enough the sampling of the colour using the onboard RGB sensor which can be assumed as a 1 pixel color camera is done. Even though few limitations are there with this method but it provides a quick way of classifying the objects using a small amount of resources using Arduino board [2] which includes an onboard colorimeter [2]. The model is trained using tensor flow by using jupyter. As per the article [26] the main feature of Arduino Nano 33 BLE sense board is selection of sensors, and the possibility of running Edge Computing applications (AI) on it using Tiny ML.

a) Tensor Flow:

Tensor Flow™ is an open source software library for using data flow graphs and the Nodes in the graph which represents mathematical operations for numerical computation. The Multidimensional data arrays (tensors) communicated between them are represented by the graph edges [23]. The architecture which is quite flexible facilitates to deploy computation to CPUs or GPUs in a desktop, server, or mobile device with a single API [23]. For implementing machine learning and deep learning applications it has been used [18]. Google team created the Tensor Flow framework in Python programming language to develop and research on fascinating ideas on artificial intelligence (AI). It is an easy to understand framework for anyone who know Python language [29]. As discussed Tensor Flow is a software library or framework, designed to implement the machine learning and deep learning concepts in the easiest manner. It is an end-to-end open source platform for machine learning. Machine learning and deep learning basically constitute the artificial intelligence AI. Developers easily build and deploy ML powered applications as it has comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in ML and AI. Tensor Flow is used in several projects, such as the Inception Image Classification Model which introduced a state of the art network for classification and detection in the Image Net Large-Scale Visual Recognition Challenge 2014, in this project while keeping the computational budget constant and adjusting the network width and depth the usage of the computing resources is improved [3]. In Another project, Tensor Flow framework is employed in Deep Speech by Mozilla. Based on Baidu's Deep Speech architecture which is a state of the art recognition system developed using end-to-end deep learning, an open source Speech-To-Text engine Deep speech was developed by Mozilla.

b) Machine Learning:

Machine learning is way of getting the computers to do as per the designed programs and algorithms which on perfection can one day lead us progress towards human-level AI [29]. Following types of patterns are included in Machine learning

- Supervised learning pattern
- Unsupervised learning pattern

c) Deep Learning :

Deep learning is inspired by the structure and function of the brain called artificial neural networks. Through supervised learning or learning from labeled data and algorithms deep learning goes through the same process which includes a hierarchy of nonlinear transformation of input that can be used to generate a statistical model as output [29].

d) Transfer Learning:

Transfer learning (TL) is a technique in machine learning (ML) that uses the concept of storing knowledge gained while solving one problem and applying the learnt techniques to solve a different related problem [4]. The lessons learnt to recognize cycles could be applied on when trying to recognize the Bikes or scooter. Reusing or transferring the information from previously learned tasks for the learning of new tasks is the concept of transfer learning and it has the potential to significantly improve the sample efficiency of a reinforcement learning agent. Hence Transfer learning can be defined as a machine learning technique which utilizes a pre-trained neural network [22]. For example, the image recognition model called Inception-v3 consists of two parts:

- Feature extraction part with a convolutional neural network.
- Classification part with fully-connected and soft max layers.

The Inception-v3 pre-trained model achieves accuracy for recognizing general objects with 1000 classes, like “flowers”, “Zebra” and “Dishwasher” etc. In the first part the model extracts general features from input images and then in second part it classifies them based on those features. In transfer learning, to build a new model to classify the original dataset, we reuse the feature extraction part and then re-train the classification part with the original dataset. Since there is no need to train the feature extraction part we can train the model with less computational resources and training time. In this paper implementation we have taken the Google's image recognition model called Inception and have trained our model based on the fruits data set downloaded from the Mendeley’s [22-23]. After the training has been performed the image categorization has been done, the environment set up in Raspberry Pi is done & the model has been run. For Results study, the model has been trained in laptop with Ubuntu as OS and the same model has been used in Pi for image classification. The image classification probability has been studied & compare in laptop & raspberry Pi. Transfer learning and fine-tuning: A pre-trained model is a saved network of large dataset that was previously trained on a large-scale image-classification task. Usually the pre trained model is used as it is or with the transfer learning it would be customized for a given task. Using transfer learning for image classification once a model is trained on a large and general enough dataset, the trained model will serve as a generic model of the visual world, the advantage of learned feature maps helps in without having to start training from scratch.

The two ways to customize a pre trained model used in this paper include:

1. Feature Extraction: Extract meaningful features from new samples by using representations learned by a previous network. We are not required to (re)train the entire model but add a new classifier, which will be trained from scratch, on top of the pre trained model so that it can repurpose the feature maps learned previously for the dataset. The Features that are generically useful for classifying pictures are already contained in base convolutional network. However, the final, classification part of the pre trained model is specific to the original classification task and subsequently specific to the set of classes on which the model was trained.
2. Fine-Tuning: To "fine-tune" the higher-order feature representations in the base model, Unfreeze a few of the top layers of a frozen model base. Then jointly train both the newly-added classifier layers and the last layers of the base model in order to make them more relevant for the specific task.

