FRUIT QUALITY ASSESSMENT AND SORTING SYSTEM

Mrs. Uzma I Satteekar, Ms. Tarannum B Kodekar, Ms. Swaleha R Kolhar

Assistant Professor, student, student

Department of Electronics and Communication Engineering,
SECAB INSTITUTE OF ENGINEERING AND TECHNOLOGY
VIJAYAPUR-586101, INDIA.

Abstract: Grading and classification of fruits is based on observations and through experiences. The system utilizes image-processing techniques to classify and grade quality of fruits. Two-dimensional fruit images are classified on shape and colour based analysis methods. However, different fruit images may have similar or identical colour and shape values. Hence, using colour or shape features analysis methods are still not effective enough to identify and distinguish fruits images. Therefore, we used a method to increase the accuracy of the fruit quality detection by using colour, shape, and size-based method with a combination of artificial neural network (ANN). Proposed method grades and classifies fruit images based on obtained feature values by using cascaded forward network. The proposed system starts the process by capturing the fruit's image. Then, the image is transmitted to the processing level where the fruit features like colour, shape and size of fruit samples are extracted. After that, by using artificial neural network, fruit images are going through the training and testing. In this proposed paper neural network is used to detect shape, size and colour of fruit and with the combination of these three features, the results obtained are very promising.

Index Terms **– ANN, Tensor Flow, Sorting, Grading, Fruit images.**

Introduction

India is a country of agricultural lands and fields. Different types of fruits and Vegetables are produced in India. India is at second number after china in producing fruits. In India. all the pre-harvest and post-harvest processes are done manually with the help of labor. Manual process is very time consuming, less efficient. To get accurate results, automation in agriculture industry is needed. The post-harvest process includes sorting and grading of fruits. Different quality factors are considered for sorting and grading of fruits. These factors are internal quality factors and external quality factors. The external quality factors are texture, shape, color ,size and volume and internal quality factors are test ,sweetness, flavor, aroma, nutrients , carbohydrates present in that fruit[3]. Automation is playing a vital role in day today life. In India, more than half population depends upon agriculture. Their main source of income is agriculture. Exporting of fresh fruits is getting increased day by day from India. People are very conscious about their health; they prefer only fresh, good quality eatables.

CNN TRANSFER LEARNING DEVELOPMENT

2.1 CNN (Convolution Neural Network)

Convolutional neural network (CNN), a class of artificial neural networks that has become dominant in various computer vision tasks. A tremendous interest in deep learning has emerged in recent years. The most established algorithm among various deep learning models is convolutional neural network (CNN), a class of artificial neural networks that has been a dominant method in computer vision tasks since the astonishing results were shared on the object recognition competition known as the Image Net Large Scale Visual Recognition Competition (ILSVRC) in 2012. Medical research is no exception, as CNN has achieved expert-level performances in various fields.

Deep learning especially CNN is a prominent architecture that can extract features of the images and classify respectively. Many researches on medical image analysis applying CNN. Grading nuclei of hepatocellular carcinoma have been done successfully using CNN. Mitotic and non-mitotic nuclei in breast cancer histopathological images can be classified by 2-phase and result in 0.79 f-score outperform handcrafted feature extraction. CNN based on CAD for segmenting adipose tissue volume has applied and reveals high accuracy up to 96.8%. LeNet architecture of CNN model was used in the research here. Comparison between CNNs and massive training artificial neural network also conducted for lung nodule detection and classification. The challenges of automatic detection of histopathological images are a high variety of images, special texture or mimic on images and the deformation of the images. With the ability of as generic feature extractor, CNN can be enabled as an approach.

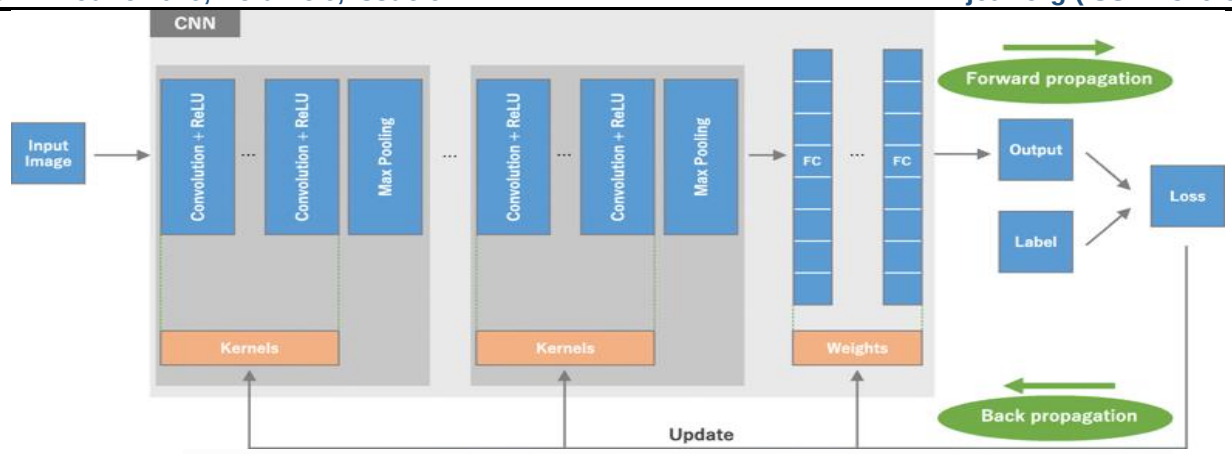

Fig 2.1 An overview of a convolutional neural network (CNN) architecture and the training process

