A MINOR PROJECT REPORT
ON
FRUIT RECOGNITION

SUBMITTED IN PARTIAL FULFILLMENT FOR THE AWARD OF
DEGREE OF
BACHELOR OF TECHNOLOGY
IN
ELECTRONICS AND COMMUNICATION ENGINEERING

Submitted By:
KAPIL SHARMA (9917102148)
SANCHIT GUPTA (9917102167)
SATYAM GUPTA (9917102181)

Under the guidance of :
Mr. VARUN GOEL

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY, NOIDA,
JUNE 2020

CERTIFICATE

This is to certify that the minor project report entitled, "**Fruit Recognition**" submitted by **"Kapil Sharma"** ,**"Sanchit Gupta"** and **"Satyam Gupta"** in partial fulfillment of the requirements for the award of Bachelor of Technology Degree in Electronics and Communication Engineering of the Jaypee Institute of Information Technology, Noida is an authentic work carried out by them under my supervision and guidance. The matter embodied in this report is original and has not been submitted for the award of any other degree.

Signature of Supervisor(s)

Name:Mr. Varun Goel
ECE Department
JIIT Sec-128
Noida-201304

DECLARATION

We hereby declare that this written submission represents our own ideas in our own words and where others' ideas or words have been included, have been adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission.

Place : Noida

Name : Kapil Sharma
Enrollment No. : 9917102148

Name : Sanchit Gupta
Enrollment No. : 9917102167

Name : Satyam Gupta
Enrollment No. : 9917102181

ABSTRACT

The ability to identify the fruits based on quality in the food industry is very important now a days where every person has become health conscious. There are different types of apples available in the market. However, to identify best quality apples is cumbersome task. Therefore, the need arises to develop an algorithm that can detect the apples and classify it in order to increase its market value. Moreover, recognizing and classifying the both types of apples which are good and which are bad with same algorithm is not possible in one go. Therefore, the researchers proposed two different types of approaches for recognizing the apples in the images, one in which apples are good and other in which apples are bad. For the type of images in which apples are bad, one approach is used in which analysis is performed on the basis of color and shape. After recognizing the apple in the respective image, quality is checked by classifier (CNN). Proposed apple recognizing technique analyzes, identifies and classifies apple successfully by CNN.

ACKNOWLEDGEMENT

We would like to take this opportunity to express our profound gratitude and deep regards to our guide and mentor **Mr. Varun Goel** for his exemplary guidance , monitoring and constant encouragement throughout the course of this project.We owe the success of this project to him and are indebted to him for his blessings, help and guidance.

We would also like to extend our gratefulness to our parents and family for their persistent love and support.

Contents

List of Figures

Chapter 1

INTRODUCTION

This is the introduction to the project report that describes the justification for doing the project. The objectives of the work are also described.

1.1 General background

Computer vision is an interdisciplinary field which has been gaining a huge amount of attraction in recent years. The goal of computer vision is to teach a computer to extract information from image as close as possible to humans who can naturally understand everything by seeing the same image. In order to achieve this goal, one of the integral parts of computer vision is object detection and recognition that deals with localizing a particular object region or contour from an image or video and then classifying it. Object detection and recognition have been applied to different problem domains over so many years, whether the objects were handwritten characters , house numbers , traffic signs , objects from the VOC dataset, objects from the 1000-category ImageNet dataset or Caltech-101 dataset. In recent days, object detection is being used for so many applications. There are some state-of-the-arts which work for different types of object detection such as flower detection , fruit detection , food segmentation and detection ,cats and dogs detection etc.The main goal of all these detection algorithms is to obtain higher efficiency and cover different complex use cases by overcoming different limitations.[1]

Fruits provide an essential role as a food in our everyday life. It provides nutrients vital for our health and maintenance of our body. Those who eat more fruits as a part of a healthy diet are likely to have reduced risk of some chronic diseases. However, not all fruits are treated equally and it is a matter of concern that not every person knows about every fruit well. With the help of Artificial Intelligence (AI) and Machine Learning (ML) we can develop an automatic fruit recognition system with an information dataset of each fruit. This system can help us to select fruit that is suitable for us and teach us about the characteristics of that particular fruit. These types of systems can help us to educate children and familiarize them with fruits. Furthermore, these systems can be used to teach a robot to find the correct fruit for its user and this becomes much important for those robots which are being used for fruit harvesting related works.[2]

1.2 Problem statement

One of the main challenge of developing a perfect fruit detection and recognition system is that it needs extra effort in the detection part where in an image the difference between two different fruit is very limited. Fig. 1.1 shows an image of apple and tomato which are very similar in some color and shape. Differentiating red apple from red tomato needs an efficient detection algorithm that can identify objects of similar color and shape by exploiting the difference in their texture.