The general followed machine learning workflow will include

a) Examining and understanding the data
b) Building an input pipeline using Keras Image Data Generator
c) Compose the model by Loading in pre trained base model and Stack the classification layers on top
d) Then Train and Evaluate model

II. LITERATURE SURVEY

Many techniques have been developed over the years for the fruits detection and quality maintenance. With the availability of the machine learning frame works such as Tensor Flow Lite Micro in the Arduino Library Manager, the fruit detection has been implemented using Arduino and raspberry pi to make easy enough for enthusiasts to use machine learning.

A. Maintaining the Integrity of the Specifications

The article Fruit identification using Arduino and tensor flow, has discussed through an simpler end-to-end tutorial using the Tensor Flow Lite Micro library and the Arduino Nano 33 BLE Sense’s colorimeter and proximity sensor to classify objects which is achieved by running a small neural network on the board itself [3]. TinyML is known for doing more on the device with less resources- in smaller form-factors, less energy and lower cost silicon. The proximity sensor on the board means we get an instant depth reading of an object in front of the board instead of using a camera. As per this tutorial, the sampling is done on the color – the onboard RGB sensor when the object is close enough which can be viewed as a 1 pixel color camera. The tutorial demonstrates a simple but complete end-to-end TinyML application can be achieved quickly and without a deep background in ML or embedded. Data capture, training, and classifier deployment has been covered in the tutorial. A paper by Horea Muresan etal, “Fruit recognition from images using deep learning”, introduces a new, high-quality, dataset of images containing fruits along with results of some numerical experiment for training a neural network to detect fruits. The paper proposes a new dataset of images named Fruits-360 which is containing popular fruits. The same data set has been used in our paper as well. Madhura.R. Lodam etal, proposed the “Fruit Quality Management & Sorting system” based on GUI in MATLAB used as a reference point to calculate size, height & weight on the basis of that the fruits are sorted. In this paper they are focused on to increase the quality of food products with minimum amount of time by using the GUI which displays all the features and the fruits sorted. Ms. Rupali. S. Jadhav, etal. proposed in the paper “A Fruit Quality Management System Based On Image Processing” that to investigate and control quality, embedded system is used to provide high accuracy of grading, consistent and quantitative information. Grey scale images were used to get proper images to find size of a fruits.

III. FRUITS DATA SET PREPARATION

To prepare the fruits data-set the images were obtained by filming the fruits while they are rotated by a motor and then extracting frames [4]. A short movie of 20 seconds was recorded by planting the fruits in the shaft of a low speed motor (3 rpm) by placing a white sheet of paper as background. Then fruit image was scaled down to 100x100 pixels [8].

Figure 1: *Pomegranate image.*

The below figures are some of the example of the defected fruits.

Figure 2: *Anthraconose diseased fruits (a) Apple (b) Mango (c) Orange (d) Tomato (e) Pomegranate*

IV. IMPLEMENTATION AND RESULT ANALYSIS

A. Problem Statement:

Despite of advancement in computer vision, image processing, recognition and advancement in computer technology, automatic fruit classification is a challenging task. The primary parameters that play vital role while classifying a fruit include the machine learning algorithm that is being used, quality of images in the fruit database, fruit's images' shape and size and fruit's color. Secondary parameters that affect the classification are similar characters of fruits like color, shape, size, etc. If both primary and secondary parameters are not analyzed properly in the beginning then it may cause problem during classification and may lead to less accuracy and unexpected results. Many related works have been conducted in fruit classification using different classification algorithms but those approaches still lack in some aspects. A research in fruit classification has been carried out by just taking only three fruits into consideration with 100% accuracy. However, considering only three fruit in the sample is not enough because the trained model may not recognize the fruit's images' that are out of the training sample. Similarly, proper implementation of machine learning algorithm should also be taken into consideration while performing classification.

B. Scope of the Paper:

The scope of this paper is only limited to edible fruits that are available in the fruit data set which is used to train the system. So, when images of these are provided as input to the system, it may not recognize and may not produce the desired result.

C. **Objective :** The main objectives are to extract features from the fruit's image and To implement the Tensor Flow & Machine Learning algorithms for automatic fruit classification. The main objective of this paper is:

- Detection of Good or Bad Quality fruits efficiently.
- To identify the surface defects of fruits based on the application of Machine learning & Tensor Flow algorithms.
- To develop a web interface platform for testing the prediction of fruit image.

In this section we will discuss about the implementation and results discussion for the fruit detection and categorization using Raspberry Pi. The project model has been trained in an hp core i3 processor based laptop having Ubuntu OS. After the trained model has been tested in both laptop as well as the Raspberry Pi setup.

D. Block Diagram

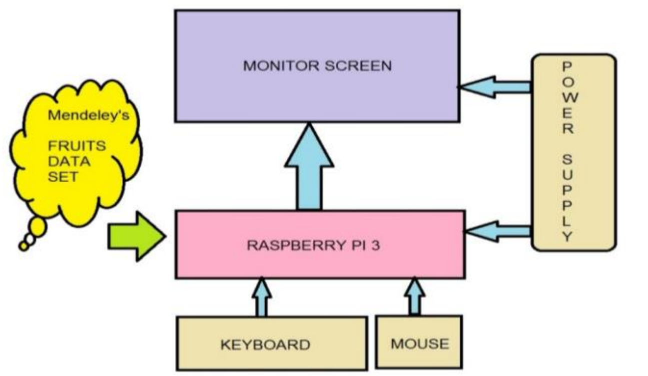

***Figure 3**: Block diagram for fruit detection and categorization using Raspberry Pi*

a) Hardware Components used

The components used for the implementation of the project are listed below.

