A Project Report

On

OBJECT DETECTION MODEL USING DEEP LEARNING

Submitted in partial fulfillment of the

requirement for the award of the

degree

of

Bachelor Of Technology

IN

Computer Science and Engineering

GALGOTIAS UNIVERSITY

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

Under the supervision of

Mr.V.Gokul Rajan

Assistant Professor

Submitted By

Vishesh Kumar
18SCSE1180032 /18021180031
Saurabh Chaudhary
18021050111 /18SCSE1050006

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA
INDIA,
DECEMBER 2021

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the thesis/project/dissertation, entitled **"OBJECT DETECTION MODEL IN DEEP LEARNING"** in partial fulfillment of the requirements for the award of the **Bachelor of technology** submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of July 2021 to December 2021, under the supervision of **Mr. V. Gokul Rajan**, Assistant Professor, Department of Computer Science and Engineering, Galgotias University, Greater Noida

The matter presented in the thesis/project/dissertation has not been submitted by me/us for the award of any other degree of this or any other places.

Vishesh Kumar – 18SCSE1180032

Saurabh Chaudhary - 18SCSE1050006

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Mr. V Gokul Rajan

Assistant Professor

CERTIFICATE

The Final Project Viva-Voce examination of Vishesh Kumar – 18SCSE1180032 and Saurabh Chaudhary – 18SCSE1050006 has been held on ________________ and his/her work is recommended for the award of Bachelor of Technology.

Signature of Examiner(s)

Signature of Supervisor(s)

Signature of Project Coordinator

Signature of Dean

Date: 20 December, 2021

Place: Greater Noida

ABSTRACT

There is growing number of image data in the world and it is increasing rapidly. According to the estimates of InfoTrends in 2016 cameras and mobile devices captured more than 1.1 trillion images and in 2021 the figure rises to 1.6 trillion. In 2014 more than 180 million images were uploaded daily to many social media platforms such as Instagram, Facebook, twitter etc.

Other than consumer devices there are more than millions of cameras are there, taking pictures around the world, monitoring the roads.

To effectively manage all this data, we need to have some idea about its contents. Automated processing of image contents is useful for a variety of image related tasks. Computer vision bridge the gap between the pixel level information stored in image files and human understanding of the images.

Objects can be located and identified in image files automatically. This is known as an object detection and is one of the basic problems of computer vision. We will demonstrate here that convolutional neural networks (CNN) are the currently the solution for object detection.

In the theoretical part, we will show that the literature and study of convolutional object detection methods. In the experimental part, we learn how a convolutional object detection can be implemented in real world.

The goal of this project is “object detection,” to find the location of an object in each image with at most accuracy and show the object with their appropriate category. Object detection is technologically challenging and useful problem in the field of computer vision.

Table of Contents

CHAPTER 1
INTRODUCTION

1.1 Introduction

“Object detection model” is an idea of finding the location of an object int the given pictures and marking the object int their appropriate categories. Unlike the human eye, A computer processes an image in the two dimensions.

Object detection is incredibly challenging in terms of technical and practically used in solving the problem of the computer vision fields. Object detections are achieved successfully in controlled environment but the problems remain unsolved in uncontrolled places like when an object in arbitrary poses in occluded environment. Like in an example a robot can be easily train to detect coffee machine with nothing else in image. On the other hand, difficulty is occurred when there are utensils, gadgets, and many tools are there in the kitchen slab.

Object detection research is multi-disciplinary involves the fields of image processing, machine learning, linear algebra, optimization, topology, statistics, probability etc.

We have used four computer vision and machine learning concepts: 1) Sliding window to extract sub- images from the images, 2) feature extraction to get meaningful data from the sub-images 3) Support Vector Machines (SVM) to classify the objects in sub- images, 4) and Principal Component Analysis (PCA) to improve efficiency.

Object detection model are used in various fields like as facial recognition “Deep Face” developed by group of researchers in Facebook, which identifies the human faces in digital image very effectively. Google has its own facial recognition used in Google photos. People Counting, it is also used for counting people in crowd during festivals. In Industrial quality checks to identify the products, Self-driving cars are the futures.

1.2 Project Purpose

The objective of the object detection is to detect all the instances of objects from a known class, such as people, cars and faces in an image etc.

One of the best examples that why we need object detection is the elevated level algorithm for automated driving

For car to decide what to do next: accelerate, apply, break, or turn etc., it needed to know where all the objects are around the car and what those objects are, that require object detection.

1.3 Tools and Technologies used

Python 3.7

TensorFlow

Anaconda Software

Machine learning Libraries

Conda is an open source, cross-platform, language-agnostic package manager and environment management system that installs, runs, and updates packages and their dependencies. It was created for Python programs, but it can package and distribute software for any language.

1.4 Hardware Specifications:

GPU For good cost/performance, I recommend an RTX 2070 or an RTX 2080 Ti. If you use these cards, you should use 16-bit models. Otherwise, GTX 1070, GTX 1080, GTX 1070 Ti, and GTX 1080 Ti from eBay are fair choices and you can use these GPUs with 32-bit (but not 16-bit). Be careful about the memory requirements when you pick your GPU. RTX cards, which can run in 16-bits, can train models which are twice as big with the same memory compared to GTX cards.

1.5 Expected Outcome

Detection accuracy is usually measured on a given test set where the expected outcome for a detection sample is compared to the actual outcome of the object detection system. The detection accuracy is the percentage of samples for which the expected outcome matches the actual outcome of the detection system

CHAPTER 2
LITERATURE SURVEY

2.1 Machine Learning

Machine learning are useful tools for modeling problems that are difficult to solve. With machine learning some portion of the problem is solved by learning algorithm. There are two types of machine learning algorithm.

2.2 Types of machines learning algorithm

A straightforward way of using machine learning is supervised learning. In supervised leaning multiple examples are shown to the algorithms that are labelled by humans. For example, in object detection problem we use training images where they are marked according to their locations and classes of relevant objects. After learning the algorithm can detect the labels of unseen data. Classification are regression are most important types. Classification predicts the correct class of new piece of data and regression instead of discrete class-ties to predict a continuous output.

In unsupervised learning, the algorithm attempts to learn useful proper-ties of the data without a human teacher telling what the correct output should be. Best example of unsupervised learning is clustering.

2.3 Features extraction

Pre-processing data is always required. Pre-processing the data into new, separate variable space is called features extraction. It is not possible to use the full dimensional training data directly. Detectors are first programmed to extract interesting features of data.

2.4 Generalization

We know that the train data cannot include each instance of the inputs. The learning algorithm able to handle the unseen data points. Sometimes simple model fails to estimate the capture important aspects of the model, sometimes over complex model is overfit to the model by modelling details and noise of unimportant data. An overfit model learns to modelling by given training data but it never understands how to match them.

2.5 Neural networks

Neural networks are especially important type of machine learning model. There is a special case of neural network called convolutional neural network (CNN) is the focus of our thesis.

Even though the inspiration from biology is apparent, it would be mis- leading to overemphasize the connection between artificial neurons and biological neurons or neuroscience. The human brain contains approximately 100 billion neurons operating in parallel. Artificial neurons are mathematical functions implemented on more-or-less serial computers. Research into neural networks is mostly guided by developments in engineering and mathematics rather than biology.

Neural networks were originally called artificial neural networks because they were developed to mimic the neural function of the human brain. Pioneering research includes the threshold logic unit by Warren McCulloch and Walter Pitts in 1943 and the perceptron by Frank Rosenblatt in 1957.

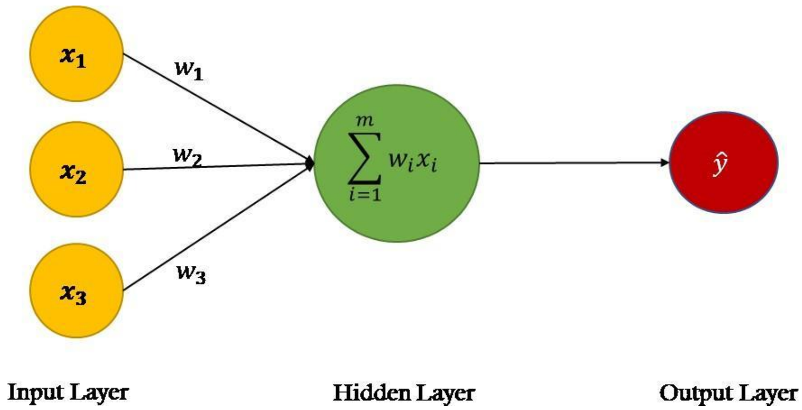


Figure 2.1 An artificial neural network

An artificial neuron based on the McCulloch-Pitts model is shown in Figure. The neuron k receives m input parameters xj. The neuron also has m weight parameters wkj. The weight parameters often include a bias term that has a matching dummy input with axed value of 1. The inputs and weights are linearly combined and summed. The sum is then fed to an activation function' that produces the output yk of the neuron:

The neuron is trained by carefully selecting the weights to produce a desired output for each input.