Figure 1.1: Fruits of similar color and shape

Another major challenge of fruit detection is that fruits are likely to be found in groups and inside of an image they might appear overlapping one another within a same region.

1.3 Project justification

Technological advancement is gradually finding applications in the agricultural and food products, in response to one of the greatest challenges i.e. meeting the need of the growing population. Efforts are being geared up towards the replacement of human operator with automated systems, as human operators are inconsistent and less efficient. So automatic grading of fruit is essential before packaging and possible transportation to different distance according to the number of days left to get mature. The objective of this work is to making this fruit grading systems automatic and fast with the help of computer vision, image processing and fuzzy logic technique.

1.4 YOLO

Humans glance at an image and instantly know what objects are in the image, where they are, and how they interact. The human visual system is fast and accurate, allowing us to perform complex tasks like driving with little conscious thought. Fast, accurate algorithms for object detection would allow computers to drive cars without specialized sensors, enable assistive devices to convey real-time scene information to human users, and unlock the potential for general purpose, responsive robotic systems.

Current detection systems repurpose classifiers to perform detection. To detect an object, these systems take a classifier for that object and evaluate it at various locations and scales in a test image. Systems like deformable parts models (DPM) use

a sliding window approach where the classifier is run at evenly spaced locations over the entire image [3]
More recent approaches like R-CNN use region proposal methods to first generate potential bounding boxes in an image and then run a classifier on these proposed boxes.After classification, post-processing is used to refine the bounding boxes, eliminate duplicate detections, and rescore the boxes based on other objects in the scene [4]. These complex pipelines are slow and hard to optimize because each individual component must be trained separately.
We reframe object detection as a single regression problem, straight from image pixels to bounding box coordinates and class probabilities. Using our system, you only look once (YOLO) at an image to predict what objects are present and where they are.

1.4.1 Comparison With Others

• Deformable parts models-

Deformable parts models (DPM) use a sliding window approach to object detection. DPM uses a disjoint pipeline to extract static features,classify regions, predict bounding boxes for high scoring regions, etc[3]. Our system replaces all of these disparate parts with a single convolutional neural network. The network performs feature extraction, bounding box prediction, nonmaximal suppression, and contextual reasoning all concurrently. Instead of static features, the network trains the features in-line and optimizes them for the detection task.

• R-CNN-

R-CNN and its variants use region proposals instead of sliding windows to find objects in images. Selective Search generates potential bounding boxes, a convolutional network extracts features, an SVM scores the boxes, a linear model adjusts the bounding boxes, and non-max suppression eliminates duplicate detections[5]. Each stage of this complex pipeline must be precisely tuned independently and the resulting system is very slow, taking more than 40 seconds per image at test time .

1.4.2 Similarities

YOLO shares some similarities with R-CNN. Each grid cell proposes potential bounding boxes and scores those boxes using convolutional features. However, our system puts spatial constraints on the grid cell proposals which helps mitigate multiple detections of the same object. Our system also proposes far fewer bounding boxes, only 98 per image compared to about 2000 from Selective Search. Finally, our system combines these individual components into a single, jointly optimized model.

1.4.3 Advantages

1. It is extremely fast. Since we frame detection as a regression problem we don't need a complex pipeline. We simply run our neural network on a new image at test time to predict detections.

2. It reasons globally about the image when making predictions. Unlike sliding window and region proposal-based techniques, YOLO sees the entire image dur-

ing training and test time so it implicitly encodes contextual information about classes as well as their appearance.

3. It learns generalizable representations of objects. When trained on natural images and tested on artwork, YOLO outperforms top detection methods like DPM and R-CNN by a wide margin.