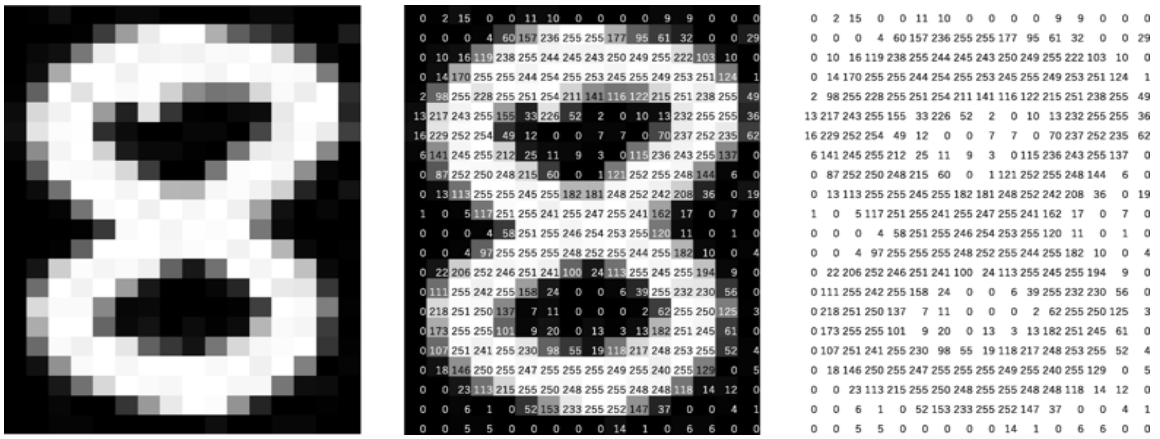

Fig 2.2 A computer sees an image as an array of numbers. The matrix on the right contains numbers between 0 and 255, each of which corresponds to the pixel brightness in the left image. Both are overlaid in the middle image.

Convolutional Neural Networks (CNN) have completely dominated the machine vision space in recent years. A CNN consists of an input layer, output layer, as well as multiple hidden layers. The hidden layers of a CNN typically consist of convolutional layers, pooling layers, fully connected layers and normalization layers (ReLU). Additional layers can be used for more complex models. Examples of a typical CNN can be seen in and it is depicted in Figure 2.3.

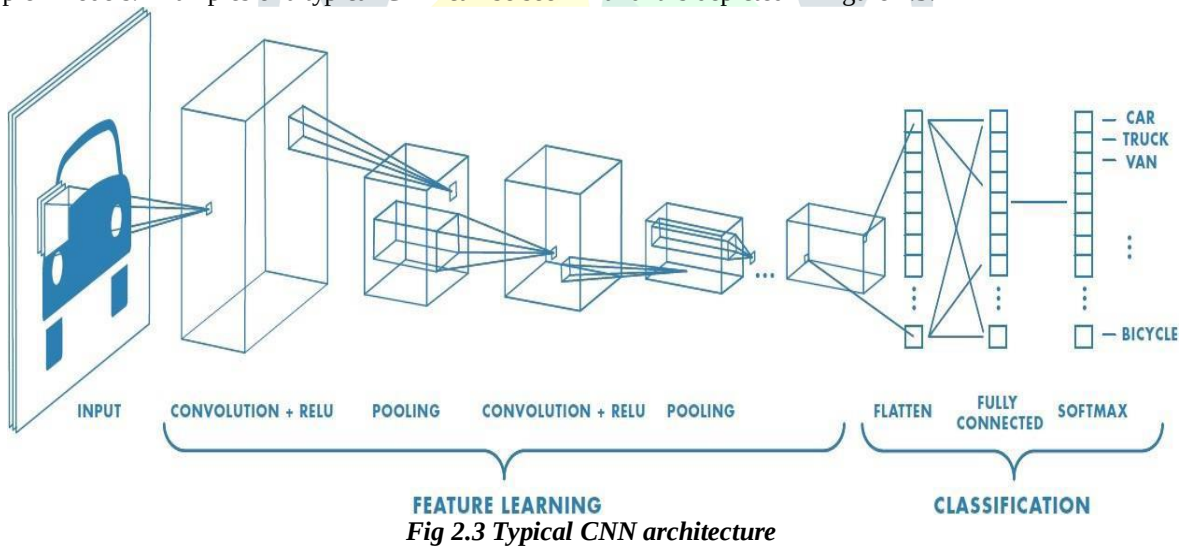

Fig 2.3 Typical CNN architecture

The above figure2.3 shows the CNN architecture, it has shown excellent performance in many Computer Vision and Machine Learning problems. CNN trains and predicts in an abstract level, with the details left out for later sections. This CNN model is used extensively in modern Machine Learning applications due to its ongoing record-breaking effectiveness. Linear algebra is the basis for how these CNNs work. Matrix vector multiplication is at the heart of how data and weights are represented. Each of the layers contains a different set of characteristics for an image set. For instance, if a face image is the input into a CNN, the network will learn some basic characteristics such as edges, bright spots, dark spots, shapes etc., in its initial layers. The next set of layers will consist of shapes and objects relating to the image which are recognizable such as: eyes, nose and mouth. The subsequent layer consists of aspects that look like actual faces, in other words, shapes and objects which the network can use to define a human face. CNN matches parts rather than the whole image, therefore breaking the image classification process down into smaller parts (features). A 3x3 grid is defined to represent the features extraction by the CNN for evaluation. The following process, known as filtering, involves lining the feature with the image patch. One-by-one, each pixel is multiplied by the corresponding feature pixel, and once completed, all the values are summed and divided by the total number of pixels in the feature space. The final value for

the feature is then placed into the feature patch. This process is repeated for the remaining feature patches followed by trying every possible match- repeated application of this filter, which is known as a convolution.