A neural network is a combination of artificial neurons. The neurons are typically grouped into layers. In a fully-connected feed-forward multi-layer network, shown in Figure 2.2 each output of a layer of neurons is fed as input to each neuron of the next layer. Thus, some layers process the original input data, while some process data received from other neurons. Each neuron has a number of weights equal to the number of neurons in the previous layer.

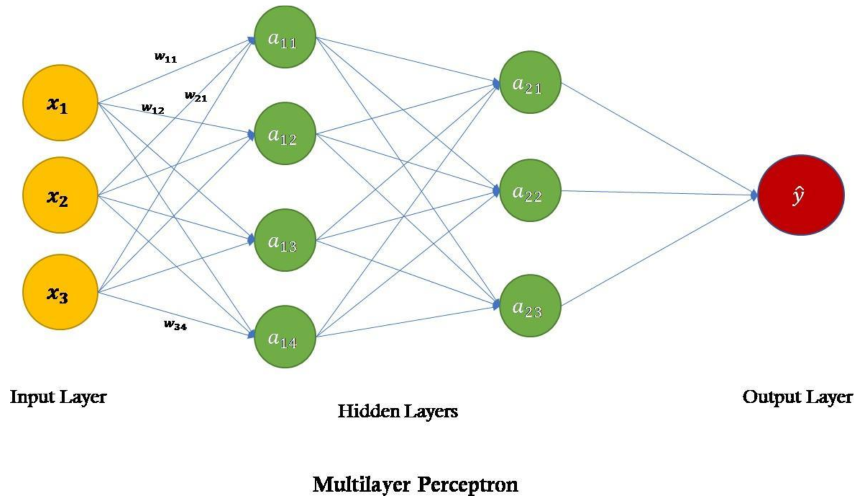


Figure 2.2 A multi-layer neural network

A multi-layer network typically includes three types of layers: an input layer, one or more hidden layers and an output layer. The input layer usually merely passes data along without modifying it. Most of the computation happens in the hidden layers. The output layer converts the hidden layer activations to an output, such as a classification.

In this thesis, we will mostly discuss fully-connected networks and convolutional networks. Convolutional networks utilize parameter sharing and have limited connections compared to fully-connected networks.

2.6 Back-propagation

A neural network is trained by selecting the weights of all neurons so that the network learns to approximate target outputs from known inputs. It is difficult to solve the neuron weights of a multi-layer network analytically. The back propagation algorithm provides a simple and effective solution to solving the weights iteratively. The classical version uses gradient descent as optimization method. Gradient descent can be quite time-consuming and is not guaranteed to find the global minimum of error, but with proper configuration (known in machine learning as hyper-parameters) works well enough in practice.

In the first phase of the algorithm, an input vector is propagated forward through the neural network. Before this, the weights of the network neurons have been initialized to some values, for example small random values. The received output of the network is compared to the desired output (which should be known for the training examples) using a loss function. The gradient of the loss function is then computed. This gradient is also called the error value.

The error values are then propagated back through the network to calculate the error values of the hidden layer neurons. The hidden neuron loss function gradients can be solved using the chain rule of derivatives. Finally, the neuron weights are updated by calculating the gradient of the weights and subtracting a proportion of the gradient from the weights. This ratio is called the learning rate. The learning rate can be fixed or dynamic. After the weights have been updated, the algorithm continues by executing the phases again with different input until the weights converge.

2.7 Activation function types

The activation function' determines the final output of each neuron. It is important to select the function properly to create an effective network.

The perceptron's and other linear systems had severe drawbacks, being unable to solve problems that were not linearly separable, such as the XOR problem. Sometimes, linear systems can solve these kinds of problems using hand-crafted feature detectors, but this is not the most advantageous use of machine learning. Simply adding layers does not help either, because a network composed of linear neurons remains linear no matter how many layers it has.

A light-weight and effective way of creating a non-linear network is using rectified linear units (ReLu). A rectified linear function generates the output using a ramp function such as:

For multi-class classification problems, the SoftMax activation function is used in the output layer of the network: The SoftMax function takes a vector of K arbitrarily large values and outputs a vector of K values that range between 0...1 and sum to 1. The values output by the SoftMax unit can be utilized as class probabilities.

2.8 Computer vision

Computer vision deals with the extraction of meaningful information from the contents of digital images or video. This is distinct from mere image processing, which involves manipulating visual information on the pixel level. Applications of computer visioninclude image classification, visual detection, 3D scene reconstruction from 2D images, image retrieval, augmented reality, machine vision and traffic automation.

Today, machine learning is a necessary component of many computer vision algorithms.Such algorithms can be described as a combination of image processing and machine learning. Effective solutions require algorithms that can cope with the vast amount of information contained in visual images, and critically for many applications, can carry out the computation in real time.

Object detection is one of the classical problems of computer vision and is often described as a difficult task.To detect an object, we need to have some idea where the object might be and how the image is segmented. To recognize the shape (and class) of an object, we need to know its location, and to recognize the location of an object, we need to know its shape.

2.9 Object detection

Object detection is one of the oldest problems with computer vision and it often occurs described as hard work. In many respects, it is similar to other computer vision functions, because it involves creating a consistent solution to change and change light and vision. What makes the discovery of an object a different problem is that it includes both finding and separating image regions. Part of the space is not needed, because

for example, the separation of the whole image.

To find something, we need to have an idea of where and how to get it

image separated. This creates a kind of chicken-and-egg problem, there, to be aware of the shape (and category) of an object, we need to know its location, and be careful the location of an object, we need to know its shape. Some different looking features, such as like a person's clothing and face, it may be part of the same thing, but it is it is difficult to know this without first seeing the object. On the one hand, some things stand out only slightly in the background, which need to be separated from the front recognition.

Low image visual features, such as a saliency map, can be used as a guide to find candidate items. Location and size are usually defined using binding box, stored in the form of corner links. Using a rectangle is easier than ever using the wrong polygon, and many functions, such as convolution, are done in rectangles anywhere. The small image contained in the binding box is then separated by an algorithm trained using machine learning. Object parameters can be continuously improved, after initial guesses.

In the 2000s, popular discovery solutions used feature adjectives, such as the scale-invariant feature transform (SIFT) developed by David Lowe in 1999 and

The histogram of oriented gradients (HOG) became popular in 2005. In 2010, there was a shift in the use of convolutional neural networks.

Prior to CNN's widespread adoption, there were two competing solutions

to produce binding boxes. In the first solution, a dense set of regional proposals is produced and most of them are rejected. This usually involves a slippery window detector. In the second solution, a small set of binding boxes is produced using a

regional suggestion method, such as Selected Search. Combining sub-regional proposals with convolutional neural networks has yielded positive results and is currently popular.

2.10 Convolutional Neural Network

2.10.1 R-CNN

The R-CNN calculation up front has a few sections, shown in the picture. First, the regions of interest is produced. RoIs are independent binding boxes with high chances of containing something exciting. On paper, a different approach is called Selected Search, used to generate these, but other regional production methods may be used instead. Selected Search, and other strategies for generating regional proposal discussed in more detail in section 3.3.

Next, a convolutional network is used to extract features from each regional proposal. I the small image contained in the binding box is distorted to match the input size of CNN and fed to the network. After the network removes the features from the input, i features inputs to support vector equipment (SVM) that provides final stages

2.10.2 Fast R-CNN

Fast R-CNN published in 2015 by Girshick provides an effective way of doing things

recognition. The main idea is to make CNN transitions complete image, instead of doing it separately on each RoI.

2.11 Region proposal generation and use

In order to use R-CNN and Fast R-CNN, we need a way to produce interesting class-agnostic regions. Next, we will discuss the common principles of RoI production, and take a look at two popular methods: Selected Search Boxes and Edge.

The purpose of producing a regional proposal on acquisition of an object is to increase memory i.e. to produce enough regions to make all the real objects available. The generator does not care much about accuracy, as it is the job of the detector to determine the correct

regions from the production of a regional proposal generator.

However, the number of suggestions produced affects performance. As mentioned in paragraph 2.3.2 there are two main methods of regional production: dense production and the appointed generation.

Solid set solutions attempt to produce aggressively by forcing a complete set of binding boxes that encompass every possible object. This can be achieved by sliding recovery window across the image. However, searching everywhere for an image is mathematically costly and requires a quick detector. Additionally, windows of different shapes and sizes need to be considered. Thus, most sliding window options limit the amount of candidate items by using coarse step size and approximate value fixed aspect ratio.