1.4.4 Disadvantages

1. It lags behind state-of-the-art detection systems in accuracy.

2. While it can quickly identify objects in images it struggles to precisely localize some objects, especially small ones.

1.5 Image Processing

Image processing is an area that keeps on growing, with new applications being developed with every passing minute. It is captivating area which has evolved with applications in various areas like entertainment industry, space program etc. The most concerning prospect of this data transition is its capability to transmit and acquire complex information that oversteps normal written text. Visual information sent in the way of images, has evolved as one of the major mode of communication in the 21st century. Description of image processing in detail is described as below.

It is a process of converting an image into the digital image to obtain the enhancement in image or to select some effective information from it. It uses any type of signal processing in which image is input and output of processing may result into an image or features of an image. It includes vast applications in various trades such as business, medical, industries etc.
Following are the main steps including in image processing:

1. Acquiring an image using scanner or by digital cameras.

2. Manipulation and analysis of the image is performed, which includes data compression and enhancement of image.

3. Obtained outcome of image processing can be an image or the details of the characteristics of image based upon the analysis of an input image.

1.6 Purpose of Image processing

Image processing can be used for different objectives. Broader division of purposes is shown below:
• Image Recognition: Objects present in image is get qualified.
• Measurement of pattern: Different objects in an image are measured.
• Image sharpening and restoration: this is for the creation of better image.
• Image retrieval: Image of interest is explored.

1.7 Basics of Digital Image Processing

Basics of digital image processing are explained in the following section, which includes the basic representation of digital image.

1.7.1 Representation of digital Image

Digital image is saved in the form of matrix and can be represented in two dimensional functions of space. Let f (x, y) be the function to represent an image. Input parameter of the function: x represents the row coordinates of image for each pixel and y represents the column coordinates of image for each pixel in the matrix of image. Value at each pixel represents the intensity value of that pixel. Representation of coordinates of digital image is shown in figure 1.2.

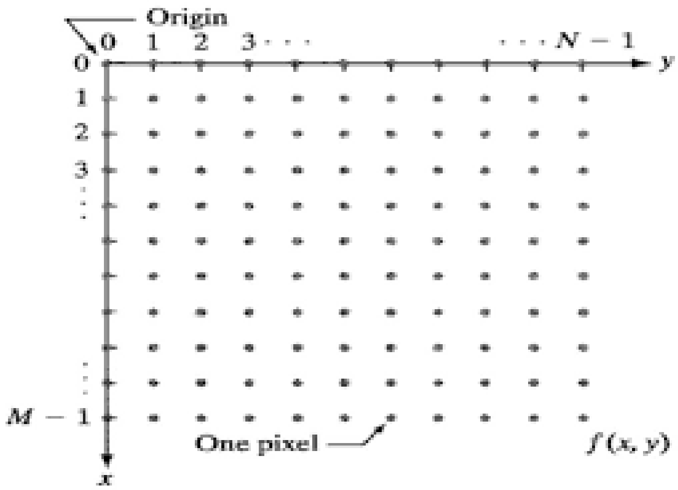

Figure 1.2: Showing coordinates of digital image

As coordinates shown in figure 1.2 signifies that a digital image can be represented in the matrix form mathematically as shown below.

$$f(x,y) = \begin{bmatrix} f(0,1) & f(0,2) & ... & f(0,N) \\ f(1,1) & f(1,2) & ... & f(1,N) \\ \vdots & \vdots & & \vdots \\ f(M,1) & f(M,2) & ... & f(M,N) \end{bmatrix}$$

In mathematical representation of an image, there are M rows and N columns of matrix. That means represented image contains M $\times$ N pixels , where M and N represents width and height of an image. In mathematically form of image , f(x,y) of matrix represents the pixel value of image.

1.8 Types of Digital Image

Digital image includes various types of images in order to perform the processing on images. Basic four types of images are mentioned in following sections:

1.8.1 Binary Image

In a binary image, each pixel assumes one of only two discrete values: 1 or 0. A binary image is stored as a logical array as shown in figure 1.3.