The next layer of a CNN is referred to as "max pooling", which involves shrinking the image stack. In order to pool an image, the window size must be defined (e.g. usually 2x2/3x3 pixels), the stride must also be defined (e.g. usually 2 pixels). The window is then filtered across the image in strides, with the max value being recorded for each window. Max pooling reduces the dimensionality of each feature map whilst retaining the most important information. The normalisation layer of a CNN, also referred to as the process of Rectified Linear Unit (ReLU), involves changing all negative values within the filtered image to 0. This step is then repeated on all the filtered images, the ReLU layer increases the non-linear properties of the model. The subsequent step by the CNN is to stack the layers (convolution, pooling, ReLU), so that the output of one layer becomes the input of the next. Layers can be repeated resulting in a "deep stacking". The final layer within the CNN architecture is called the fully connected layer also known as the classifier. Within this layer every value gets a vote on determining the image classification. Fully connected layers are often stacked together, with each intermediate layer voting on phantom "hidden" categories. In effect, each additional layer allows the network to learn even more sophisticated combinations of features towards better decision making [6]. The values used for the convolution layer as well as the weights for the fully connected layers are obtained through backpropagation, which is done by the deep neural network. Backpropagation is whereby the neural network uses the error in the final answer to determine how much the network adjusts and changes.

The Inception-v3 model is an architecture of convolutional networks. It is one of the most accurate models in its field for image classification, achieving 3.46% in terms of "top-5 error rate" having been trained on the ImageNet dataset [7]. Originally created by the Google Brain team, this model has been used for different tasks such as object detection as well as other domains through Transfer Learning.

The CNN learning process can rely on vector calculus and chain rule. Let z be a scalar (i.e., $z \in \mathbf{R}$) and $y \in \mathbf{R}$H be a vector. So, if z is a function of y, then the partial derivative of z with respect to y is a vector, defined as:

$$(\partial z\ \partial y\ )i = \partial z\ \partial yi. \quad (1)$$

Specifically, $(\partial z\ \partial y\ )$ is a vector having the same size as y, and its i-th element is $(\partial z\ \partial y\ )i$. Also note that $(\partial z\ \partial yT\ ) = (\partial z\ \partial y)T$. Furthermore, presume $x \in$ RW is another vector, and y is a function of x. Then, the partial derivative of y with respect to x is defined as:

$$(\partial y\ \partial xT)ij = \partial yi\ \partial xi. \quad (2)$$

This fractional derivative is a H ×W matrix, whose entry at the juncture of the i-th row and j-th column is $\partial yi\ \partial xi$. It is easy to see that z is a function of x in a chain-like argument: a function maps x to y, and another function maps y to z. A chain rule can be used to compute:

$$(\partial z\ \partial xT\ ), \text{ as } (\partial z\ \partial xT\ ) = (\partial z\ \partial yT\ )(\partial y\ \partial xT\ ). \quad (3)$$

One can use a cost or loss function to measure the discrepancy between the prediction of a CNN xL and the target t, $x1 \to w1, x2 \to \cdots, xL \to wL = z$, using a simplistic loss function $z = \|t - xL\|2$. However more complex functions are usually employed. A prediction output can be seen as argmax$xixiL$. The convolution procedure can be expressed as:

$$yil+1,jl+1,d = \Sigma\Sigma\Sigma fi.j.dDd=0Wj=0Hi=0 \times xil+1+,i,jl+1+j,dL . \quad (4)$$

The filter f has size $(H \times W \times Dl)$, thus the convolution will have the spatial size of $(Hl - H+1) \times (Wl - W+1)$ with D slices, which means $y\ (xl+1)$ in R$Hl+1 \times Wl+1 \times Dl+1$, $Hl+1 = Hl - H+1, Wl+1 = Wl - W+1, Dl+1 = D$.

When it comes to Inception V3, the probability of each label $k \in \{1, \ldots K\}$ for each training example is computed by $P(k|x) = \exp(zk)\Sigma\exp(zi)Ki$, where z is a non-normalized log probability. Ground truth distribution over labels $q(k \mid x)$ is normalized, so that $\Sigma q(k|x) = 1.k$ For this model, the loss is given by cross-entropy:

$$\ell = \Sigma\log Kk = 1(p(k))q(k). \quad (5)$$

Cross-entropy loss is differentiable with respect to the logits zk and thus it can be used for gradient training of deep models, where the gradients has the simple form $\partial\ell\partial zk = p(k) - q(k)$, bounded between -1 and 1. Usually, when minimizing the cross entropy, it means that log-likelihood of the correct label is maximized. Since it may cause some overfitting problems, Inception V3 considers a distribution over labels independent of training examples $u(k)$ with a smooth parameter ϵ, where for a training example, the label distribution $q(k \mid x) = \delta k,y$ is simply replaced by:

$$q'(k|x) = (1 - \epsilon)\delta k,x + \epsilon u(k), \quad (6)$$

which is a mixture of the original distribution $q(k \mid x)$ with weights 1- ϵ and the fixed distribution $u(k)$ with weights ϵ. A label-smoothing regularization is applied, with uniform distribution $u(k) = 1/K$, so that it becomes:

$$q'(k|x) = (1 - \epsilon)\delta k,x + \epsilon K . \quad (7)$$

Alternatively, this can be interpreted as cross-entropy as follows:

$$H(q',p) = -\Sigma\log Kk = 1(p(k))q'(k) = (1 - \epsilon)H(q',p) + \epsilon H(u,p). \quad (8)$$