Many regional suggestions in the dense set do not contain interesting content. These are suggestions needs to be discarded after the acquisition phase. Discovery results can be discarded, if they fall behind a predefined confidence level or if their confidence level is low

area size (non-maximum pressure).

Instead of dumping circuits behind the acquisition stage, a regional proposal the generator itself can measure the regions in a class-agnostic manner and discard the low position regions. This creates a small set of acquisitions. As with conventional setup methods, threshold and non-maximum suppression can be performed after detection phase to improve acquisition quality. Uncertain set solutions can be integrated into unattended and unattended roads.

One of the most popular unsupervised methods is Search Options (see paragraph 3.3.2) using repeated combinations of large pixels. There are other methods used the same way. Another way is to measure the variability of the sliding window. A prominent example of this is the Edge Boxes (see section 3.3.2) which count opposition earn points by counting the number of edges inside the binding box and subtracting i number of edges that exceed the box boundary. There is also a third group of methods based on seed classification.

Supervised approaches handle the production of a regional proposal such as division or retreat problem. This means using a machine learning algorithm, as a supporting vector machine. It is also possible to use a convolutional network to produce regions interest. An example of using CNN in calculating Multi-Box binding boxes.

2.12 Advanced convolutional object detection

1.12.1 Faster R-CNN

Ren et al's fast R-CNN. it is an integrated approach. The main idea is to use the shared layers of convolution for regional proposal and acquisition. Authors find that feature maps produced by discovery networks can also be used to produce regional proposals. Fully modified part of the Faster R-CNN network which produces a feature proposal is called a regional proposal network (RPN). I the authors used R-CNN's fast-paced structures in the acquisition network.

The fast-paced R-CNN network is trained in exchange for RoI generation training and adoption. First, two different networks are trained. Then, these are the networks neatly integrated and configured. During fine tuning, certain layers are kept consistent and secure the layers are trained in sequence.

A trained network receives a single image as included. What is shared is fully compatible layers produce feature maps in the image. These feature maps are provided by RPN. I Outgoing RPN regional suggestions, inputs, and feature featured maps, so that final layers of adoption. These layers include the integration layer of RoI and extract the final one in stages.

With the use of shared convolutional layers, regional proposals are almost free.

Making computer-based suggestions on CNN has the added advantage of being implemented in a GPU. Traditional RoI production methods, such as Selective Search, are used using the CPU. To deal with the different shapes and sizes of the viewing window, the method is used special function boxes instead of using the scaled-up image pyramid or the pyramid of various filter sizes (see section 7.2 for a discussion of scale consistency). Anchor boxes serves as reference points in various regional proposals focusing on the same pixel.

2.12.2 SSD

The Single Shot MultiBox Detector (SSD) captures even more integrated detection. I method does not generate suggestions at all, nor does it involve re-image resizing parts. It produces the acquisition of objects using more than one convolutional network.

Somehow resembling a sliding window path, the algorithm starts with a default set of binding boxes. This includes different scales and scales. Into the predictions listed in these boxes include o set limits, predicting how much a fine binding box around the object identifiers from the fixed box.

The algorithm deals with different scales using feature maps from many different ones convolutional layers (i.e. large and small feature maps) as included in the section. Since the method produces a dense set of binding boxes, the stage is followed by a non-high pressure section that removes many boxes under a certain confidence.

CHAPTER 3

SYSTEM DESIGN

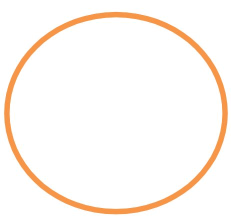
Data Flow Diagrams

System presentation at different levels of data with a clear combination of data representing data flow, data stores, processes and data storage areas as a source and locations is called Data Flow Diagram.

Design Notations

Process

A process or process performs tasks and provides output to what is provided conflicts. Pure activities are considered to be a low-level process with no negative consequences. The process part of the data flow is represented as an ellipse.



Data Flows

The combination of one process to another or the single identification of the mother is represented by the average value or label on it.

Actors

The element that drives the data flow by taking the inputs and thereby computing the out is termed as the actor.

Data Store

Sometimes data is required for later access to data flows generated by the DFD data store.

External Entity

Any foreign organization that can access the flow to DFD as a library secretary, is called part of the Foreign Organization. It is represented as a rectangle.

Graphical Representation

Output Symbol

While user interaction with the system DFD displays it in the form of a polygon below.

Graphical Representation

Detailed Design

Zero level DFD – object identification system

Fig. 4.3.1

First Level DFD-

Pre-Processing image

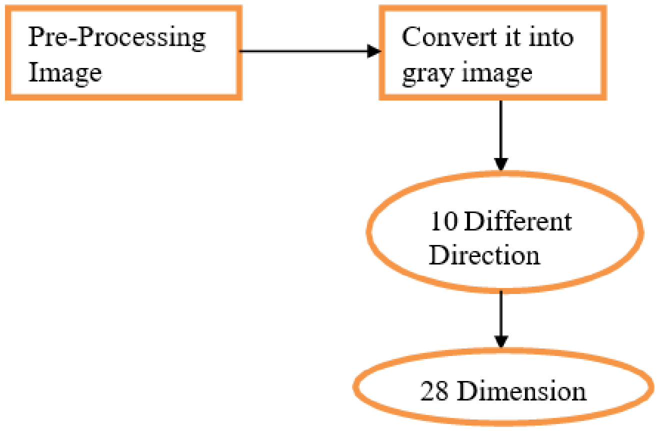
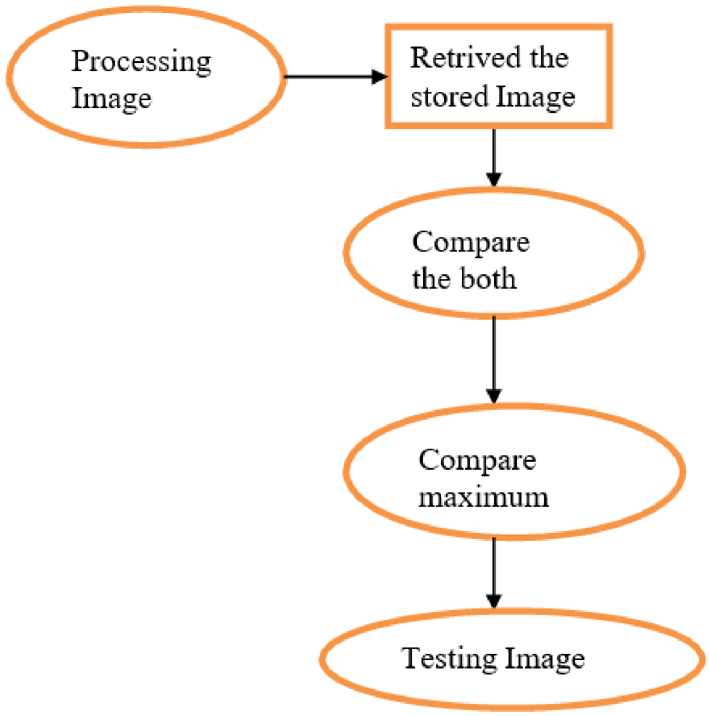