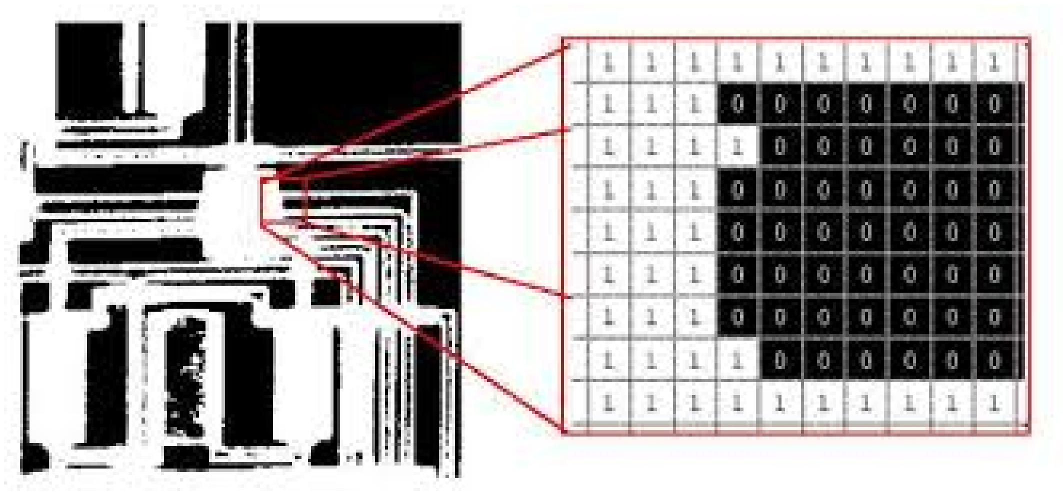

Figure 1.3: Binary Images with its pixel values

1.8.2 Indexed Image

An indexed image (also called pseudocolor image) is that image which comprises of an array and a colormap matrix as shown in figure 1.4. Colormap is directly indexed by the pixel values of array. Traditionally colormap is represented by map.
Colormap matrix is consisted of array of dimension m × 3 where red, green and blue components are stored in each row of colormap. Direct mapping of pixel values to colormap is used by the indexed image. Color of each pixel is according to the stored value in colormap. Values of colormap can be arbitrary.

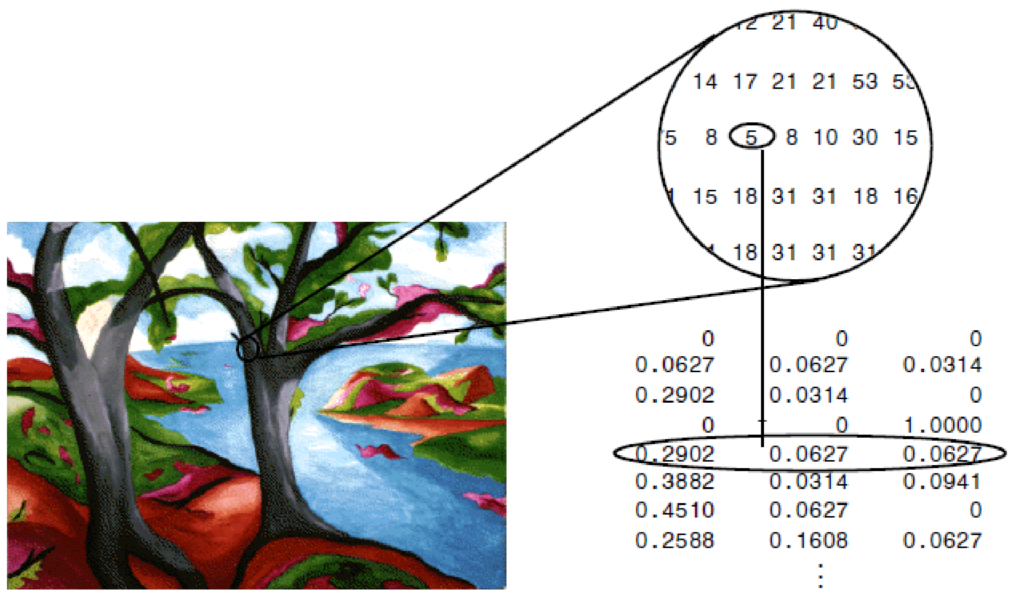

Figure 1.4: Indexed image with its pixel value mapped to colormap

1.8.3 Grayscale Image

A grayscale image (also called intensity, or gray-level) is a single matrix whose each element represents the intensity of each pixel as shown in figure 1.4. Pixel value as 0 signifies black color, 1 signifies the white color and intermediate values are represented by the shades of gray.