- ➢ Raspberry Pi 3B Model
- ➢ USB Keyboard and USB Mouse
- ➢ Monitor screen
- ➢ Power supply
- ➢ Arduino (Optional)

b) Software used

We use the Tensor flow framework with trained neural network model for the project. Python 3 language has been used for the implementation. Py Serial library comes in handy for use to connect to the other devices or laptop and interexchange the data between them.

c) Hardware Set Up:

Below screenshots describe the hardware setup, how raspberry pi has been connected with other equipment's.

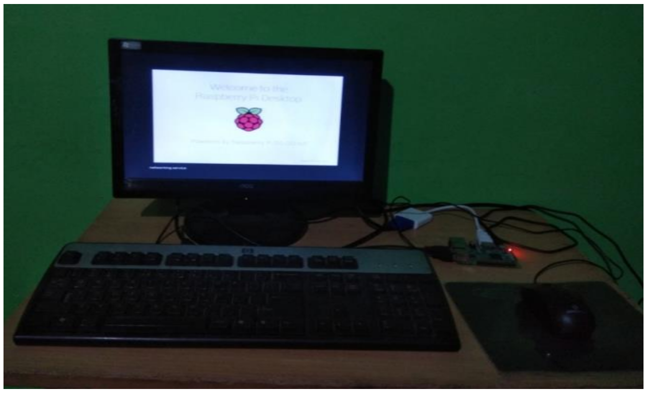

Figure 4a: *Set up of the hardware*

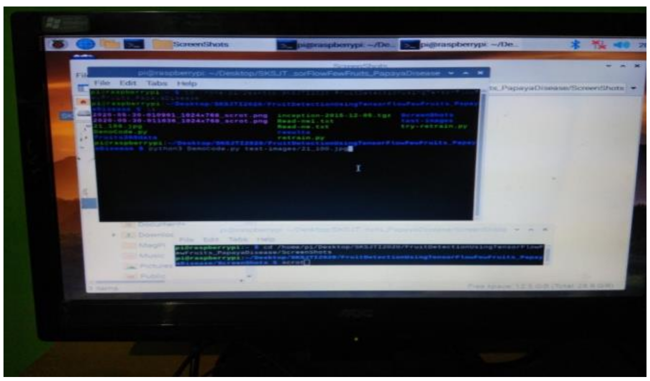

Figure 4b: *Set up of the hardware*

d) **Training Process in Ubuntu:**

The images data set called as Mendeley's fruits 360 data set has been collected from reference and the same has been used to train the model to detect & categorize the fruits. We have used fruits such as Apple (along with its varieties), Banana, papaya (with good and diseased type of papayas) etc. for our study. Using Ubuntu OS, the training has been done in HP core i5 processor based laptop.

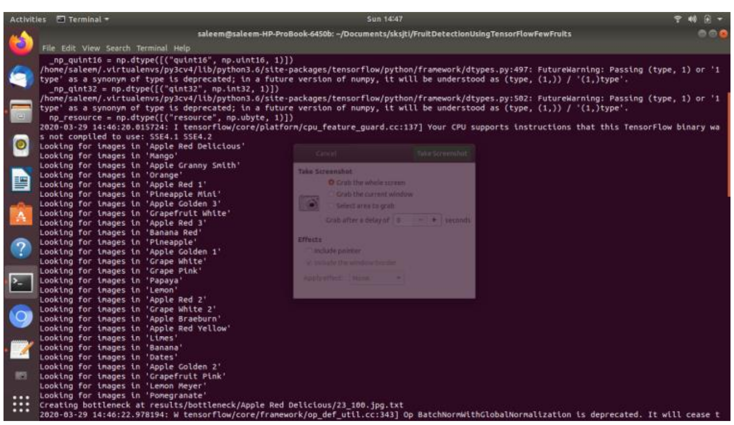

Figure 5: *Depicts the training process start – Capturing the fruits names for training*

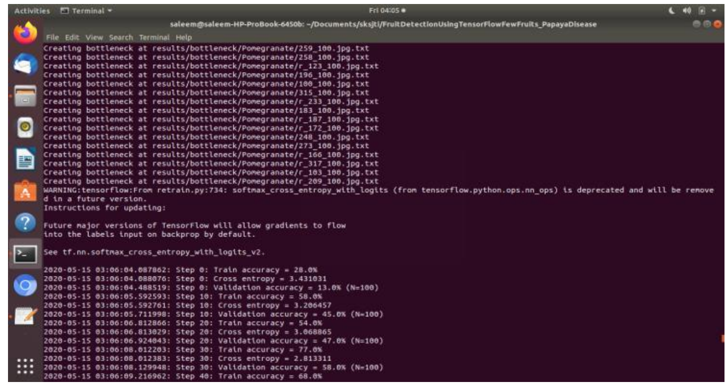

Figure 6 : *Depicts the training process capturing the pomegranate fruit details to train*