Fig. 4.3.2

Recognition



Processing

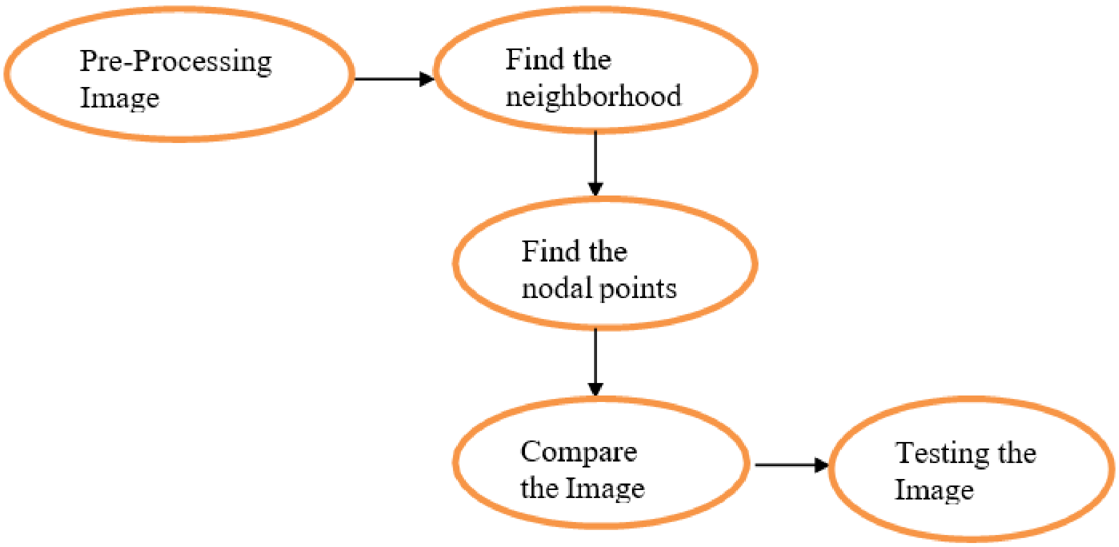


Fig. 4.3.3

Testing the Image

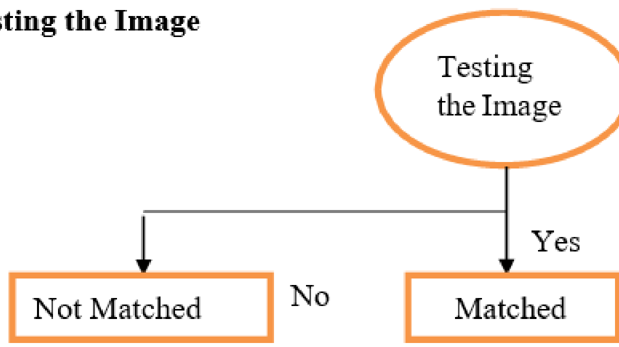
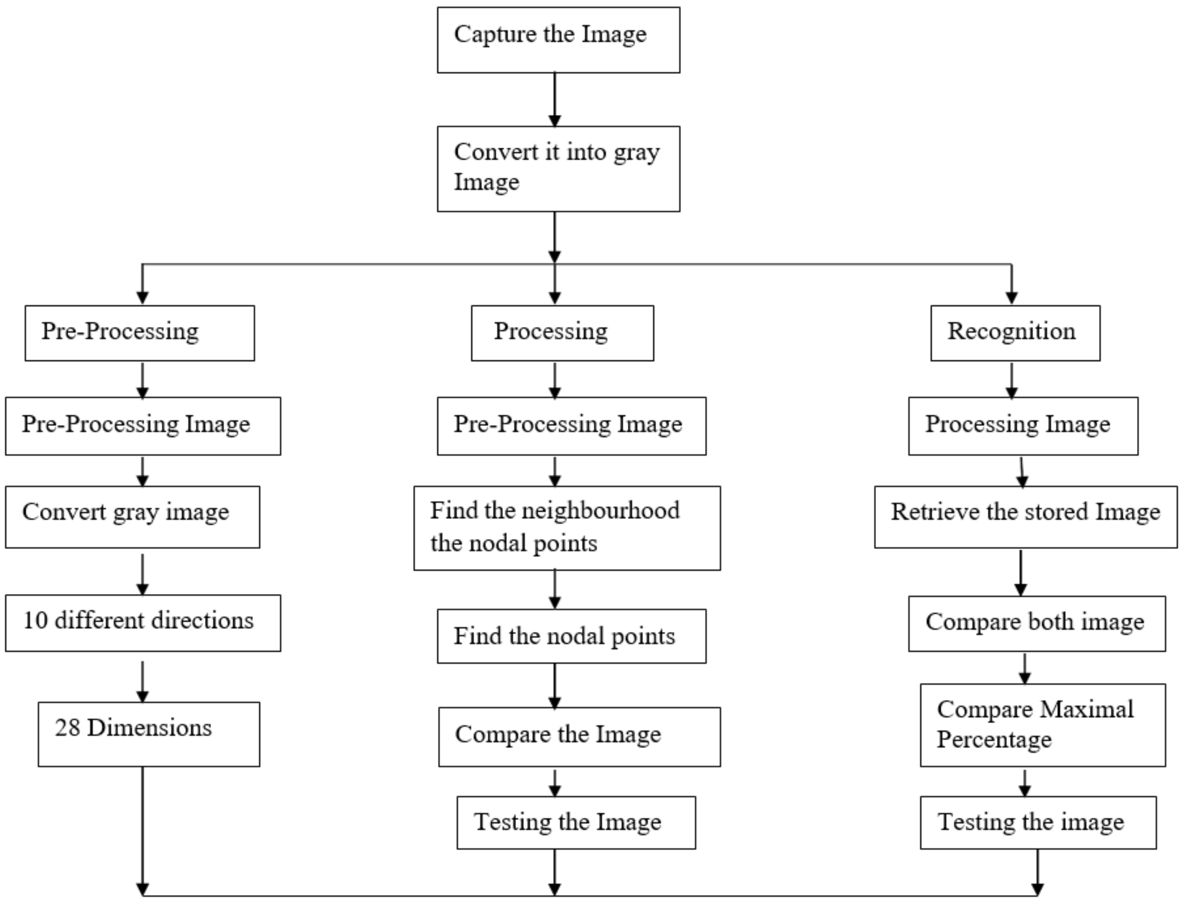


Fig. 4.3.5

Second level DFD



Use case

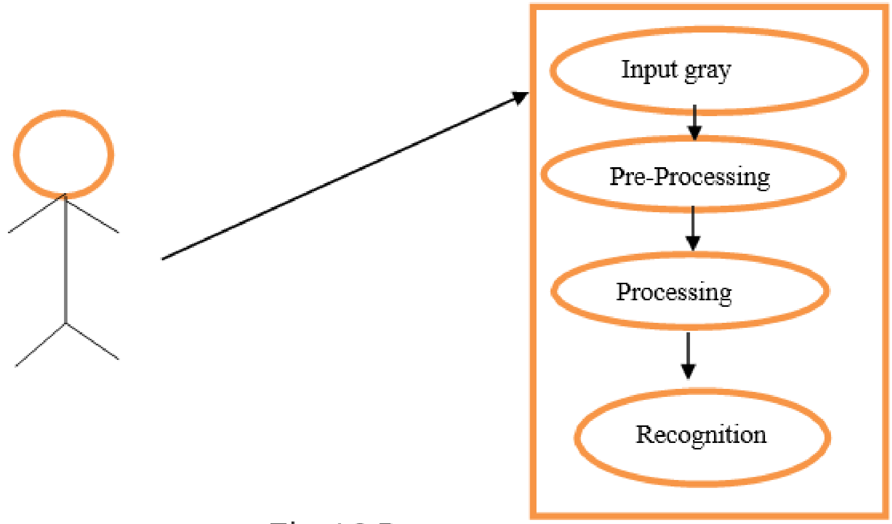


Fig. 4.3.7

Sequence Diagram

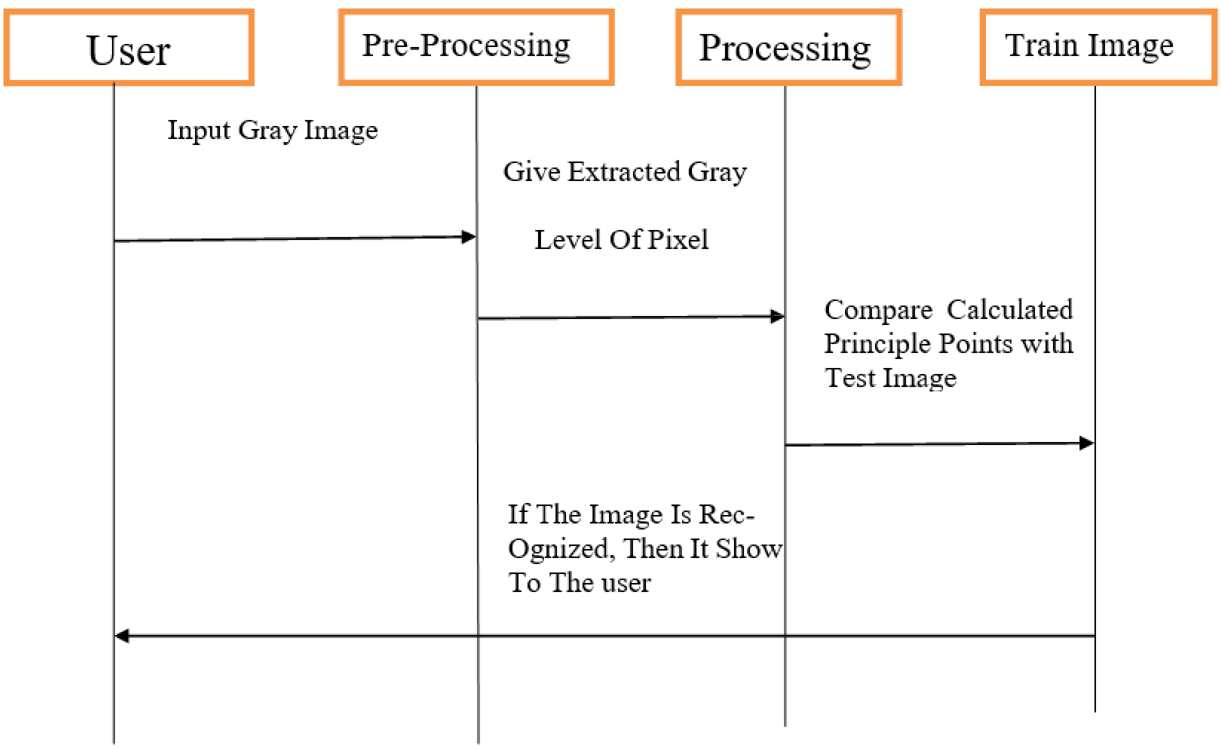


Fig.5.3.8

CHAPTER 4

Working of project

Snapshots:

