

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY  
BELGAUM, KARNATAKA-590018**



**A Project Report on  
“OBJECT DETECTION IN REAL TIME AND VOICE OUTPUT USING YOLO  
AND PYTTX3”**

**By**

**THANMAI S K (1SZ15CS007)  
ADITHYA (1SZ16CS001)  
PAGADAL KARTHIK (1SZ17CS004)  
PATI SRAVANI (1SZ17CS005)**

In the partial fulfillment for the award of degree of **Bachelor of Engineering** in  
Computer Science & Engineering of the Visvesvaraya Technological University,  
Belgaum during the year 2020-21.

**Under the Guidance of  
Ms. Shalet Benvin  
Head of the Dept.  
Dept. of CSE  
SITAR.**



2020-2021

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING  
SAMPOORNA INSTITUTE OF TECHNOLOGY & RESEARCH  
BELEKERE, CHANNAPATNA Tq, RAMANAGAR Dist-562160**

*Brindavan Education Trust (R)*  
SAMPOORNA INSTITUTE OF TECHNOLOGY & RESEARCH  
BELEKERE, CHANNAPATNA Tq, RAMANAGAR Dist-562160

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**



**CERTIFICATE**

This is to certify that the Project entitled **OBJECT DETECTION IN REAL TIME AND VOICE OUTPUT USING YOLO AND PYTTX3** is a bonafide work carried out by **Thanmai S K (1SZ15CS007), Adithya (1SZ16CS001), Pagadala Karthik (1SZ17CS004), Pati Sravani (1SZ17CS005)** in the partial fulfillment for the award of degree of Bachelor of Engineering in Computer Science & Engineering of the Visvesvaraya Technological University, Belgaum during the year 2020-21. The Project report has been approved as it satisfies the academic requirements with respect to the Project work prescribed for Bachelor of Engineering degree.

-----  
**Ms.Shalet Benvin**  
**Head of the Dept.**  
**Dept. of CSE**

-----  
**Ms.Shalet Benvin**  
**Head of the Dept.**  
**Dept. of CSE**

## **ACKNOWLEDGEMENT**

*I owe my deepest gratitude to Almighty for Everything*

I sincerely owe my gratitude to all the persons who helped and guided me in completing this Project work.

I would like to thank **Dr. B S.M. Naidu, Chairman SITAR**, a well known academician for his modest and helping for all our academic Endeavors.

I are indebted to **Dr. Sampoorna Naidu, Director, SITAR**, for her moral support and for providing me all the facilities during my College days

We would like to Thank **Governing Council** Members of our Organization.

I am thankful to **Dr. H.V Byregowda** Principal SITAR, Channapatna without his help this Project would be dream.

We are thankful to **Ms.Shalet Benvin** Professor & Head of Department of Computer Science and Engineering for his suggestions & Support.

I would like to sincerely thank my Project guide **Ms.Shalet Benvin**, Professor & Head of the Department of Computer Science and Engineering for her invaluable guidance, constant assistance and constructive suggestions for the effectiveness of Project, without which this Project would not have been possible.

I would also like to thank all Department staff members who have always been with me extending their precious suggestions, guidance and encouragement throughout the Project.

Lastly, I would like to thank our parents and friends for their support, encouragement and guidance throughout the Project.

Thanmai S K (1SZ15CS007)  
Adithya T R (1SZ16CS001)  
Pagadala Karthik (1SZ17CS004)  
Pati Sravani (1SZ17CS005)

## **ABSTRACT**

Many people suffer from temporary and permanent disabilities. There are many blind people around the globe. According to WHO it is noted that almost 390 lakh people are completely blind and 2850 lakh people are purblind that is they are visually impaired. For improving their daily life to travel from one place to other place many supporting or guiding system is developed and being developed. So, the basic idea for our proposed system is to design an auto- assistance system for visually impaired person. The disable person will not be able to visualize the object so this Auto-assistance system may helpful for them. Many systems have been implemented to achieve assisting system for blind people. Some system is still under research. Model that were implemented were having numerous disadvantages in detecting the objects. We propose a new system it will assistance the visually impaired person and is was developed using CNN (Convolution Neural Network). In deep learning model the most popular algorithm for object detection is CNN. The accuracy of the object would also be more than 95% which depends on the clarity of the image taken by the camera. The object detected would also be given message for the blind people with the object name detected. This system is a prototype model for assisting blind people. In this system we would be detecting the obstruction in the path of visually impaired person using Web Camera & help them to avoid the collisions. Here we are using object detection.