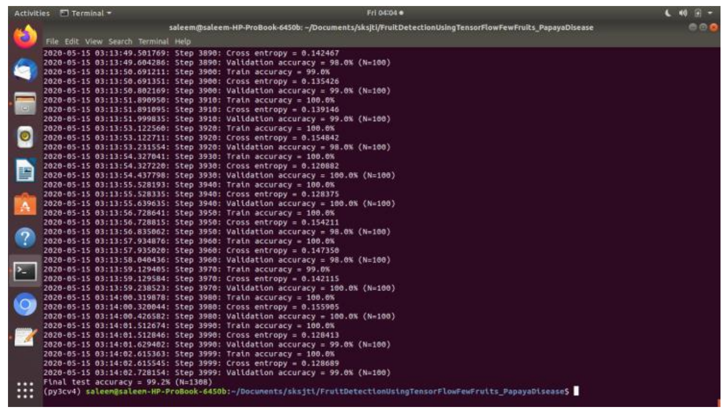

Figure 7: *Depicts the training process completion*

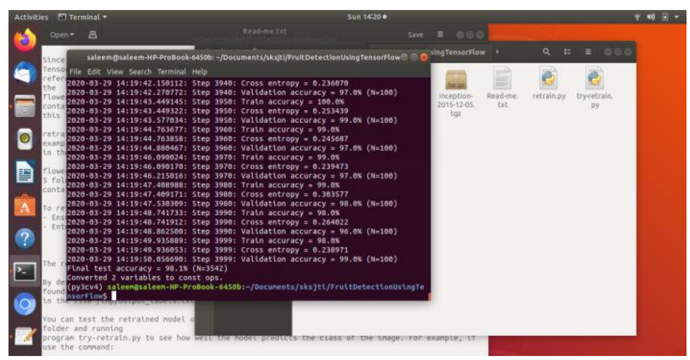

Figure 8: *Depicts the training process validation accuracy for detection*

E. Flow chart

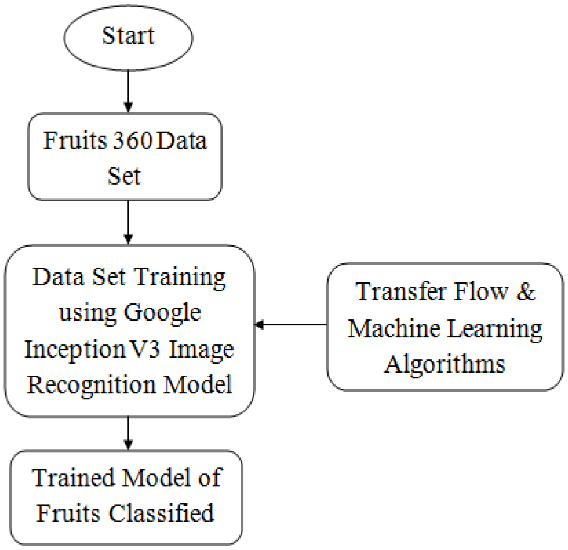

Figure 9: *Flow chart for training of the model using TF & ML algorithms*

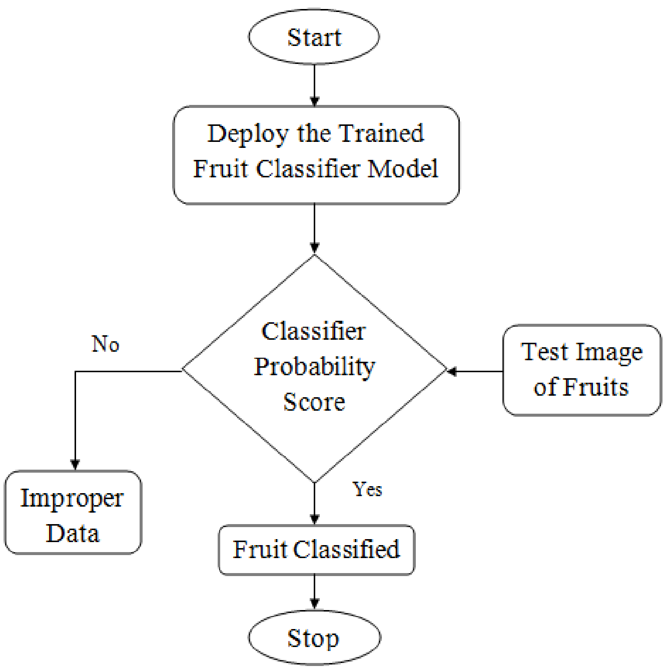

Figure 10: *Flow chart after training the model for estimation of fruits*

F. Results Analysis:

Once the training process is completed, the trained model is used to detect the fruits, its detection and quality based on the accuracy score. Below results has been obtained by using the trained model tested in raspberry Pi.