3.1 Raw images

The first step is collecting images for our project. Download them from goggle .I ensured that images were taken from multiple angles, brightness, scale etc.so that the detector can work under different conditions of lightning and angles. Overall 100–150 pics will suffice. Some sample images below:

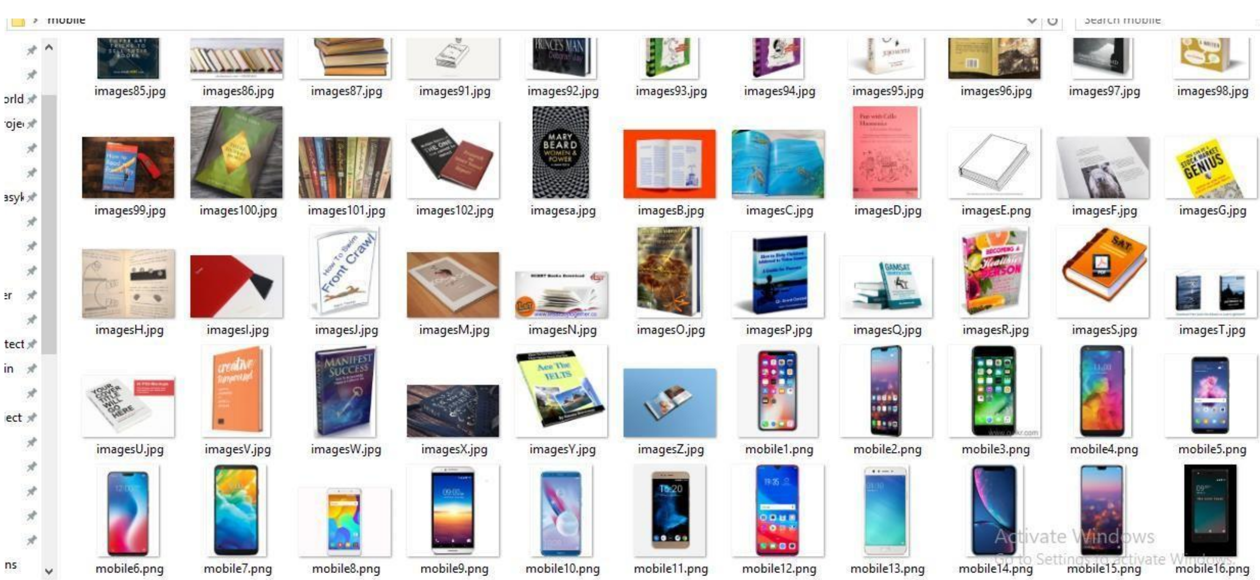


Fig 3.1

3.2 Labelling the image

I used [labelimg]() to annotate the images. Annotations are created in the Pascal VOC format which is useful later on. It is written in Python and uses Qt for interface. I used Python3 + Qt5 with no problems. example of annotated image. Essentially we identify xmin, ymin, xmax and ymax for the object and pass that to the model along with the image for training