Therefore, the label-smoothing regularization is similar to applying a single cross-entropy loss $H(q, p)$ with a pair of losses $H(q, p)$ and $H(u, p)$, with the second loss penalizing the deviation of the predicted label distribution p from the prior u with relative weight $\epsilon(1 - \epsilon)$, which is equivalent to computing the Kullback–Leibler divergence. More details (step-by-step) and the mathematical formulation of CNN and Inception V3 can be found in and. In this work we aim to retrain this model on a new dataset and study the results. Choosing not to train the model from scratch as it would be computationally intensive task and depending on the computing setup, may take several days or even weeks. In addition, it would also require multiple GPUs and/or multiple machines. Instead, we will be comparing results obtained from the proposed retrained model to that of related works to prove our hypothesis.

2.2 Transfer learning

Transfer Learning is a Machine Learning technique whereby a model is trained and developed for one task and is then re-used on a second related task. It refers to the situation whereby what has been learnt in one setting is exploited to improve optimization in another setting [8]. Transfer Learning is usually applied when there is a new dataset smaller than the original dataset used to train the pre-trained model.

This paper proposes a system which uses a model (Inception-v3) in which was first trained on a base dataset (ImageNet), and is now being repurposed to learn features (or transfer them), to be trained on a new dataset (CIFAR-10 and Caltech Faces). With regards to the initial training, Transfer Learning allows us to start with the learned features on the ImageNet dataset and adjust these

features and perhaps the structure of the model to suit the new dataset/task instead of starting the learning process on the data from scratch with random weight initialization. TensorFlow is used to facilitate Transfer Learning of the CNN pre-trained model. We study the topology of the CNN architecture to find a suitable model, permitting image classification through Transfer Learning. Whilst testing and changing the network topology (i.e. parameters) as well as dataset characteristic to help determine which variables affect classification accuracy, though with limited computational power and time.

LITERATURE SURVEY

1. **SooHo Jeong, Hyun Yoe (2018)**
   The system proposed in this study is a Deep Learning based fruit quality classification system that can be used in small farms. Recently, Deep Learning technology for the implementation of artificial intelligence is rapidly developing as the field of artificial intelligence advances. Currently, libraries using various algorithms are open sourced to utilize Deep Learning.
2. **Nagganaur and Sannanki (2015)**
   Presented the sorting and grading of fruits using image processing techniques. The system starts the process by capturing the fruit's image. Then the image is transmitted to the matlab for feature extraction, classification and grading. Both classification and grading realized by fuzzy logic approach.
3. **H. Alimohamadi *et al* (2013)**
   Designed a system for skin defect detection in fruits. Gabor wavelet Filter is used for defect detection. Convert colour image into texture image and then on that image Bank of Gabor filter is applied. Gabor Filter is linear filter and used as edge detector. Gabor filters with 4 scales and 6 rotations used in this paper. Obtained response shows image pixel is as defected or normal skin. Optimal filter is chosen from bank of Gabor filters depending upon the response. Thresholding the response of the optimal filter. Based on thresholding skin defect is detected.

INTRODUCTION TO TENSOR FLOW

4.1 What is TensorFlow?

Currently, the most famous deep learning library in the world is Google's TensorFlow. Google product uses machine learning in all of its products to improve the search engine, translation, image captioning or recommendations.

To give a concrete example, Google users can experience a faster and more refined the search with AI. If the user types a keyword in the search bar, Google provides a recommendation about what could be the next word. Google wants to use machine learning to take advantage of their massive datasets to give users the best experience. Three different groups use machine learning:

- Researchers
- Data scientists
- Programmers.

They can all use the same toolset to collaborate with each other and improve their efficiency. Google does not just have any data; they have the world's most massive computer, so Tensor Flow was built to scale. TensorFlow is a library developed by the Google Brain Team to accelerate machine learning and deep neural network research. It was built to run on multiple CPUs or GPUs and even mobile operating systems, and it has several wrappers in several languages like Python, C++ or Java.

4.2 TensorFlow Architecture

Tensor flow architecture works in three parts:

- Preprocessing the data
- Build the model
- Train and estimate the model

It is called TensorFlow because it takes input as a multi-dimensional array, also known as **tensors**. You can construct a sort of **flowchart** of operations (called a Graph) that you want to perform on that input. The input goes in at one end, and then it flows through this system of multiple operations and comes out the other end as output. This is why it is called TensorFlow because the tensor goes in it flows through a list of operations, and then it comes out the other side.

4.3 Where can TensorFlow run?

TensorFlow can hardware, and software requirements can be classified into
Development Phase: This is when you train the mode. Training is usually done on your Desktop or laptop.
Run Phase or Inference Phase: Once training is done TensorFlow can be run on many different platforms. You can run it on

- Desktop running Windows, macOS or Linux
- Cloud as a web service
- Mobile devices like iOS and Android

You can train it on multiple machines then you can run it on a different machine, once you have the trained model.