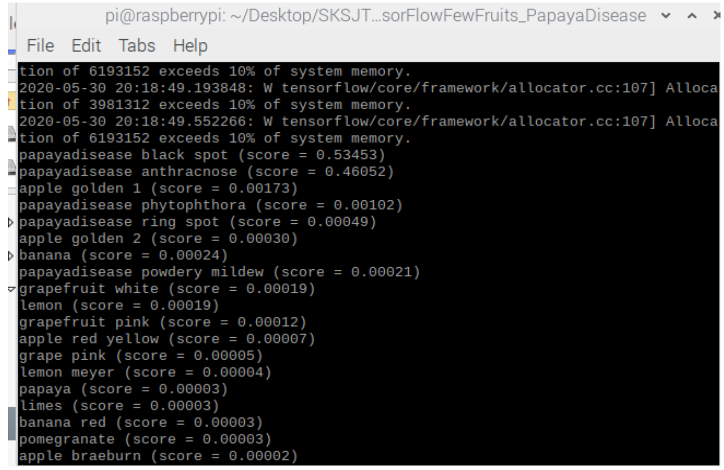

Figure 11: *Execution of democode to illustrate the identification score for Papaya Disease*

Figure 12: *Estimated image of Papaya fruit*

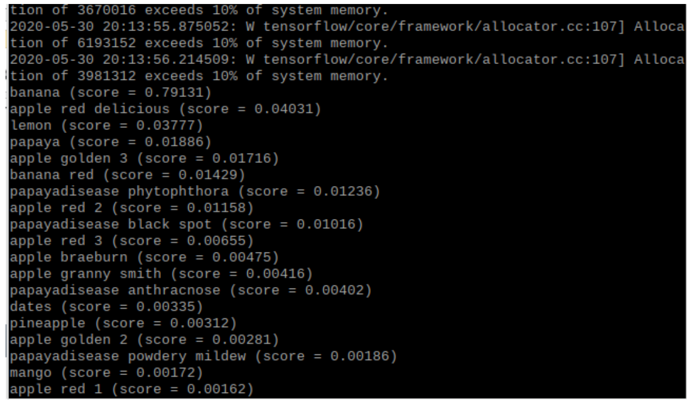

Figure 13: *Execution of democode to illustrate the identification score for banana*

Figure 14: *Estimated image of banana fruit*

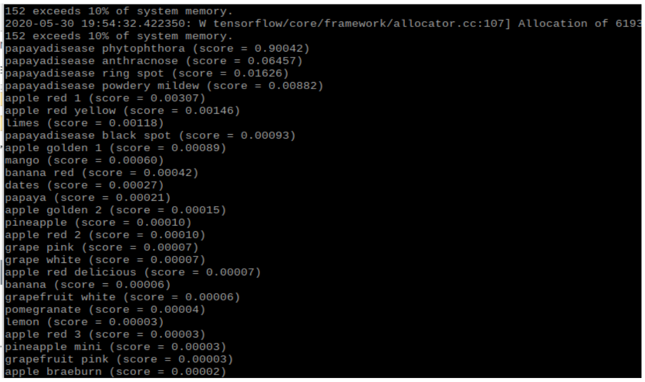

Figure 15: *Execution of democode to illustrate the identification score for Papaya Disease Phytophthora*

Figure 16: *Estimated image of Papaya fruit infected with Papaya disease Phytophthora*

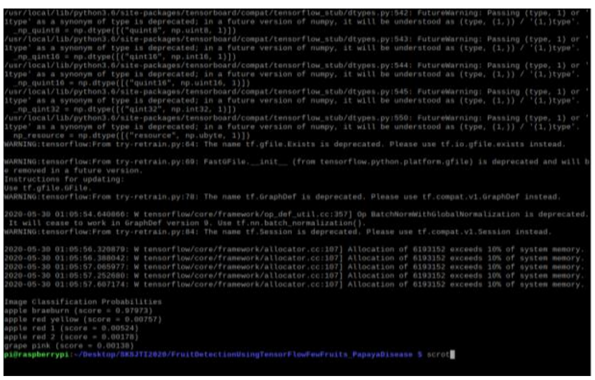

Figure 17: *Execution of democode to illustrate the identification score for Apple Braeburn*

Figure 18: *Estimated image of Apple Bareburn fruit*

During evaluation of the fruit detection, by installing the LAMP webserver in the laptop, an effort to make the execution and detection of the fruits remote, we have implemented the execution of the tasks using a webpage using any browser. To achieve this TTY terminal has been installed and to access remotely this has to be up & running in the webserver.

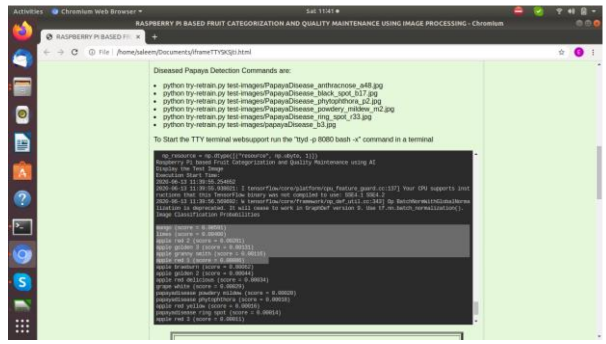

Figure 19: *Execution through webpage terminal to access the system remotely*

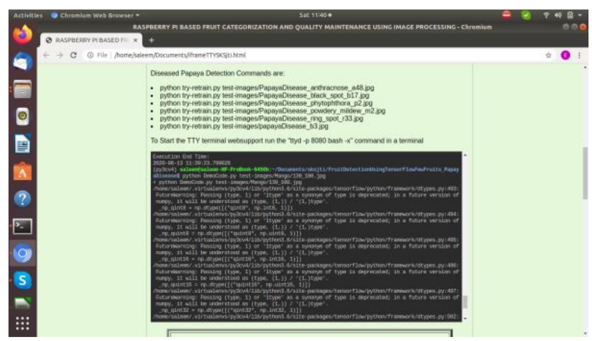

Figure 20: *Execution through webpage terminal to access the system & results remotely*