# **CONTENT\**

<b>CHATERS</b>	<b>PAGE NO</b>
<b>1. INTRODUCTION</b>	<b>1</b>
<b>1.1 GENERAL INTRODUCTION</b>	
<b>1.2 SIGNIFICANCE OF THE DOMIN</b>	
<b>1.3 MOTIVATION</b>	
<b>1.4 OBJECTIVES</b>	
<b>2. LITERATURE SURVEY</b>	<b>4</b>
<b>3. SYSTEM ANALYSIS</b>	<b>11</b>
<b>3.1 EXISTING SYSTEM</b>	
<b>3.1.1 DISADVANTAGE</b>	
<b>3.2 PROPOSED SYSTEM</b>	
<b>3.2.1 ADVANTAGES</b>	
<b>4. SYSTEM REQUIREMENT</b>	<b>14</b>
<b>4.1 SYSTEM ANALYSIS</b>	
<b>4.2 FUNCTIONAL REQUIREMENT</b>	
<b>4.3 NON-FUNCTIONAL REQUIREMENT</b>	
<b>4.4 TOOLS AND TECHNOLOGY</b>	
<b>4.4.1 HARDWARE REQUIRED</b>	
<b>4.4.2 SOFTWARE REQUIRED</b>	
<b>4.5 DEEP LEARNING</b>	
<b>5. SYSTEM DESIGN</b>	<b>24</b>
<b>5.1 SYSTEM ARCHITECTURE</b>	
<b>5.1.1 ELEMENTS OF IMAGE</b>	
<b>5.1.2 PROCESSING IMAGES</b>	

5.1.3	INPUT/OUTPUT DESIGN	
5.2	OBJECT ORIENTED DESIGN	
5.2.1	FLOW CHART	
5.2.2	USE CASE DIAGRAM	
5.2.3	SEQUENCE DIAGRAM	
5.2.4	ACTIVITY DIAGRAM	
5.2.5	DATA FLOW DIAGRAM	
5.3	ALGORITHM USED	
5.3.1	DEEP LEARNING ALGORITHM	
5.3.2	CONVOLUTIONAL NEURAL NETWORK	
5.3.3	KERNELS	
5.4	LAYER TYPES	
6.	SYSTEM IMPLEMENTATION	51
6.1	MODULES	
6.2	MODULES DESCRIPTION	
6.3	FUNCTIONS	
7.	SYSTEM TESTING	58
7.1	TESTING	
7.2	MANUAL AND AUTOMATION TESTING	
7.3	UNIT TESTING	
7.4	INTEGRATION TESTING	
7.5	ACCEPTANCE TESTING	
7.6	TEST CASE	
8.	RESULT AND DISCUSSION	67
9.	CONCLUSION	68
10.	REFERENCES	69

## CHAPTER-1

### INTRODUCTION

#### 1.1 General Introduction

Purblind (vision loss) people are almost in millions around the world among the present population. Their presence in the society plays an important role. Many efforts has been made by the people of different fields to make sure that proper health care is been provided for those people. Many kinds of assisting system has developed and being developed for purblind people which would guide them in their day to day life while they travel inside or outside surroundings.

Advanced technologies like processing of image and computer vision is used for the development of the assisting systems which would provide best performance related to speed and processing. The system developed has to work in real time with great speed and taking action with no time irrespective of the technology used. While the purblind person is travelling at any environment the main aim of the assisting technology is to detect objects, recognizing them and producing an audio alert.

Below figure 1.1 is the image of the analysis of number of people with low vision, blind and purblind per million in all six World Health Organization regions, separately India and China.