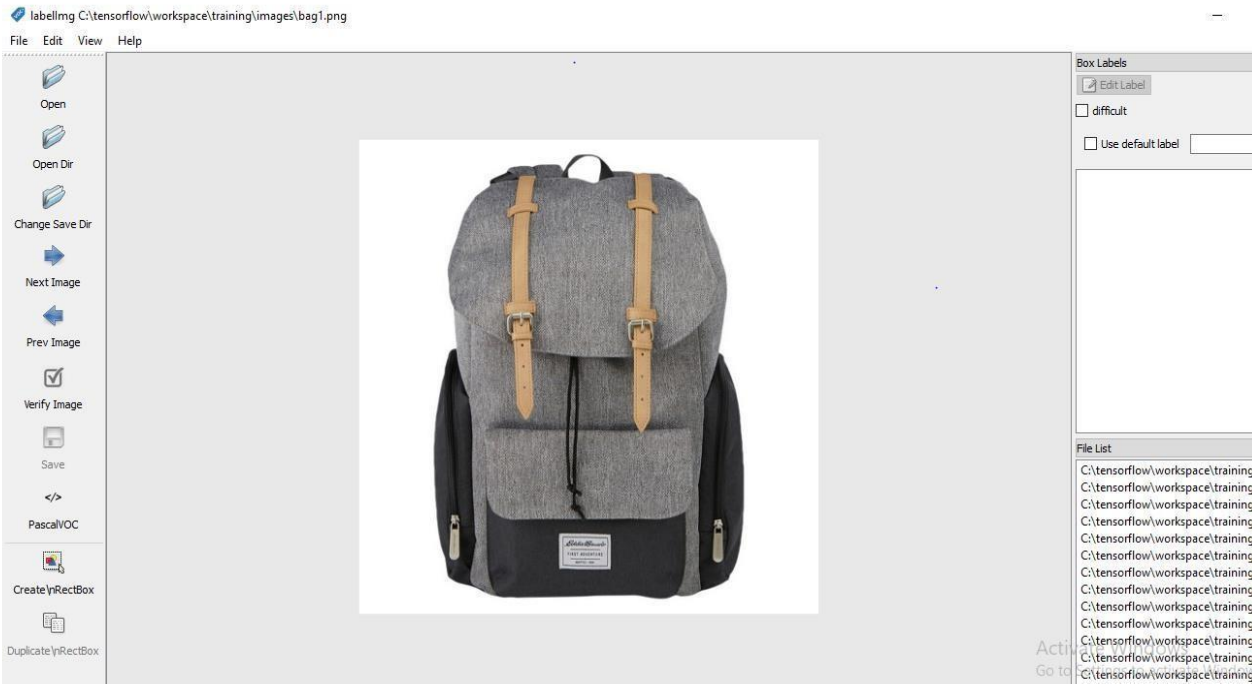


Fig 3.2

Creating XML files

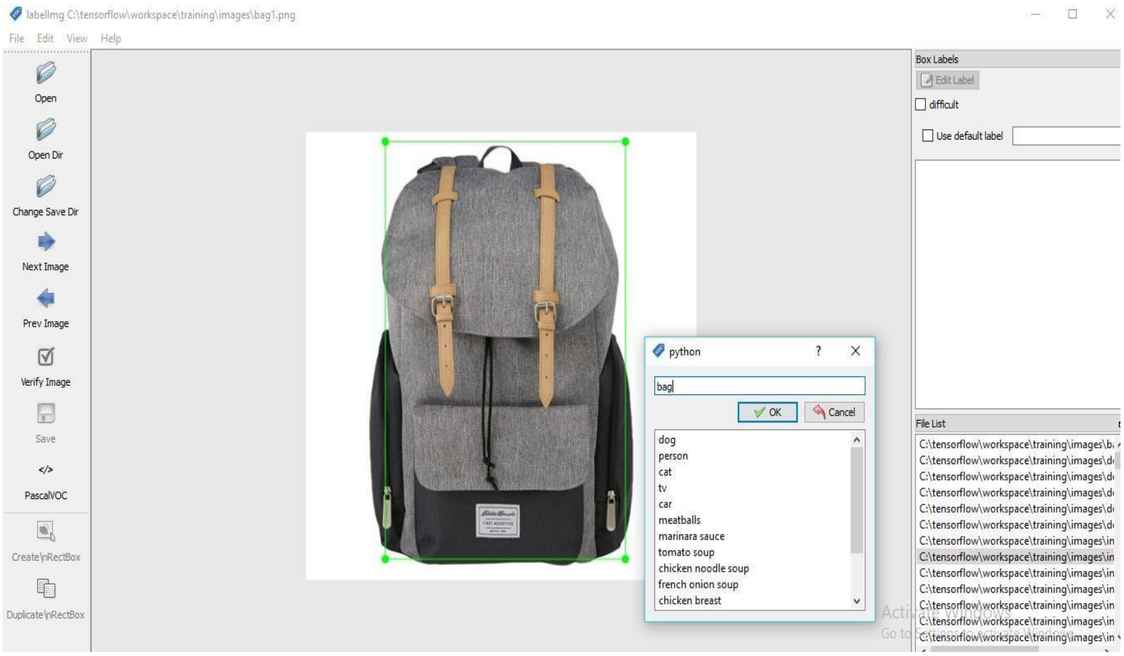
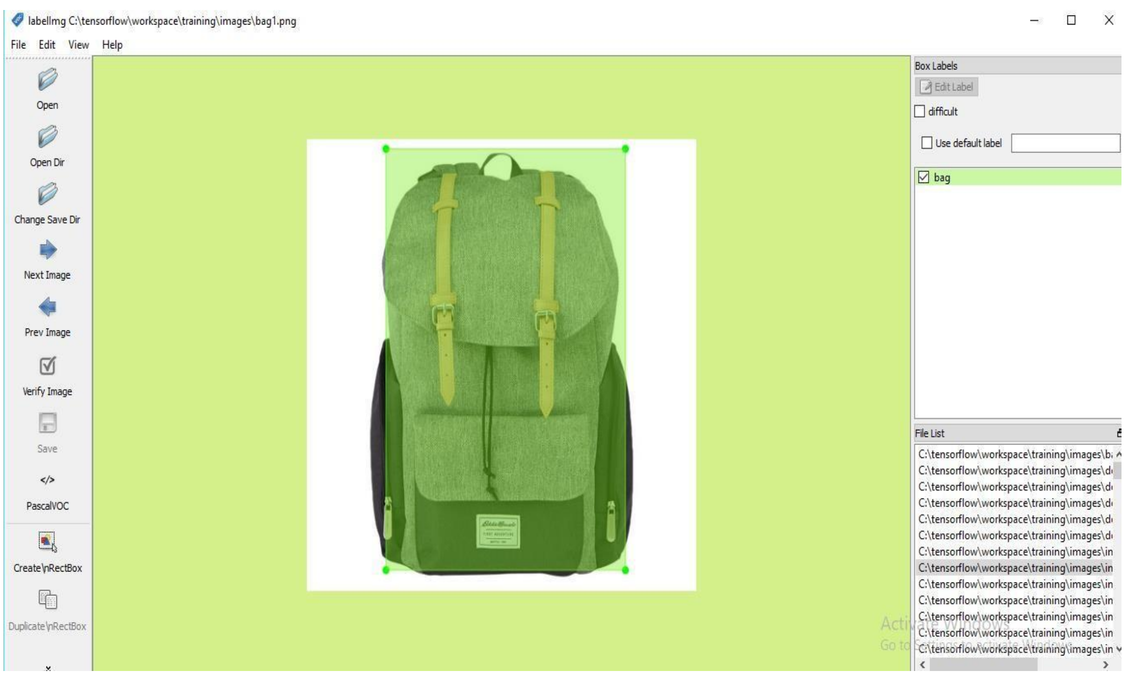

labelImg C:\tensorflow\workspace\training\images\bag1.png
File
Edit
View
Help
Open
Open Dir
Change Save Dir
Next Image
Prev Image
Verify Image
Save
PascalVOC
Create\nRectBox
Duplicate\nRectBox
Box Labels
Edit Label
difficult
Use default label
python
bag
OK
Cancel
dog
person
cat
tv
car
meatballs
marinara sauce
tomato soup
chicken noodle soup
french onion soup
chicken breast
File List

Fig 3.3


labelImg C:\tensorflow\workspace\training\images\bag1.png
File
Edit
View
Help
Open
Open Dir
Change Save Dir
Next Image
Prev Image
Verify Image
Save
PascalVOC
Create\nRectBox
Duplicate\nRectBox
Box Labels
Edit Label
difficult
Use default label
bag
File List

Another example of annotate the image we use upto 100 images for each object

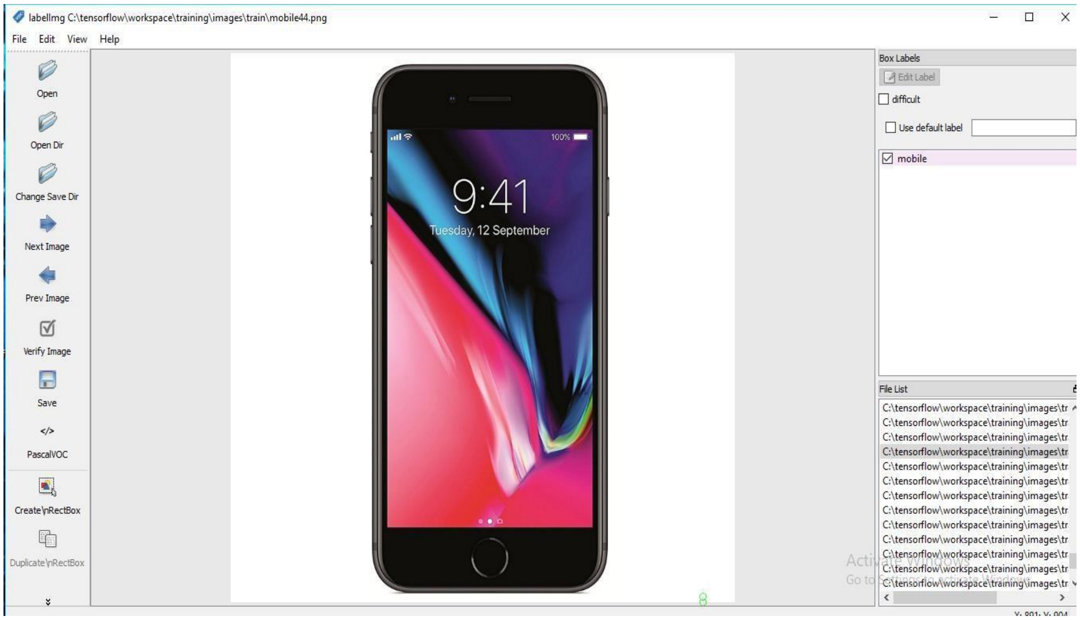
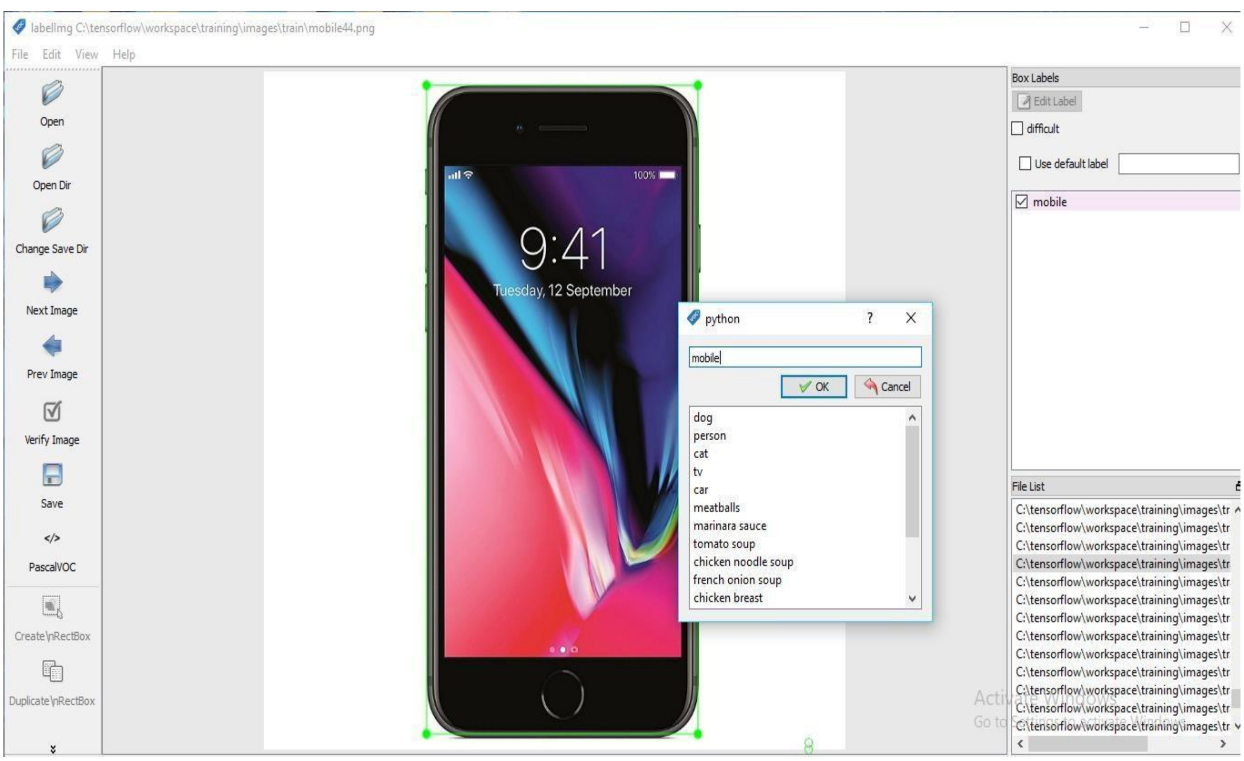
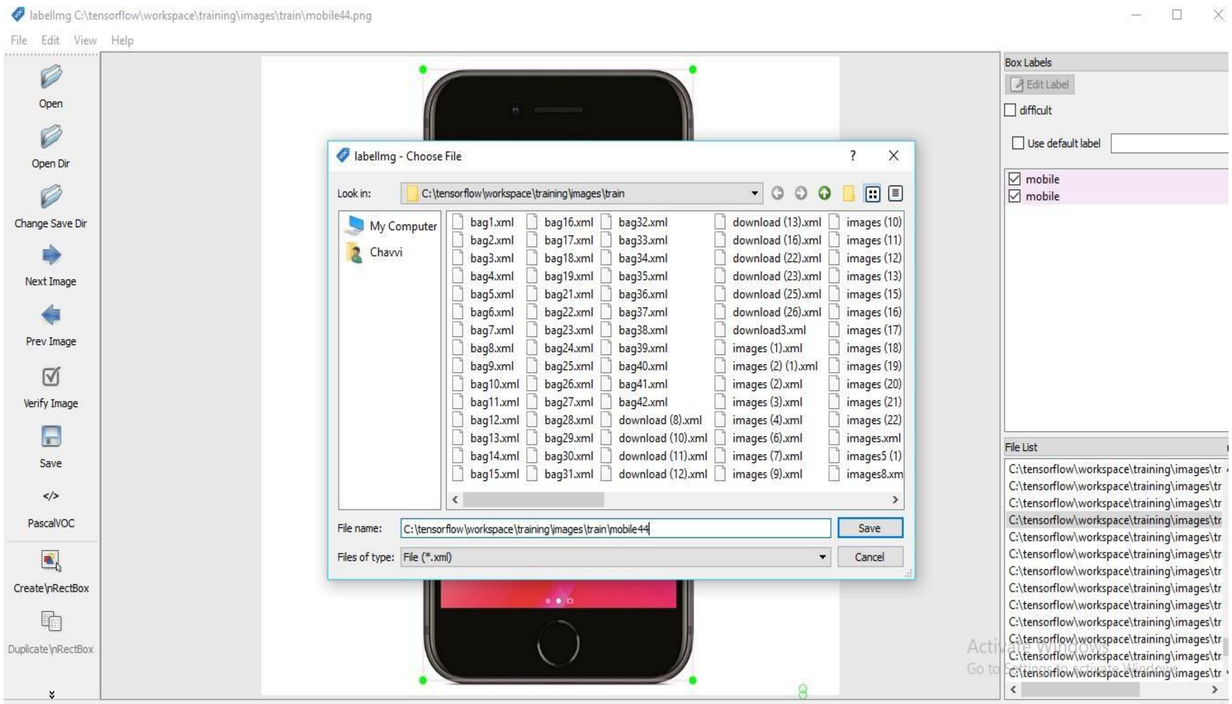