The model can be trained and used on GPUs as well as CPUs. GPUs were initially designed for video games. In late 2010, Stanford researchers found that GPU was also very good at matrix operations and algebra so that it makes them very fast for doing these kinds of calculations. Deep learning relies on a lot of matrix multiplication. TensorFlow is very fast at computing the matrix multiplication because it is written in C++. Although it is implemented in C++, TensorFlow can be accessed and controlled by other languages mainly, Python. Finally, a significant feature of TensorFlow is the Tensor Board. The Tensor Board enables to monitor graphically and visually what TensorFlow is doing.

EXISTING SYSTEM

In our project all the drawbacks overcome in the literature survey. ARM Based Fruit Grading and Management System Using Image Processing in this project grading system is not automatic & color sensor is not available and ARM based project is very expensive. In our project grading system is automatic & less expensive. In Rapid Color Grading for Fruit Quality Evaluation, Using Direct Color Mapping project, there are only detect the color. And in our project detect the color, height & weight also. In Microcontroller and Arduino once a program is dumped then the program cannot be changed i.e. it is a static system. We are making an attempt in our project to make the system dynamic. We are using Raspberry Pi, which is much more advanced as compared to Microcontroller. It has the feature of Microcontroller along with additional features of computer like Ethernet, USB, HDMI interface.

OBJECTIVES

- To sort and grade fruits based on their colour, texture and size.
- To completely automate the above said system without the need of the human intelligence/intervention.
- To achieve efficient design of compact, realistic, affordable and standalone prototype.

METHODOLOGY

The program is dumped into the Raspbian OS is the OS involved. The purposed system starts the process by capturing the fruit image. Then the image is transmitted to the processing level in open CV where the fruit features like colour, shape and size of fruit samples are extracted. In the purposed paper open CV method is used to detect the shape, size and colour of fruit and with the combination of these three features the results obtained are very promising. Captured image is segmented using edge detection algorithms in order to find the detected fruit.

The Infrared sensor is placed on conveyor belt, when fruit comes in front of infrared sensor message will display as fruit detected then conveyor belt moves with small distance an stop when fruit come exactly in front of camera. Camera (High Quality CMOS sensor, 25 MPs, 30fps) always in video mode. When fruit is detected the image processing is done on that image captures and color is detected. Red, Green, Yellow color are detected. The system is divided into hardware control and image processing. The image processing results is based on camera image. The results such color detected. Second part is hardware is controlled based on color detection.

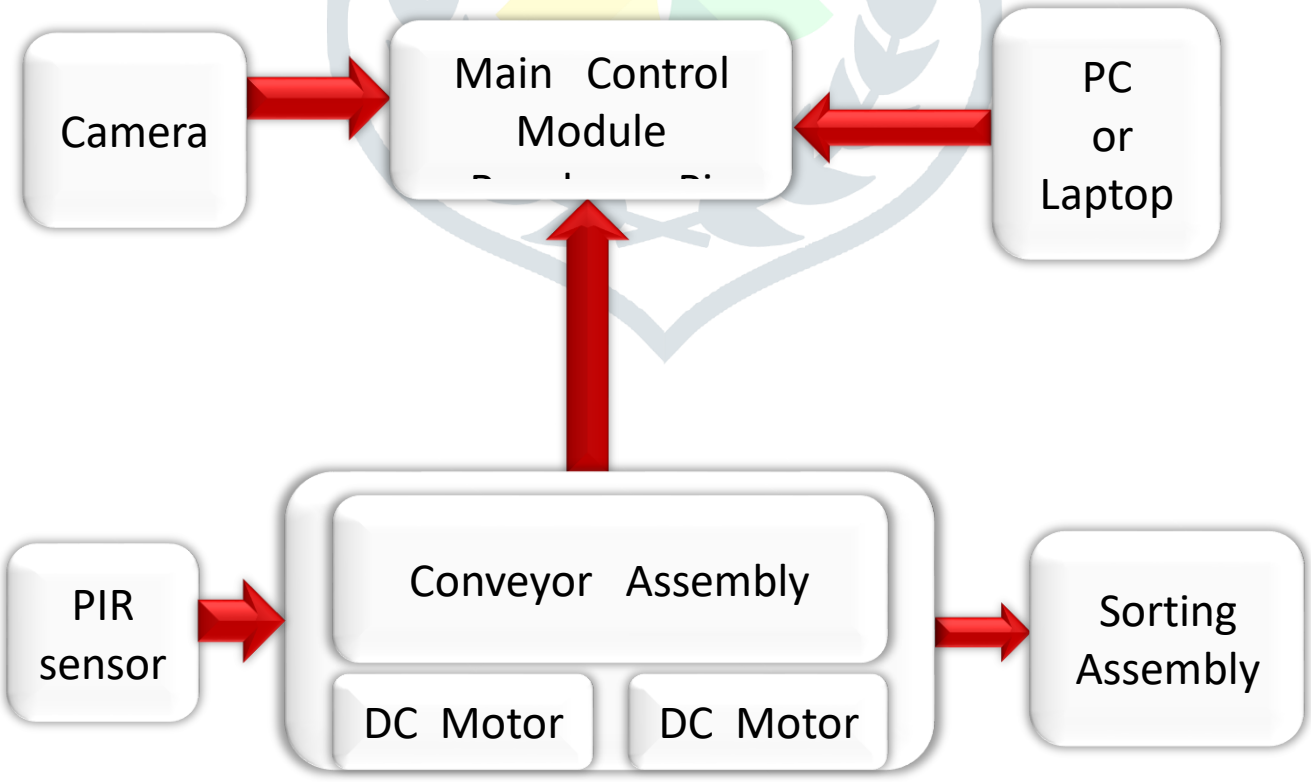

Fig 7 Block Diagram of fruit grading and sorting system

The image processing is done by software OpenCv using a language python. The software is divided into two parts first one is for image analysis and other is for controlling hardware based on image processing results. As per Fig 7 the system is operated

in two different scenarios in first the image is captured with camera the all the image processing is done in the control module. All the process are shown on monitor and then based on decision taken by control module. The conveyor assembly is operated.