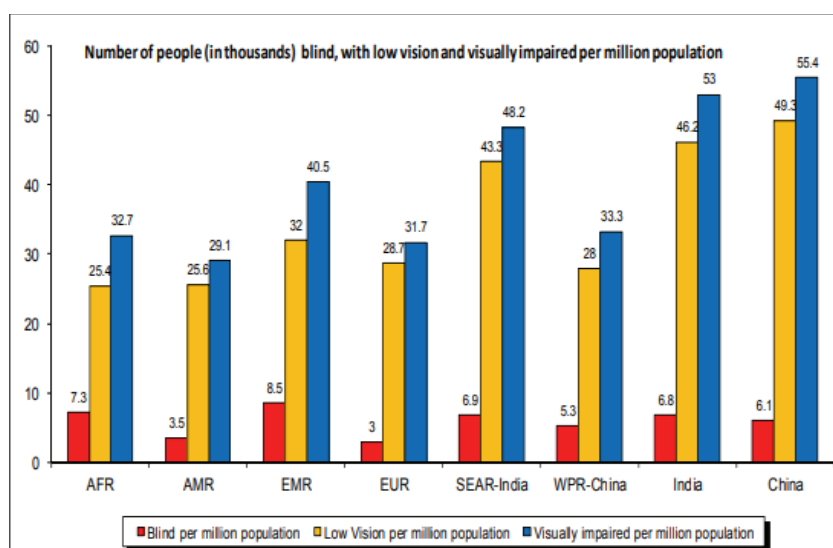


Figure 1.1 Number of people with blindness

The objects that are present at indoor environment like table, bed, chairs etc. should not be near them. The images of the objects can be downloaded or can be captured.

Images are classified by giving them a class label and it is called as localization of object if around the image object there is a bounding box drawn. Combining these two processes and for the object of image assigning a class label for which a bounding box drawn is a process of detecting an object. All the three process together is for recognizing an object.

The approach for detecting objects with more speed is YOLO-You Only Look Once. This method would take image as an input, draw the bounding box and name the class label as this particular approach has a neural network that is single and peer to peer trained. This method would offer less accuracy but operates with more speed. In this approach, the image that is taken as input will be split into matrix of cells for bounding box prediction. By using x, y coordinates along with the height and width and the confidence bounding box will be calculated for the matrix of cell. Based on matrix cell class is also predicted. A simple CNN method was proposed by YOLO which displays result with high speed and good quality. Below figure 1.2 is the image of architecture of YOLO

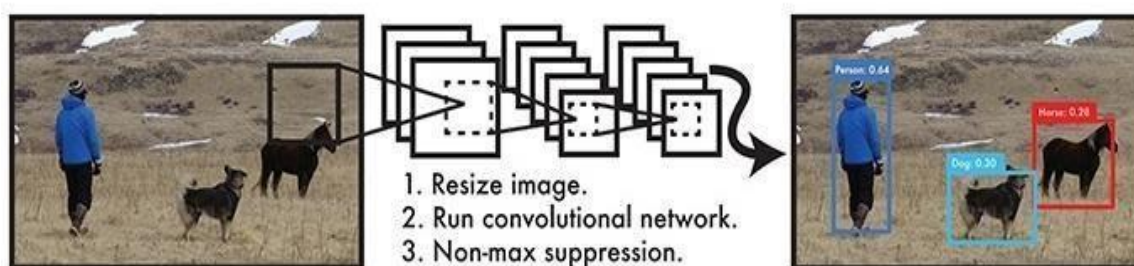


Figure 1.2 Design of YOLO

## 1.2 Significance of the Domain of Working

AI (Artificial Intelligence) mainly aimed for constructing machines and systems which can improve the logical operations under computer science subject. The tasks that are related to human intelligence such as language translation, image recognition, recognizing speech, decision making, vision perception etc can be executed by artificial intelligence. These tasks would be difficult for human being to collect, execute and making decisions. It is nothing but making the systems as human being think and act.