Figure 1.5: Grayscale image

1.8.4 Truecolor Image

A truecolor image (also called RGB image) is an image showing natural colors of an object in which each pixel is specified by three values, each for the red, blue, and green planes of the pixel's color. Colormap is not used in truecolor images. The combination of red, green and blue colors saved in each channel gives the color of corresponding pixel.
A pixel whose color planes values are (0,0,0) is displayed as black, and if a color planes values are (255,255,255) is displayed as white. Such as, the red, green, and blue color planes values of the pixel (20,15) are stored in RGB(20,15,1), RGB(20,15,2), and RGB(20,15,3), respectively.

Figure 1.6: Colored Image

1.9 Phases of Digital Image Processing

In processing the digital image, there are various phases which are to be followed in order to extract the required information from the digital image. Phases must be followed one after the other. Figure 1.7 shows the order of phases for processing of digital image.
The phases of digital image processing are:
(a) Preprocessing: It includes the operations, which are to be performed before segmentation.
(b) Segmentation: In this phase, object of interest is identified.
(c) Feature Extraction: Features of segmented object are obtained in this phase.
(d) Classification: Based upon extracted feature, classification is done.

Figure 1.7: Phases of Digital Image Processing

Chapter 2

System Requirements and Software Requirements

2.1 Functional Requirements

1. Processor Required: Intel Core i5 and above
2. RAM Required: 4 Gb and above

2.2 Software Requirements

1. Operating System like Windows (64- bit), Mac, Linux
2. Anaconda Navigator
3. Python 3.5 and above

2.2.1 Operating System

An operating system (OS) is a system software that manages computer hardware, software resources, and provides common services for computer programs. Every computer must have at least one OS to run other programs. An application like Chrome, MS Word, Games, etc. needs some environment in which it will run and perform its task. The OS helps to communicate with the computer without knowing how to speak the computer's language. It is not possible for any user to use any computer or mobile device without having operating system.

2.2.2 Anaconda

Anaconda is a free and open source distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. Package versions are managed by the package management system conda.The packages provided by Anaconda Python distribution includes all those packages that we need in our project. Thus, we use Anaconda here. A key part of the Anaconda Python distribution is Jupyter, an interactive development environment for Python.

2.2.3 Python

Python is an example of a high level language such as C++, PHP, Pascal, C#, and Java. It is a powerful programming language which is easy to learn which has a simple but effective approach to object-oriented programming. It was created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code reliability with its notable use of significant whitespace.
• Python is an interpreted, high level, general purpose programming language.
• It is a dynamic yet modern object -oriented programming language which can be used to do a lot of things and is easy to learn.
• It is a high level language that means this language is closer to humans than computer.
• It is a very flexible language and is also known as a general purpose programming language.
• Python is object -oriented means it regards everything as an object.
• Python being an interpreted language does not need to be complied. For example java programming language.
• Python has been used in a lot of places like in creating games for statistical data and visualization, speech and face recognition.
• Python's name is derived from the British comedy group Monty Python.

2.3 Python Packages Used

MATPLOTLIB
Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack. It was introduced by John Hunter in the year 2002.
One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. Matplotlib consists of several plots like line, bar, scatter, histogram etc.

Chapter 3

Project Implementation And Description

It can be said that You only look once (YOLO) has became very familiar with researchers and developer working in on Computer Vision, especially Object Detection System. According to authors, YOLOv3 is extremely fast and accurate according to other object detection systems such as SSD513, R-FCN, RetinaNet.
In this project we pre-process/prepare our dataset as well as train/save our model with YOLOv3 using GOOGLE COLAB. The object example which we tried to differentiate is apple vs rotten apple.
The dataset required to train a detector with YOLOv3 contains 2 components: images and labels. Each image will be associated with a label file (normally a txt file) which defines the object class and coordinates of object in the image following this syntax: ⟨object-class⟩ ⟨x_center⟩ ⟨y_center⟩ ⟨width⟩ ⟨height⟩
We used labeling Software to label our images and then we trained our model on google colab due to free gpu feature and got our correct results with 94% accuracy.