Fig.3.5

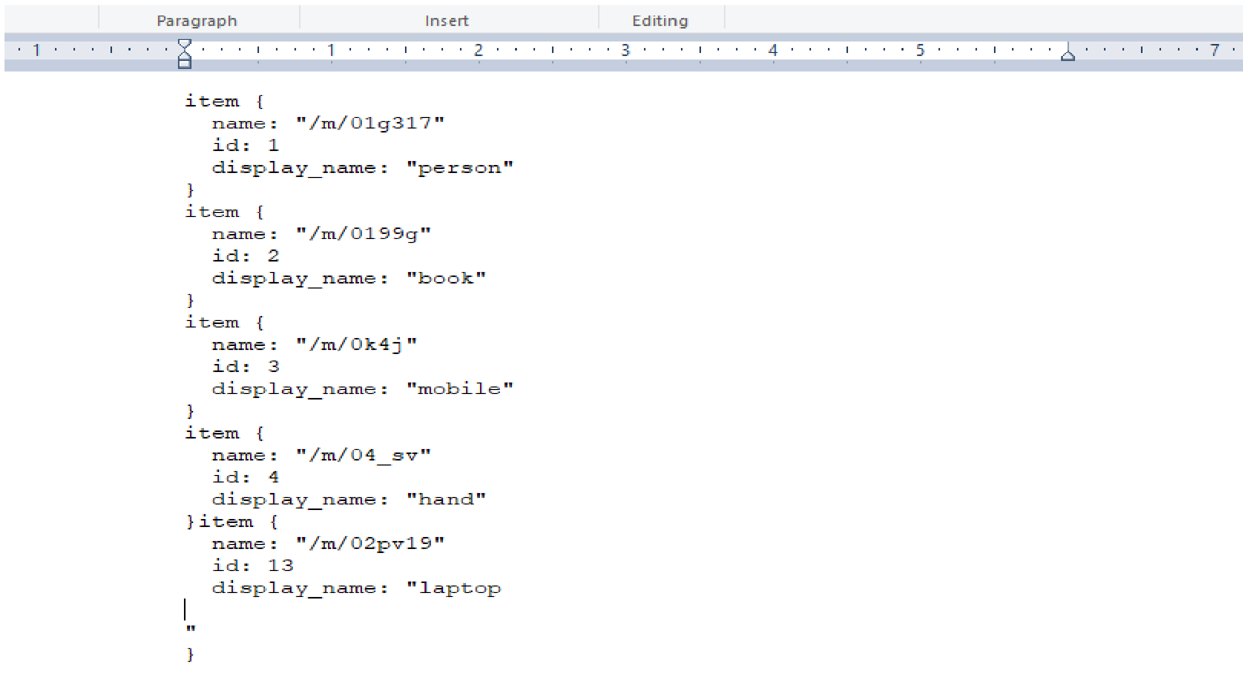
Bounding Box





After annotating the image we create a label map which includes item name, id and display name there is one label for each object

create a label.pbtxt file that is used to convert label name to a numeric id.

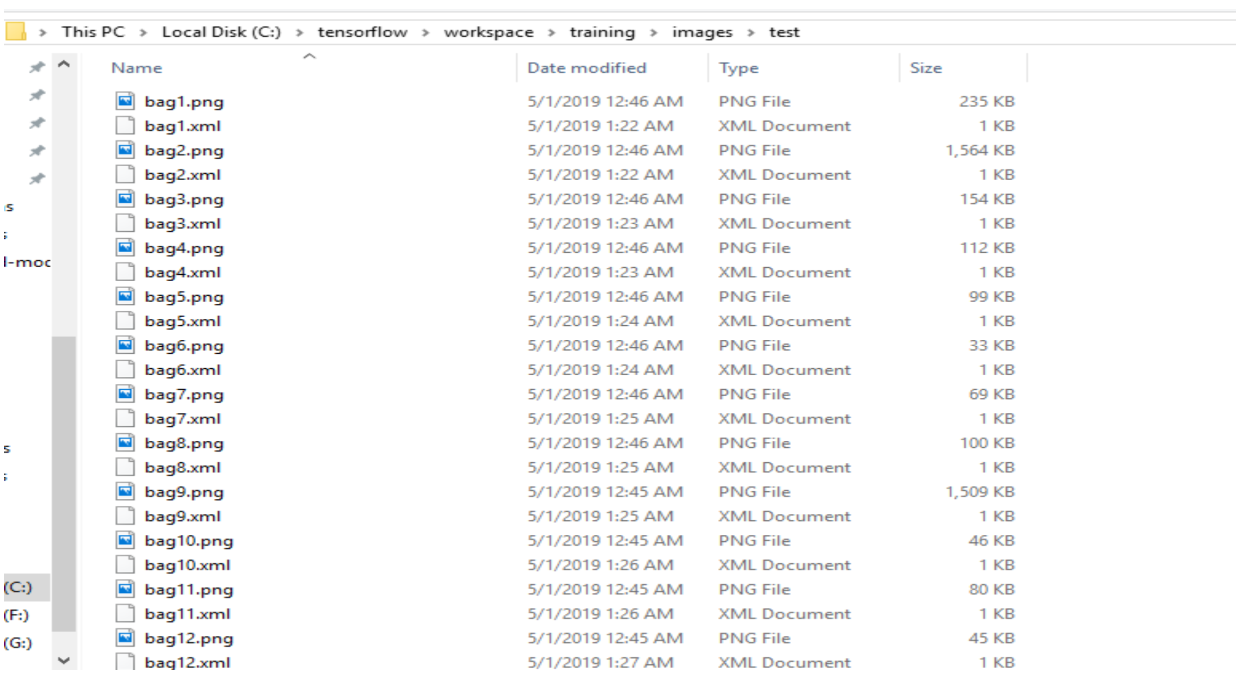



```
item {
  name: "/m/01g317"
  id: 1
  display_name: "person"
}
item {
  name: "/m/0199g"
  id: 2
  display_name: "book"
}
item {
  name: "/m/0k4j"
  id: 3
  display_name: "mobile"
}
item {
  name: "/m/04_sv"
  id: 4
  display_name: "hand"
}item {
  name: "/m/02pv19"
  id: 13
  display_name: "laptop

"
}
```