### 1.3 Motivation

Vision loss or completely blind people cannot detect the object or obstacles in their surroundings because of their vision problem. They always need some assisting or supporting system in their life. Solution has been found many years ago for this now gradually the techniques are improving due to evolution and integration in technology. In daily life blind people are using assisting systems that are developed while some are still in the research stage.

### 1.4 Objectives

The project aims to facilitate the movement of blind and visually impaired. The plan defines a platform (vision based) for the identification of indoor objects to guide visually impaired people in real life. Using Python and OpenCV library functions, the software is developed and eventually ported to a laptop.

The main aim of the proposed system is:

- Studying and understanding the present vision module systems.
- Designing of frameworks for the image acquisition system.
- To study how to classify object using CNN.
- Finding objects position in the given input frame.
- Programming both the objects detected and position of the objects to a speech output using text to speech convert.

## CHAPTER-2

### LITERATURE SURVEY

- **Sarthak K, Sanjay K, Ronak S, Samarth G (2018), “Object Detection in Foggy Conditions by Fusion of Saliency Map and YOLO”, 12<sup>th</sup> ICST, IEEE, pp. 154- 159.**

#### Methodology

In the fog time many problems will be caused to human beings because of decrease in the visibility. This may cause accidents on road and risk during driving. So the objects and obstacles need to be detected in the surroundings in this situation. A solution is proposed using YOLO algorithm the saliency map image during fog is detected and a sensor called VESY is used. The sensors are added to the stereo camera to sense the image when fog sensor is activated and to calculate the collision distance map is produced. The image frame saliency generated based on region will be improved in quality using an algorithm called dehaze. The objects detected from the saliency map as well as from the YOLO technique will be given bounding box using the fusion algorithm that was proposed for the real time applications.

#### Merits

- The objects that are present in the foggy image are detected using VESY sensor.
- Detecting objects and recognising objects in fog is done using saliency map.

#### Demerits

- Under foggy situation it is not able to detect all the objects using an algorithm called YOLO.
- For predicting the bounding boxes there is some limitations in YOLO.

#### Conclusion

The saliency map and the bounding boxes had drawn using YOLO for the object at certain threshold is achieved. The objects that are present in fog image are detected. The objects that were not able to be detected by YOLO technique can be detected by a technique called VESY.

- S Kun, Hayat S, Tengtao Z, Tu T, Y Du, Yu Y (2018), “A Deep Learning Framework Using Convolutional Neural Network for Multi-class Object Recognition”, 3<sup>rd</sup> ICIVC, IEEE, pp. 194-198.

### **Methodology**

In the field of system vision certain technologies are used for detecting and recognizing complex work with help of the features of detecting techniques. The objects that are recognized present in the image is done using object recognition technique. Many researches have been done from many scientists from many years in several areas for the effective detection and recognition of the objects. Using deep learning these methods are adapted. In proposed system for multiclass object and CNN they have used deep learning technique. First initialization is done and later system is trained using nine different categories of objects that are trained along with the dataset of sample images for testing to create CNN. A framework, tensor flow is used to implement the output. This CNN system has an accuracy of 90.12% when compared with BOW method.

### **Merits**

- CNN accuracy is better when compared with an approach BOW (Bag of Words) that has five different objects images.
- Competitive approach and need less computing time.

### **Demerits**

- Caltech-101 datasets has very few limited categories.

### **Conclusion**

A heuristic method was acquired by CNN for the recognizing of objects of multiclass to improve their performance. The performance of recognizing was improved and the system was tuned further. Nine differing objects are chosen from Caltech-101 image dataset and a CNN is deployed with 5 layers.

Different traditional BOW methods with five differing objects of class are compared based on the performance with the proposed system. The performance of the model proposed in tested to be better and 90% accurate than BOW methods.