3.1 Result 1: Apple

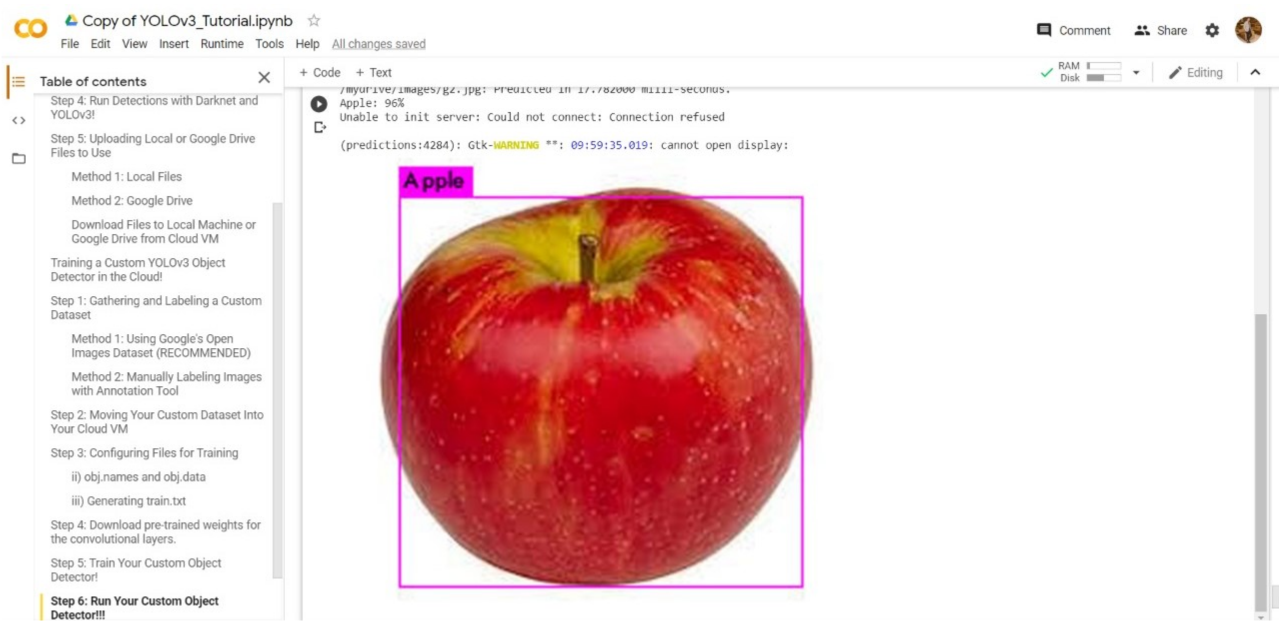

Figure 3.1: Apple

3.2 Result 2: Rotten Apple

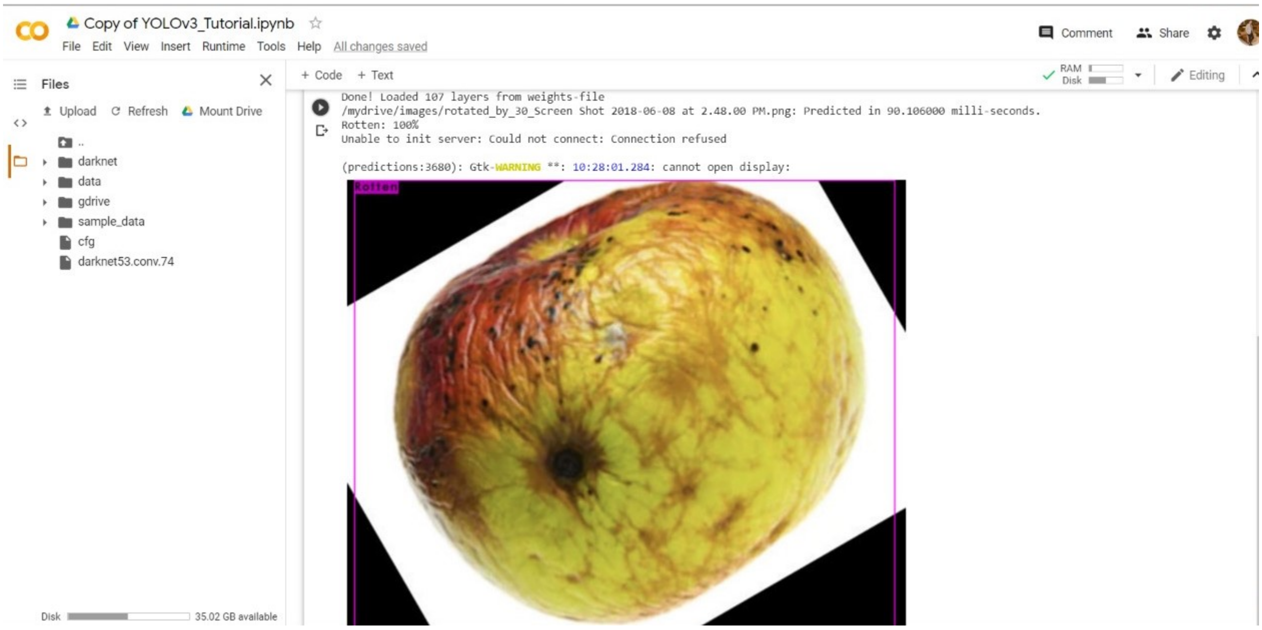

Figure 3.2: Rotten Apple

Chapter 4

Discussion , Conclusion And Recommendations For Further Work

This chapter has the discussion, conclusion and recommendations for further work with regard to this project.

4.1 Discussion

Surface defects are the most essential trait of fruits, which not only affect their market value, but also purchasers inclination to their inward quality also to some extent.
Therefore, it is very essential and important to get a fruit without defects otherwise it becomes cumbersome task at post harvesting stage. Inadequate manual investigation will require a lot of labor and time. In the present work, quality identification of fruit is done by using proposed algorithm for segmentation of fruit.
Total 1926 sample images of apples, where 963(approx.) images of each type (good and bad quality) are taken for testing the proposed algorithm. Segmentation methods are used for images of apples that are Good and images of apples that are Rotten. For the images of apples that are good, segmentation of apple is done on the basis of edge and shape. For the image of apples that are rotten, the segmentation of apple is done based upon its color and shape. After the segmentation, single apple is selected by user. Further feature are extracted of selected apple by utilizing CV yolo approach.

4.2 Conclusion