Raw Images and XML files

This show all the images store in the test and train folder. This images help in taining and testing of the object



Name	Date modified	Type	Size
bag1.png	5/1/2019 12:46 AM	PNG File	235 KB
bag1.xml	5/1/2019 1:22 AM	XML Document	1 KB
bag2.png	5/1/2019 12:46 AM	PNG File	1,564 KB
bag2.xml	5/1/2019 1:22 AM	XML Document	1 KB
bag3.png	5/1/2019 12:46 AM	PNG File	154 KB
bag3.xml	5/1/2019 1:23 AM	XML Document	1 KB
bag4.png	5/1/2019 12:46 AM	PNG File	112 KB
bag4.xml	5/1/2019 1:23 AM	XML Document	1 KB
bag5.png	5/1/2019 12:46 AM	PNG File	99 KB
bag5.xml	5/1/2019 1:24 AM	XML Document	1 KB
bag6.png	5/1/2019 12:46 AM	PNG File	33 KB
bag6.xml	5/1/2019 1:24 AM	XML Document	1 KB
bag7.png	5/1/2019 12:46 AM	PNG File	69 KB
bag7.xml	5/1/2019 1:25 AM	XML Document	1 KB
bag8.png	5/1/2019 12:46 AM	PNG File	100 KB
bag8.xml	5/1/2019 1:25 AM	XML Document	1 KB
bag9.png	5/1/2019 12:45 AM	PNG File	1,509 KB
bag9.xml	5/1/2019 1:25 AM	XML Document	1 KB
bag10.png	5/1/2019 12:45 AM	PNG File	46 KB
bag10.xml	5/1/2019 1:26 AM	XML Document	1 KB
bag11.png	5/1/2019 12:45 AM	PNG File	80 KB
bag11.xml	5/1/2019 1:26 AM	XML Document	1 KB
bag12.png	5/1/2019 12:45 AM	PNG File	45 KB
bag12.xml	5/1/2019 1:27 AM	XML Document	1 KB

CHAPTER 5
RESULT AND DISCUSSION

After running the program a new window will open, which can be used to detect objects in real time.

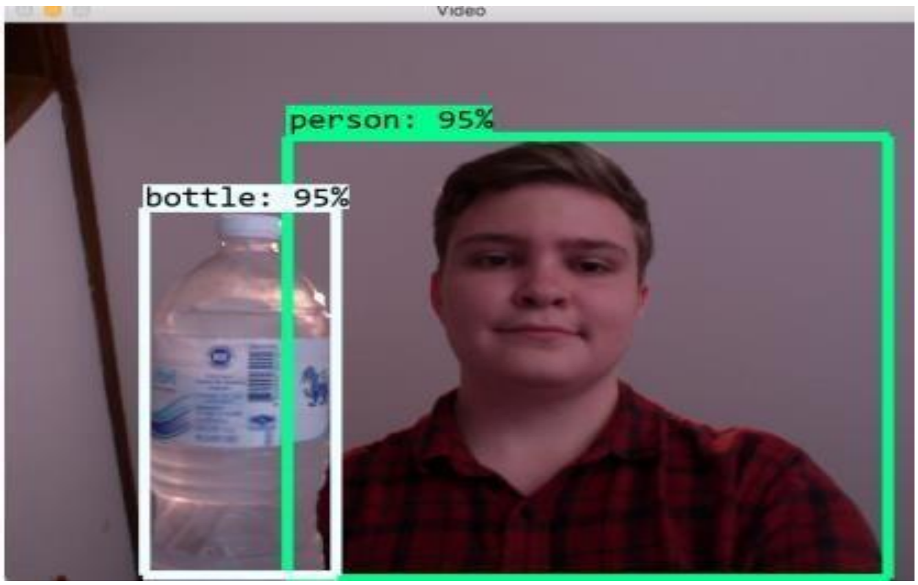
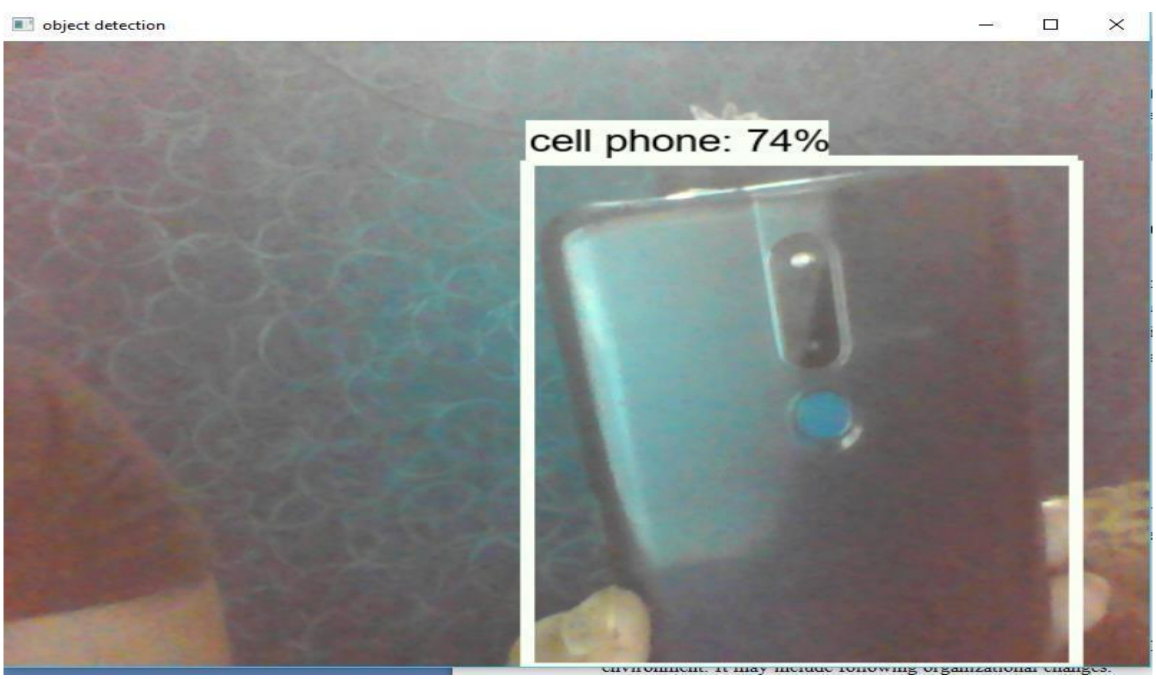


Fig. 5.1



CHAPTER 6

CONCLUSION AND FUTURE SCOPE

The Object Detection system in Images is web based application which mainly aims to detect the multiple objects from various types of images. To achieve this goal shape and edge feature from image is extracted. It uses large image database for correct object detection and recognition. This system will provide easy user interface to retrieve the desired images. The system have additional feature such as Sketch based detection. In Sketch detection user can draw the sketch by hand as an input. Finally the system results output images by searching those images that user want.

Scope of Object Detection and Recognition

The project has wide scope in multiple areas and can easily increase its utilization by adding more efficient algorithms. Some of the areas are as follows-

Medical Diagnose: Use of object detection and recognition in medical diagnose to detect the X-Ray report, brain tumors.

Shapes recognition: Recognize the shape from whole region in images.

Cartography: The cartography as the discipline dealing with the conception, production dissemination and study of maps.

Robotics: In robotics use of object detection is movement of body parts and motion sensing.

References

- He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016), pp. 770–778.
- Hoiem, D., Efros, A. A., and Hebert, M. Automatic photo popup. ACM transactions on graphics (TOG) 24, 3 (2005), 577–584.
- Hoiem, D., Efros, A. A., and Hebert, M. Geometric context from a single image. In Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on (2005), vol. 1, IEEE, pp. 654–661.
- Hoiem, D., Efros, A. A., and Hebert, M. Putting objects in perspective. International Journal of Computer Vision 80, 1 (2008), 3–15.
- Hornik, K. Approximation capabilities of multilayer feedforward networks. Neural networks 4, 2 (1991), 251–257.
- Huang, T. Computer vision: Evolution and promise. CERN EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH-REPORTSCERN (1996), 21– 26.
- Hubel, D. H., and Wiesel, T. N. Receptive fields and functional architecture of monkey striate cortex. The Journal of Physiology 195, 1 (1968), 215–243.
- Ioffe, S., and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. CoRR abs/1502.03167 (2015).
- Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long,