- **Aishwarya S, Kaiwant S, C Anandji, Tatwadarshi N (2018) “An Innovative Machine Learning Approach for Object Detection and Recognition”, 2<sup>nd</sup> ICICCT, IEEE, pp.1008-1010.**

### **Methodology**

In computer science the technology that thinks and reacts as a human beings intelligence system is AI (Artificial Intelligence). Human beings have a capacity to recognize and detect the objects since they can distinguish and find out through their eye. This is not the case with machines and they cannot find out. So this issue can be overcome by NN (Neural networks) also known as ANN (Artificial Neural Network). Many researches are on process in the area of detecting and recognizing objects. This research is based on motion of objects ie dynamic objects. On static objects the proposed system applies detection and recognition.

### **Merits**

- For set of ten classes 90% of accuracy and for 20 set of classes 75% accuracy achieved.
- Good processing speed.

### **Demerits**

- Accuracy decreases as and when increase in the size of sample.

### **Conclusion**

In the proposed system faster RCNN approach is used for detecting and recognizing objects. This approach would produce the result with a great accuracy and good processing speed. Many processing approaches are applied when images are given to the model. Above 90% accuracy for 10 set of classes of image and above 75% for 20 set of classes of images are obtained. The system will be trained on huge dataset as to overcome with the problem that is if image size increases then the accuracy of the detecting of object will be reducing.

- **Jeong H J, Park K S, Ha Y G (2018) “Image preprocessing for Efficient Training of YOLO Deep Learning Networks”, BigComp, IEEE, pp.635-637.**

### **Methodology**

The most challenging in AI is big data training. For detecting and recognizing the data from from the crawler are not processed images means they cannot be used as a data for training.

Hence a preprocessing model is built for refining the data downloaded from crawler. All the images are collected from the spiderbot (crawler) for training. The model is an preprocessor for image in training using YOLO. From the spider the objects are downloaded and saved in other image. There can be one or many objects in the spiderbot and the objects that are present in image along with size, location and class it is explained.

**Image picker:** Crops the region that is annotated in the object class of image.

**Modifying scale:** Decreases the size of cropped object to exact size.

**Making image:** Altered image object will be fixed to base image. Creating annotation: In base image size of fixed objects will be reduced.

#### **Merits**

- The object that is present in image is detected effectively.
- Accuracy of the detected object is more.

#### **Demerits**

- Only 6 classes are used.

#### **Conclusion**

The system is proposed for preprocessing the images that are trained which are downloaded from the spider bot for YOLO. Two key factors are important. The detected image and the training image size must be similar this is first point, second the image in the training and detected must be same with the area of proportion of object image that is occupied. Four steps are designed for generating training image appropriately: Image picker, modifying scale, making image and creating annotation.

- **Yawei H, Huailin Z (2017), “Handwritten Digit Recognition Based on Depth Neural Network”, Track 2 ICIBMS, IEEE, pp. 35-38.**

#### **Methodology**

In the area of processing the image NN (Neural network) and DN (Depth network) is mostly used. For the network systems that are complex the output must be with good recognition.

These complicated network system takes a huge time in training and it would be difficult. BP system and CNN is introduced with Mnist dataset to recognise with good and simple model. Later the proposed system with combined DN is introduced for recognition. This shows that combined DN is better than a simple network for recognition of data.

**Merits**

- The obtained recognition rate is high.
- The result of combined DNN is 99.55% optimal compared to other simple networks.

**Demerits**

- Time taken for training the network is too long.
- In simple CNN the recognition rate is limited.

**Conclusion**

To obtain a high recognition rate for network system, combined DNN is introduced. The result obtained by the combined DNN is 99.55% optimal with the mnist dataset. The system proposed a BP neural network that is double and hidden for obtaining high recognition rate. The recognition rate output when compared with other single NN is improved.

- **Malay S, Rupal K (2017), “Object Detection Using Deep Neural Networks”, ICICCS, IEEE, pp. 787-790.**

**Methodology**

An advanced CNN approach was proposed in this system called as RCNN. In this system first the images are divided into numerous regions and later for that region convolution will be applied. There are 3 phases or steps in RCNN:

In the first step the image is divided and categorized into individual regions. The second step is, for every convolution network the weights and layers required are calculated that is called extraction of feature. In the third step, the convolution networks with the help of images with label are trained. The training process is classified into 3 steps:

First step is that the images will be trained using conventional CNN where for an image there is just one object this is supervised pretraining. The second step is CNN layers works with domain level which is domain specific. The third step is objects are positioned at their respective classes and categories this is called category classifier of objects.