This project shows a method for automatic grading of Apple fruit using computer vision techniques. The fruit grading system could help the industries to move the apple fruit to the respective bins depending on the output of the image processing module. The software classified the apple image as good Apple/Rotten with an accuracy of 92% and 96.47% respectively. Thus, the developed system will be helpful for the former and agriculture industry in effectively sorting the Apple Fruit. However, the proposed image processing approach does not effectively work on the apple image with high specular reflection. Thus, it needs further improvements, especially in speed and accuracy, before implementing in the field.

4.3 Future Scope

• The outcomes in this research are based on results that involve only sample datasets. It is necessary that additional datasets should be considered for the evaluation of different classification problems as the information growth in the recent technology is extending to heights beyond assumptions. We will also try to improve the accuracy in real time detection of Apple Fruit.

• Detected fruits from image of apples that are not of good quality can be utilized for removing those by a robot. This will speed up the segregation of good quality fruit and will save time. Proposed approach can be improved and enhanced in functionality and flexibility to recognize more efficiently for widely use. Further enhancement can be done by considering the combination of size, color and texture features. Present work can be implemented to identify quality of other fruits.

Bibliography

[1] H. Mureşan and M. Oltean, *Fruit recognition from images using deep learning*, 2018. arXiv: 1712.00580 [cs.CV].

[2] F. Firdaus, E. Rachmawati, and F. Sthevanie, “Hybrid approach for fruit recognition in high data variance”, *AIP Conference Proceedings*, vol. 2233, no. 1, p. 030 003, 2020. [Online]. Available: https://aip.scitation.org/doi/abs/10.1063/5.0001362.