HARDWARE DESCRIPTION

Hardware Required:

- Raspberry Pi 3 Model B
- DC Motor
- Conveyor Belt
- PIR Sensor
- Camera

8.1 Raspberry Pi 3 Model B:

Fig 8.1 Raspberry pi 3 model B

The above fig 8.1 shows the Raspberry pi 3 model B, it is an open-source, Linux based, credit card size minicomputer. There are two types of raspberry pi model A and B. It has 64-bit Quad core processor, and on-board Wi-Fi, Bluetooth and USB boot capabilities. It appeared with a faster 1.2Giga Hertz processor and a 3-time faster network based on Giga bit Ethernet. Other options are power over Ethernet, USB boot. It consist of 40pins GPIO(general purpose input output), 4x USB two ports, CSI camera port, DSI display port, micro SD card slot, HDMI(high definition multimedia interface), four pole stereo and composite video port.

8.2 DC MOTOR:

Fig8.2: DC Motor

The above fig 8.2 shows a DC motor, is an electric motor that runs on direct current power. Dc motor is a two-wire continuous rotation motor and the two wires are power and ground. When the supply is applied, a DC motor will start rotating until that power is detached. Most of the DC motor run at high revolutions per minute (RPM). The DC motor speed can be controlled by using PWM (pulse width modulation) technique, a technique of fast pulse the power ON & OFF. DC motors provide excellent speed control for acceleration and deceleration. Electric motors are used in vacuum cleaners, dishwashers, computer printers, fax machine, water pumping stations etc.

8.3 CONVEYOR BELT:

Fig 8.3: Conveyor Belt

The above figure 8.3 shows the Conveyor belt, it is one of the basic tools in material handling industry, belt conveyors are most commonly used in transportation of bulk materials. Belt conveyor systems consist of two or more pulleys. An endless loop of carrying medium, the conveyor belt rotates about them. To move the belt and the material it carries forward, one or both pulleys are powered. The powered pulley is called "drive pulley", the unpowered one is known as "idler pulley". Based on the purposed use, conveyor belts are manufactured using either PVC or rubber.

8.4 IR SENSOR:

Fig 8.4.1: IR Sensor

The above figure 8.4.1 shows the IR Sensor, the Sensors are basically electronic devices which are used to sense the changes that occur in their surroundings. The change may be in color, temperature, moisture, sound, heat etc. They sense the change and work accordingly. In IR sensor the there is emitter and detector. Emitter emits the IR rays and detector detects it.

8.5 Camera:

Fig 8.5: 25M USB Camera

The above figure 8.5 shows the USB Cameras are the imaging cameras that use USB 2.0 or USB 3.0 technology to transfer image data. USB cameras designed to easily interface with dedicated computers system by using the same USB technology that is found on most computers.

SOFTWARE DISCRIPTION

Software Required:

- Raspbian OS
- Python IDE

9.1 Raspbian OS

Raspbian is the Raspberry Pi's most popular operating system, a spin off of the Linux distribution Debian that works well on the Raspberry Pi's hardware. Raspbian is a competent and versatile operating system that gives your Raspberry Pi all the comforts of a PC: a command line, a browser, and tons of other programs, we can use a Raspberry Pi running Raspbian as a cheap and effective home computer. We can use it as a springboard and turn our Raspberry Pi into any of countless other functional devices, from wireless access points to retro gaming machines.

9.2 Python IDE

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales. Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object, imperative, functional and procedural, and has a large and comprehensive standard library.

Python interpreters are available for many operating systems. C Python, the reference implementation of Python, is source software and has a community-based development model, as do nearly all of its variant implementations. C Python is managed by the non-profit Python Software Foundation.

ADVANTAGES AND DISADVANTAGES

ADVANTAGES:

1. It overcomes the problem of time consuming.
2. This helps in speed up the processor.
3. It reduces the human work
4. It can easily check the freshness of any edible.
5. Any person consuming the edible is aware of the quality of fruits or vegetables before the usage.

DISADVANTAGE:

1. Time taken for the conveyor belt to move fruits is more.
2. Internal scanning of edible by means of rays will result in damaging.
3. Since this idea is based upon image processing, there is a drawback in detecting the exact freshness status of the image with low resolution.

APPLICATIONS

1. **Industry**
2. **Agriculture**
3. **Societal**

This project will help farmers and traders to have a quick inspection of their produce and to divert the product to correct market with reasonable pricing estimate saving losses in logistics and transportation to farmers and traders due to rejection of depreciation of the product at marketplace.