**Merits**

- Transfer learning gives accuracy.
- Defined architecture can be used.

**Demerits**

- Less number of objects.
- Time consuming.

**Conclusion**

At the basic level the proposed system RCNN is much optimized. But the output of the system is approved at particular parameter. In future the parameters can be enhanced or new parameters can be used for achieving low error rates that this approach.

- **Xinyi Z, Wei G, Wenlong F, Fengtong D (2017), “Application of Deep Learning in Object Detection”, 16<sup>th</sup> ICIS, IEEE, pp. 631-634.**

**Methodology**

In the application of DL for detecting objects the system vision deals with this proposed system. The system has a brief about the datasets and DL techniques that are usually used in the system vision. An rcnn algorithm is used to deal with the newly created dataset. Learning the importance of DL and dataset is achieved by the analysis of the output and for strong understanding of networks experiments are conducted.

**Merits**

- Recognizing accuracy will be increased.
- As per experiment the DL techniques are effective tool for large data.

**Demerits**

- In fact the deep learning can be affected because of the data quality.

**Conclusion**

The application of DL technology and the usage of new dataset using faster rcnn are expressed. From many years the tasks on system vision based on classifying image, identifying face and detecting object got big success. Feature that is done by man rely on the experience to the data drive which shows efficient tool is the technology of DL. The development of DL applications would be difficult when many applications accumulate continuously the data of application. Instead of original data some artificial data can be considered to raise the data quantity.

- Tianmet G, Jiwen D, Henjian L, Yunxing G (2017), “Simple Convolutional Neural Network on Image Classification”, 2<sup>nd</sup> ICBDA, IEEE, pp.721-724.

### **Methodology**

Deep learning is being used in estimation of pose, classifying image, detecting text, detecting object, recognizing object, detecting saliency of visual and many more in current years. In DL the most commonly used technologies are CNN and DBN (Deep Belief Networks). Out of these many DL technologies, the model that provides good performance on classifying image is CNN. In the proposed system for classifying an image a simple CNN is demonstrated. The datasets that are used for performing experiments in the system are cifar-10 and also mnist. With the help of CNN for classification of images the parameters that has to be solved optimally is done by optimal algorithm and different techniques for learning is analyzed.

### **Merits**

- Simple network is designed.
- Parameters use less memory.

### **Demerits**

- Recognition rate is low.

### **Conclusion**

A simple network for classification of images is proposed. This network consumes low cost for computation. For classification on images different learning techniques and algorithm for solving parameters optimally are also proposed. It is verified that there is a great recognizing effect on the network that is shallow also. Even if the rate of recognition is not good compared with existing network, proposed system parameters would consume less memory.

- Liu Shuying, deng Weihong (2015), “Very Deep Convolutional Neural Network Based Image Classification Using Small Training Sample Size”, 3<sup>rd</sup> ACPR, IEEE, pp. 730-734.

### **Methodology**

A many DCNN (Deep Convolutional Neural Network) techniques were developed by many researchers. Huge dataset like ImageNet is used for training DCNN in the existing system.



As deep network overfit easily cifar10 kind of datasets that are small takes advantage infrequently. A new VGG16 system is developed in the proposed system and cifar10 is used to for fitting into this network. Without overfitting cifar10 into the model an error-rate is achieved by using regularizer and normalization of 8.45%. The dataset that contains label of the data is the only dataset that is Imagenet.

**Merits**

- For adopting small datasets deep model is used.
- By regulating overfitting was reduced.

**Demerits**

- When new data will be evaluated the performance will be poor.
- Between training and testing calculation held are different.

**Conclusion**

By the usage of Batch Norm and dropout for small dataset a deep model has been adopted. To gain accuracy also these two settings are adopted which would generate high accuracy. In future other methods can also be used for adopting small datasets.