[3] D. M. D. R. Pedro F Felzenszwalb Ross B Girshick, “Object detection with discriminatively trained part based models”, *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[4] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation”, in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 580–587.

[5] T. G. J. R. Uijlings K. E. van de Sande and A. W. Smeulders., “Selective search for object recognition.”, *International journal of computer vision*, vol. 104, no. 2, pp. 154–171, 2013.

Appendix A

Code

```
# clone darknet repo
!git clone https://github.com/AlexeyAB/darknet

    # change makefile to have GPU and OPENCV enabled
%cd darknet
!sed -i ’s/OPENCV=0/OPENCV=1/’ Makefile
!sed -i ’s/GPU=0/GPU=1/’ Makefile
!sed -i ’s/CUDNN=0/CUDNN=1/’ Makefile

    # verify CUDA
!/usr/local/cuda/bin/nvcc –version

    # make darknet (build)
!make

    # get yolov3 pretrained coco dataset weights
!wget https://pjreddie.com/media/files/yolov3.weights

    # define helper functions
def imShow(path):
import cv2
import matplotlib.pyplot as plt
%matplotlib inline

    image = cv2.imread(path)
height, width = image.shape[:2]
resized_image = cv2.resize(image,(3×width, 3×height), interpolation = cv2.INTER_CUBIC)

    fig = plt.gcf()
fig.set_size_inches(18, 10)
plt.axis(”off”)
plt.imshow(cv2.cvtColor(resized_image, cv2.COLOR_BGR2RGB))
plt.show()

    # use this to upload files
def upload():
from google.colab import files
uploaded = files.upload()
```

```
for name, data in uploaded.items():
with open(name, ’wb’) as f:
f.write(data)
print (’saved file’, name)
```

```
# use this to download a file
def download(path):
from google.colab import files
files.download(path)
```

```
# run darknet detection
!./darknet detect cfg/yolov3.cfg yolov3.weights data/person.jpg
```

```
# show image using our helper function
imShow(’predictions.jpg’)
```

```
# look we can run another detection!
!./darknet detect cfg/yolov3.cfg yolov3.weights data/dog.jpg
imShow(’predictions.jpg’)
```

```
%cd ..
from google.colab import drive
drive.mount(’/content/gdrive’)
```

```
# this creates a symbolic link so that now the path /content/gdrive/My Drive/ is equal to /mydrive
!ln -s /content/gdrive/My Drive/ /mydrive
!ls /mydrive
```

```
# this is where my zip is stored (I created a yolov3 folder where I will get my required files from)
!ls /mydrive/yolov3
```

```
# copy the .zip file into the root directory of cloud VM
!cp /mydrive/yolov3/obj.zip ../
```

```
# unzip the zip file and its contents should now be in /darknet/data/obj
!unzip ../obj.zip -d data/
```

```
# upload the custom .cfg back to cloud VM from Google Drive
!cp /mydrive/yolov3/yolov3_custom.cfg ./cfg
```

```
# upload the custom .cfg back to cloud VM from local machine (uncomment to use)
#%cd cfg
#upload()
#%cd ..
```

```
# upload the obj.names and obj.data files to cloud VM from Google Drive
!cp /mydrive/yolov3/obj.names ./data
```

!cp /mydrive/yolov3/obj.data ./data

upload the obj.names and obj.data files to cloud VM from local machine (uncomment to use)
#%cd data
#upload()
#%cd ..

verify train.txt can be seen in our darknet/data folder

!ls data/

upload pretrained convolutional layer weights
!wget http://pjreddie.com/media/files/darknet53.conv.74

train your custom detector
!./darknet detector train data/obj.data cfg/yolov3_custom.cfg darknet53.conv.74-dont_show

run your custom detector with this command (upload an image to your google drive to test, thresh flag sets accuracy that detection must be in order to show it)
!./darknet detector test data/obj.data cfg/yolov3_custom.cfg /mydrive/yolov3/backup/yolov3_custo
/mydrive/images/safari.jpg -thresh 0.3
imShow('predictions.jpg')