RESULTS

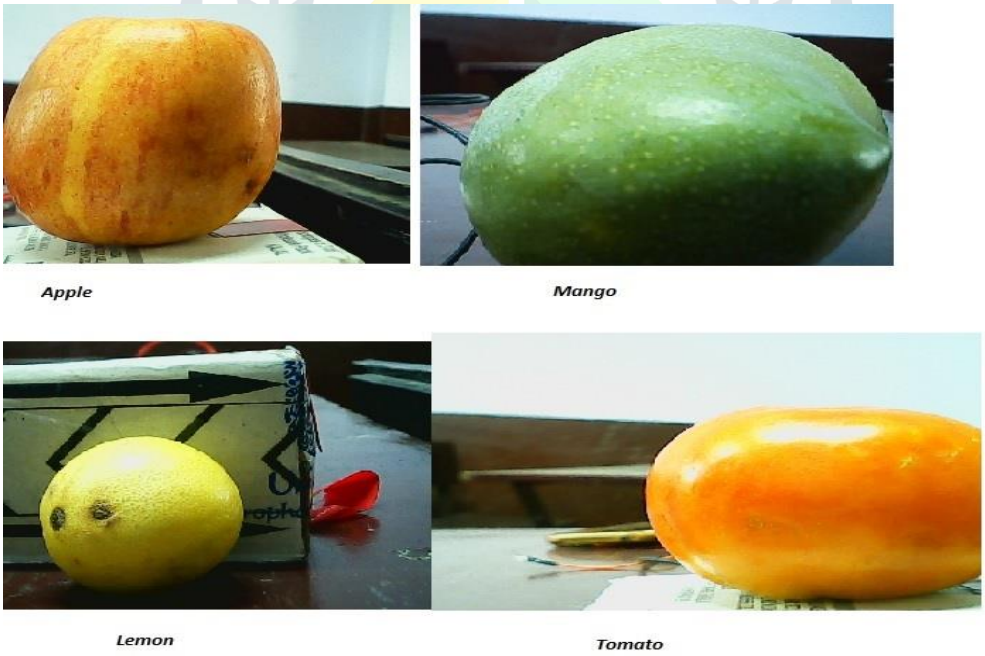

FIG 12.2.1 Images of fruit captured by camera

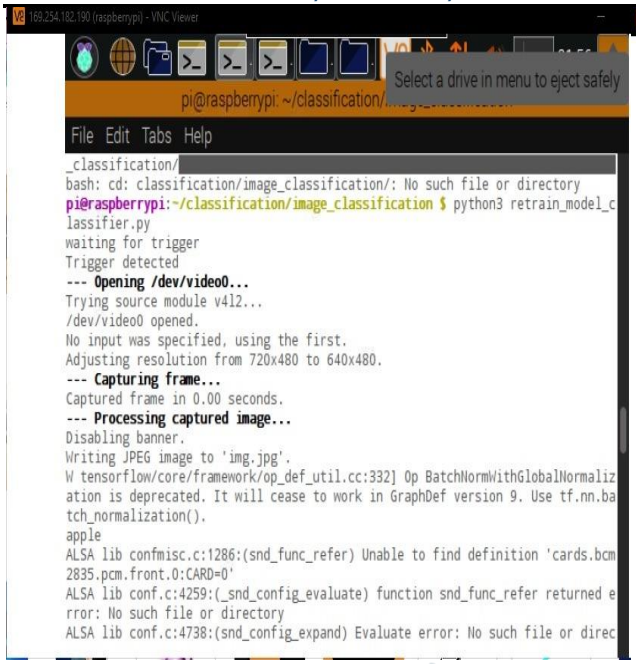

FIG 12.2.2 Result of sorting Apple

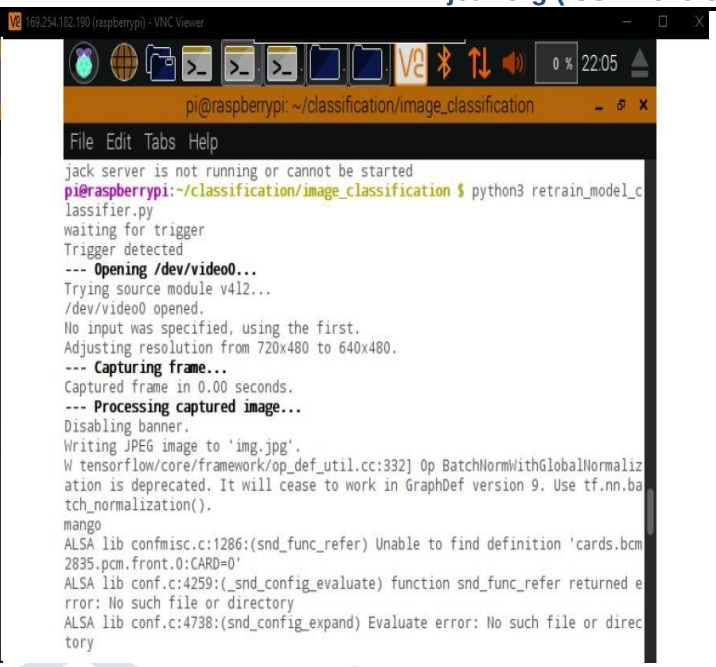

FIG 12.2.3 Result of sorting Mango

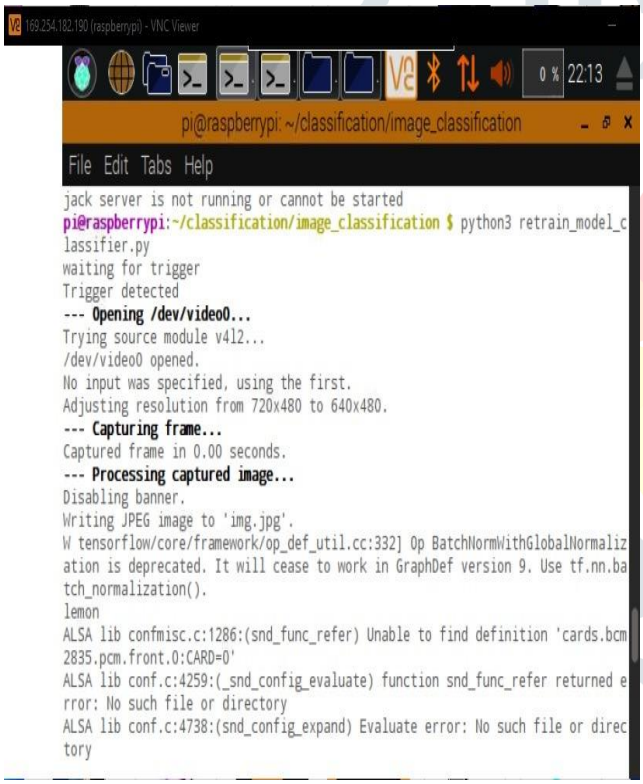

FIG 12.2.4 Result of sorting Lemon

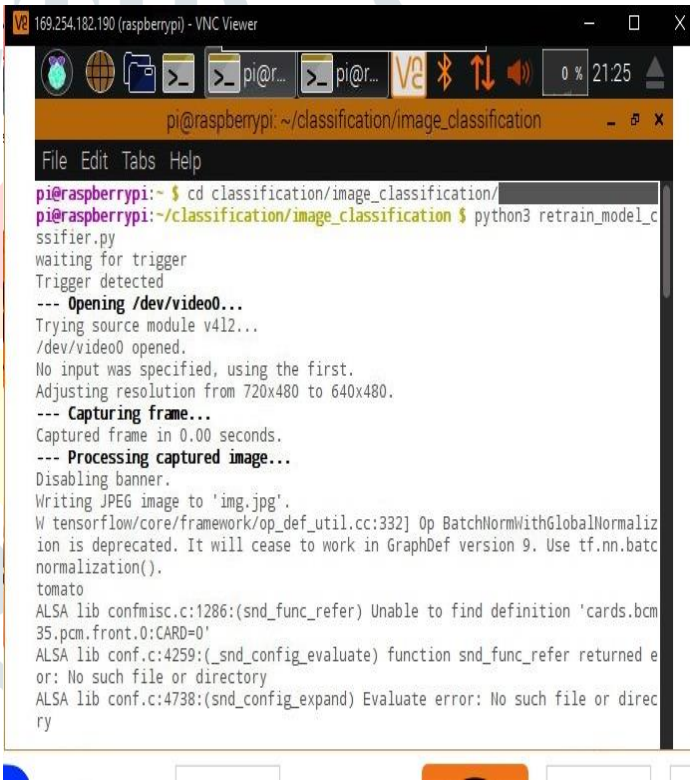

FIG 12.2.5 Result of sorting Tomato

FIG 12.2.6 Image of grading of Rotten Apple

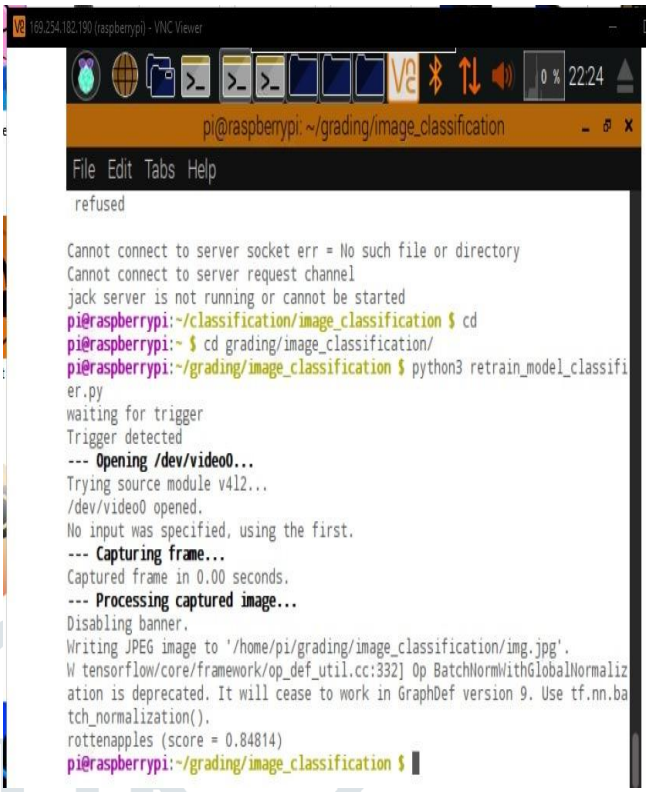

FIG 12.2.7 Result of grading of Rotten Apple

REFERENCES

[1] SooHo Jeong, Hyun Yoe. ”Development of Machine Vision based fruit quality screening system” , Korean Institute of Communication Sciences, 100-101, Nov 2018.

[2] Naganur and sannakki, “Fruits Sorting and Grading using Fuzzy Logic”, International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 1, Issue 6, August 2015.

[3] H. Alimohamadi, A Ahmadyfard, “Detecting Skin Defect of Fruits Using Optimal Gabor Wavelet Filter”, International conference on digital image processing, pp.402-406, 2013.