## CHAPTER -3

# SYSTEM ANALYSIS

### 3.1 Existing System

In Existing System, the objects are detected and it is assisted to the completely blind people to make their daily life comfortable. In this system algorithms such as Convolutional Neural Network (CNN) and also Haar Cascade are compared with each other based on the object detection. Some of the entities like cup, person and ball were used in experiment for detecting and classifying. The algorithm used for the detection of face is Haar cascade which is a basic algorithm and for detection of objects CNN is the basic algorithm. This system was built only for detecting objects and the comparison was done between the algorithms which were not a favour for blind people. There is no any message for the blind people if any object or obstacles are identified.

#### 3.1.1 Disadvantages

- Contains only 3 classes (person, ball, cup)
- So, if there are any other obstacles visually impaired person can't identify.
- Text to speech is not available (identified object is not dictated to person)
- Using Haar Cascade algorithm accuracy will be less.

### 3.2 Proposed System

We propose a new auto assisting system which will identify more than 3 classes from the video frames. So, the person can identify more obstacles in front of their way and avoid them. This makes the auto assisting system for visually impaired people more meaningful and helpful. After detecting the objects from the video frame this system will speak what object is detected. Here text-to-speech conversion is done so this system is really a boon for visually impaired people.

In the image the objects of some classes are located and identified using object detection which is a mostly used system vision. The identification of the objects that are available in the images is called detecting objects. The capacity of computer and software to identify each object by locating them in an image or screen is object detection. It is widely used for tracking objects, pedestrian detection, self driving cars, face detection and security systems. There are many other fields where object detection can be used for. As every other computer technology, extensive range of innovative and surprising uses of object detection will come because of the efforts of programmers and software developers.

Here we are using trained objects which are trained using YOLO framework and CNN algorithm for training the captured image. The image can be captured using webcam or can be downloaded. The data must be equally balanced data. The dataset collected will be divided for training and testing purpose. Here preprocessing is done. Then the system is tested by converting the identified object to pytts (python text to speech). So, our system will be useful for blind people. Object detection includes two main aims:

- Identifying all the objects that are available in the image.
- The objects that are focused will be filtered.
- This proposed system will be great boon for visually impaired person.

### **3.2.1 Advantages**

- Text to speech facility is available.
- Many objects are used for detection.
- Comfortable and safe.
- N' number of objects can be trained.

## CHAPTER-4

# SYSTEM REQUIREMENTS

### 4.1 System Analysis

#### ➤ Feasibility Study

This particular analysis is performed for checking how the idea can work better in an environment so that the model is technically and economically feasible and usable. Some of the projects may not be worth investing money, by this study it would justify whether the system is worth or not. Some of the projects do not worth investment like they may need multiple resources which may affect other resources from working and the organization should spend more money than it takes back by that project. The study that is designed must contain the history of the project like about development of project technically and the implementation of project.

#### ➤ Technical Feasibility

This study main aim is to focus on the resources technically. This study would help the organization to decide whether the tech team are able to develop the resources into a working model and the usage of the resources in feasible way. The estimation of hardware, programs and requirements for the system is done in this stage. The model proposed uses less resources which is affordable for any person or organization and technically the system is feasible.

#### ➤ Economic Feasibility

This study as the name says the system proposed must be feasible in cost. Before the resources are allotted financially the system must meet the criteria like benefits of the system and the system ability to work and the cost to propose the model. The benefits must be positive economically so that the organization can release the money for the development of the project. The model since contain less resources it is feasible in cost so that any people can afford it easily.

### ➤ Operational Feasibility

The feasibility determines how good the proposed system is for operational purpose and how the project meet the requirements of the customer once the model is developed. At the stage of analysis of the requirement done by the development team all the plan for building the project must be analyzed. Once the system is done it must meet all those requirements and must be able to use with satisfaction. The requirements for designing a prototype model is analyzed and developed as it satisfies all the needs that were analyzed during the analysis stage. The system can be operated easily and it is affordable by the organization.