V. RESULT EVALUATION

In this section we will discuss about the implementation and results discussion for the fruit detection and categorization using Raspberry Pi. The result has been analyzed below for all the test images taken for testing from Fruits 360 dataset [19]. The Image classification probability score has been calculated & listed with the maximum probability of detection & minimum probability of detection score. In Table I the numerical results have been tabulated with the identification score for top 7 fruit scores. From the probability scores we can observe how the machine learning algorithm is able to identify the fruit. The study of a particular fruit with the quality, if it's infected by any disease can be easily done. Here we have studied about Papaya fruit with its disease types such as phytophthora, anthracnose, ring spot and powdery mildew diseases.

A. Numerical Results Analysis:

TABLE I. *NUMERICAL RESULTS*

SL NO	Testing Fruit Path	Fruit Name	Image Classification Probability in Corei5 Processor
1	test-images/21_100.jpg	Banana	banana (score = 0.79131) apple red delicious (score = 0.04031) lemon (score = 0.03777) papaya (score = 0.01886) apple golden 3 (score = 0.01716) banana red (score = 0.01429) papaya disease phytophthora (score = 0.01236)
2	test-images/34_100.jpg	Apple	apple braeburn (score = 0.97973) apple red yellow (score = 0.00757) apple red 1 (score = 0.00524) apple red 2 (score = 0.00178) grape pink (score = 0.00138) apple red delicious (score = 0.00108) mango (score = 0.00089)
3	test-images/132_100.jpg	Mango	mango (score = 0.97582) limes (score = 0.00912) apple red 2 (score = 0.00290) apple red 1 (score = 0.00235) apple braeburn (score = 0.00209) apple red delicious (score = 0.00122)
4	test-images/Papaya Disease_anthracnose_a48.jpg	Papaya	Papaya disease anthracnose (score = 0.52189) papaya disease phytophthora (score = 0.19532) papaya disease ring spot (score = 0.15833) papaya disease powdery mildew (score = 0.05108) apple red yellow (score = 0.02951) apple red 1 (score = 0.01088)
5	test-images/papaya Disease_b3.jpg	Papaya	Papaya disease black spot (score = 0.83005) papaya disease ring spot (score = 0.04131) papaya disease phytophthora (score = 0.03708) papaya disease powdery mildew (score = 0.03179) apple red delicious (score = 0.01354) papaya disease anthracnose (score = 0.01162) mango (score = 0.00596)
6	test-images/Papaya Disease_black_spot_b17.jpg	Papaya	Papaya disease black spot (score = 0.53453) papaya disease anthracnose (score = 0.46052) apple golden 1 (score = 0.00173) papaya disease phytophthora (score = 0.00102) papaya disease ring spot (score = 0.00049) apple golden 2 (score = 0.00030) banana (score = 0.00024)
7	test-images/Papaya Disease_phytophthora_p2.jpg	Papaya	Papaya disease phytophthora (score = 0.90042) papaya disease anthracnose (score = 0.06457) papaya disease ring spot (score = 0.01626) papaya disease powdery mildew (score = 0.00882) apple red 1 (score = 0.00307) apple red yellow (score = 0.00146) limes (score = 0.00118)
8	test-images/Papaya Disease_powdery_mildew_m2.jpg	Papaya	Papaya disease powdery mildew (score = 0.86816) papaya disease ring spot (score = 0.07176) papaya disease anthracnose (score = 0.02720) papaya disease phytophthora (score = 0.01889) papaya disease black spot (score = 0.00511) apple red yellow (score = 0.00124) apple red 1 (score = 0.00117)
9	test-images/Papaya Disease_ring_spot_r33.jpg	Papaya	Papaya disease ring spot (score = 0.94939) papaya disease phytophthora (score = 0.02983) papaya disease powdery mildew (score = 0.00771) papaya disease black spot (score = 0.00753) papaya disease anthracnose (score = 0.00246) apple red yellow (score = 0.00121) limes (score = 0.00026)
10	test-images/pomegranate.jpg	Pomegranate	pomegranate (score = 0.97982) apple red 3 (score = 0.00863) papaya disease black spot (score = 0.00516) papaya disease anthracnose (score = 0.00369) apple red 1 (score = 0.00113) apple red 2 (score = 0.00047) grapefruit pink (score = 0.00023)
11	test-images/Apple_Braeburn/49_100.jpg	Apple	apple brae burn (score = 0.95945) apple red yellow (score = 0.01188) apple red 1 (score = 0.00679) grape pink (score = 0.00428) apple red 2 (score = 0.00424) apple red delicious (score = 0.00364) mango (score = 0.00259)
15	test-images/Apple_Golden_1/42_100.jpg	Apple	apple golden 1 (score = 0.96598) grapefruit white (score = 0.01284) lemon (score = 0.00842) apple golden 3 (score = 0.00507) apple golden 2 (score = 0.00268) papaya (score = 0.00188) papaya disease anthracnose (score = 0.00106)
20	test-images/Apple_Golden_2/r_44_100.jpg	Apple	apple golden 2 (score = 0.94423) apple golden 3 (score = 0.02116) apple granny smith (score = 0.00728) mango (score = 0.00675) grape white (score = 0.00545) apple red 2 (score = 0.00448)

25	test-images/Apple_Golden_3/r_87_100.jpg	Apple	apple golden 3 (score = 0.80218) apple golden 2 (score = 0.06571) limes (score = 0.01991) papaya (score = 0.01865) apple golden 1 (score = 0.01496)
27	test-images/Apple_Granny_Smith/78_100.jpg	Apple	apple granny smith (score = 0.98065) apple golden 3 (score = 0.00872) apple red 2 (score = 0.00316) grape white 2 (score = 0.00258) mango (score = 0.00256)
42	test-images/Apple_Red_Delicious/138_100.jpg	Apple	apple red delicious (score = 0.99017) apple red 1 (score = 0.00256) grape pink (score = 0.00193) apple brae burn (score = 0.00137) apple red 2 (score = 0.00070)
50	test-images/Banana/86_100.jpg	Banana	banana (score = 0.99358) banana red (score = 0.00195) papaya disease black spot (score = 0.00150) papaya disease anthracnose (score = 0.00095) orange (score = 0.00047)
58	test-images/Dates/95_100.jpg	Dates	dates (score = 0.98864) banana red (score = 0.00539) papaya disease ring spot (score = 0.00122) banana (score = 0.00110) apple red yellow (score = 0.00074)
68	test-images/Grape_Pink/r_3_100.jpg	Grapefruit	grape pink (score = 0.99402) apple red yellow (score = 0.00181) apple red 1 (score = 0.00099) grape white (score = 0.00090) pomegranate (score = 0.00062)
72	test-images/Lemon/r_15_100.jpg	Lemon	lemon (score = 0.98396) lemon meyer (score = 0.00719) grapefruit white (score = 0.00601) apple golden 1 (score = 0.00111) papaya (score = 0.00083)
78	test-images/Mango/139_100.jpg	Mango	mango (score = 0.98591) limes (score = 0.00460) apple red 2 (score = 0.00281) apple golden 3 (score = 0.00131) apple granny smith (score = 0.00116) apple red 1 (score = 0.00086)
85	test-images/Orange/r_325_100.jpg	Orange	orange (score = 0.89184) lemon meyer (score = 0.06077) grapefruit pink (score = 0.02753) grapefruit white (score = 0.01610) lemon (score = 0.00140)
88	test-images/Papaya/r_9_100.jpg	Papaya	papaya (score = 0.99947) apple red 1 (score = 0.00013) banana red (score = 0.00009) grape white (score = 0.00005) grape pink (score = 0.00005)
90	test-images/Pineapple/99_100.jpg	Pineapple	pineapple (score = 0.95207) pineapple mini (score = 0.04243) apple red 3 (score = 0.00199) papaya (score = 0.00044) limes (score = 0.00037)
97	test-images/Pomegranate/77_100.jpg	Pomegranate	pomegranate (score = 0.97406) apple red 3 (score = 0.01454) apple red 1 (score = 0.00539) apple red 2 (score = 0.00258)

ADVANTAGES AND APPLICATIONS

The main advantages and applications for the discussed proposed paper has been listed below.
Advantages are:

- ➢ High speed, high measuring and grading accuracy with low cost can be achieved.
- ➢ Pollution due to wastes can be reduced.
- ➢ Any person consuming the edible fruit is aware of the quality of fruits before the usage.
- ➢ It can easily check the freshness of any edible fruit.
- ➢ The Automatic handling mechanism could reduce human involvement and perform the inspections objectively and with faster speed.

Applications are:
We can write plenty of applications for the proposed system, lets list a few below.

- ➢ The devised prototype of the project can be helpful in packaging of the fruits.
- ➢ Helps in detection of the Quality while packaging.
- ➢ Can be used by farmers and fruit merchants to categorize the fruits through automation.
- ➢ Automatic Fruit Counting Vehicle can be designed.
- ➢ Can be used in Automated Harvesting
- ➢ Overlapping Fruits Recognition can be implemented
- ➢ Defects recognition in fruits can be easily achieved.

CONCLUSION

With the use of Machine Learning (ML) and deep learning concepts, we have used intelligently the concept of transfer learning for the development of the paper. An already trained inception model from Google has been reused to train out fruits data set obtained from existing verified data set and after the training, using the python language the results has been studied. Tensor flow framework looks to be a perfect open source framework to work on machine learning algorithms & techniques. The proposed project can be developed using Arduino as well by using the already developed TinyML library to detect fruits using the sensors embedded in the Arduino Nano 33 BLE sense model but we have used the raspberry Pi as an enhancement. The numerical Results are found to be in good position to conclude that the trained model is able to detect the fruits types, its quality & the diseased fruit among the selection of the fruits in both laptop environment as well as Raspberry Pi.

REFERENCES

[1]. S. Bargoti, J. Underwood, Deep fruit detection in orchards, IEEE International Conference on Robotics and Automation (ICRA), pp. 3626-3633, 2017

[2]. https://blog.arduino.cc/2019/11/07/fruit-identification-using-arduino-and-tensorflow/
[3]. export.arxiv.org
[4]. Fruits 360 Dataset on GitHub. https://github.com/Horea94/Fruit-Images-Dataset.
[5]. R. Barth, J. Ijsselmuiden, J. Hemming, E. Van Henten, Data synthesis methods for semantic segmentation in agriculture: A Capsicum annuum dataset, Computers and Electronics in Agriculture, 144, pp.284-296, 2018
[6]. Krithika Jayasankar, Karthika B, Jeyashree T, Deepalakshmi R, Karthika G, “FRUIT FRESHNESS DETECTION USING RASPBERRY PI”, International Journal of Pure and Applied Mathematics. Volume 119 No. 15 2018, 1685-1691 ISSN: 1314-3395 (on-line version)
[7]. H. Cheng, L. Damerow, Y. Sun, M. Blanke, Early Yield Prediction Using Image Analysis of Apple Fruit and Tree Canopy Features with Neural Networks, Journal of Imaging, Vol. 3(1), 2017.
[8]. Fruits 360 Dataset on Kaggle. https://www.kaggle.com/moltean/fruits.
[9]. Horea Muresan,Mihai Oltean,”Fruit recognition from images using deep learning”,ACTA UNIV. SAPIENTIAE, INFORMATICA, 10, 1 (2018) 26–42
[10]. D. Clevert, T. Unterthiner, S. Hochreiter, Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs) CoRR abs/1511.07289, 2015
[11]. K. Kapach, E. Barnea, R. Mairon, Y. Edan, O. Ben-Shahar, Computer vision for fruit harvesting robots state of the art and challenges ahead, Journal of Imaging, Vol. 3(1), pp. 4-34, 2017.
[12]. M. Liang, X. Hu, Recurrent Convolutional Neural Network for Object Recognition, IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Boston, pp. 3367-3375, 2015.
[13]. S. Puttemans, Y. Vanbrabant, L. Tits, T. Goedem, Automated visual fruit detection for harvest estimation and robotic harvesting, Sixth International Conference on Image Processing Theory, Tools and Applications, 2016
[14]. M, Rahnemoonfar, C. Sheppard, Deep count: fruit counting based on deep simulated learning, Sensors, 17(4), p. 905-, 2017.
[15]. S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, In Advances in neural information processing systems, pp. 91-99, 2015.
[16]. I. Sa, Z. Ge, F. Dayoub, B. Upcroft, T. Perez & C. McCool, Deep- Fruits: A Fruit Detection System Using Deep Neural Networks, Sensors (Basel, Switzerland), Vol. 16(8), pp. 1222-, 2016.
[17]. J. Schmidhuber, Deep learning in neural networks: An overview, Neural Networks vol. 61, pp. 85-117, 2015
[18]. R. K. Srivastava, K. Greff, J. Schmidhuber, Training very deep networks, Advances in neural information processing systems, Twenty- Eight International Conference on Neural Information Processing Systems,pp. 2377-2385, 2015, Montreal, Canada, 2015
[19]. J. Xiong, Z. Liu, R. Lin, R. Bu, Z. He, Z. Yang, C. Liang , Green Grape Detection and Picking-Point Calculation in a Night-Time Natural Environment Using a Charge-Coupled Device (CCD) Vision Sensor with Artificial Illumination Sensors, Vol. 18(4), pp. 969-, 2018.
[20]. D.C. Cires¸an, U. Meier, J. Schmidhuber, Multi-column Deep Neural Networks for Image Classification, IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Providence, pp. 3642-3649, 2012.
[21]. Convolution in mathematics https://en.wikipedia.org/wiki/Convolution.
[22]. Deep Learning article on Wikipedia. https://en.wikipedia.org/wiki/Deep_learning.
[23]. TensorFlow. https://www.tensorflow.org.
[24]. MNIST. http://yann.lecun.com/exdb/mnist.
[25]. D. Li, H. Zhao, X. Zhao, Q. Gao, L. Xu, Cucumber Detection Based on Texture and Color in Greenhouse, International Journal of Pattern Recognition and Artificial Intelligence, Vol. 31 (08), August 2017
[26]. CIFAR-10 and CIFAR-100 Datasets. https://www.cs.toronto.edu/˜kriz/cifar.html.
[27]. Britta O’Boyle and Chris Hall What is Google Lens and how do you use it?.
[28]. Google Lens on Wikipedia, https://en:wikipedia:org=wiki=Google Lens
[29]. https://www.tutorialspoint.com/tensorflow/index.htm
[30]. http://www.helpshareideas.com
[31]. www.researchgate.net
[32]. codelabs.developers.google.com