## 4.2 Functional Requirements

The function of a model and its respective components is defined as functional requirement. The input given and the output received this particular relationship is called as function. This would involve some of the technical information, data calculating and some of the processing functions. These requirements are the services that the system will offer us.

- How the Webcam is used to capture the image.
- Software that is required is Python 3.0 version, must be installed.
- The output generated is explained.
- System workflow should be explained.
- Packages or library functions that are required in python language must be installed.

Example:

- To install numpy package, type on the terminal: pip install numpy
- To install scikit-learn package, type on the terminal: pip install scikit-learn
- 

## 4.3 Non Functional Requirements

The requirement would be specifying certain measures for the operation of a model instead of specifying particular behaviour. This will tell how the model should be and it is also called as attribute of a quality of a model. The complete properties of this requirement specify whether the system developed is passed or failed.

**Usability:** The model must be allowed for using only by the particular users so that the main aim of the system proposed can be achieved efficiently in a specified manner.

The model has to be simple and understandable for the people who are using that system. Some of the blind people may not understand if the instructions are difficult. The instructions must be easy for them to assist. In that way, system does not contain any complex steps while using in their environment.

**Reliability:** The model should perform its specific functions under any circumstances without undergoing any problem or failure. The system must also perform functions at given time of interval under any situation. This would ensure for the user that the system can be reliable and purchased. For blind people the obstacles when identified by the system it should dictate that name by not delaying and failing to avoid accidents and it is achieved.

**Performance:** The system undergoes with the performance test to get to know exactly how the model is working so that if any error or warning occur it can be managed in early stage before it reach the client. This enables how to manage the user expectation on the system and how to make a plan for it. The system must be satisfied by the user in all aspects, budget, and no defect, feasible for changes. For blind people system is low budget and is feasible to train N number of objects.

**Supportability:** The model should be supportable for maintaining and for any repairs even after the system is delivered to user. The design and the requirements should be related and supported to the user requirements. The design of a model should be easy and affordable for users.

**Maintainability:** At any point of time the proposed model for maintaining must be easy. It is the process of restoring or regaining the data at any condition. The model must be able to undergo repairs and changes while it is working. If there is any defect in the model it should be improved as early as possible.

**Flexibility:** The system must be flexible for the user to use it and it should get adapted to the client in easy way once it is produced. The system must be developed in a way as it should be able for any changes in future according to the user requirement. The system proposed is flexible for blind people where it would be identifying the obstacles and it would produce an audio alert for the blind people and can be changed in future.

## 4.4 Tools and Technology

### 4.4.1 Hardware Required:

- CPU : i3 processor
- HARD DISK SPACE : 500GB
- MAIN MEMORY : 4GB
- Any laptop or desktop with higher configuration.

### 4.4.2 Software Required:

- OS : Any version of windows supported (Win 10)
- Programming Language : Python 3.7.7
- Software : Python IDLE
- Libraries : OpenCV, imutils, pytts

#### ➤ Python

- The syntax of python is mentioned clearly, less keywords and the structure is also not complex. This language can be quickly learned.
- Python language can be maintained easily and is described clearly.
- There are many libraries which is easily portable and is compatible with many operating systems.
- Interactive mode is also supported which allow to perform manual testing and the code can also be debugged.
- Supported on different hardware platforms widely and the interface is similar for all platforms.
- For python interpreter some of the modules that are low level can be introduced which would allow the programmer to add more tools and use them efficiently.
- For multiple businesses related database interfaces are provided by python.
- Applications like GUI are also supported by python which can be built and ported to numerous operating systems, libraries and system calls.

- Huge programs are supported by python better than in shell programming and the interface provided is also good structured.
- OOP concepts, methods and structured way of programming are also supported in python.
- For developing big applications python can be used as script and can obtain byte code after compiling.
- Data types that are dynamic are also provided by the python which also support checking data type dynamically.
- The garbage collection is done automatically in python.
- Can also be integrated with other programming languages easily (C, C++, Java, Com, ActiveX and Cobra).

#### ➤ Libraries

**OpenCV:** This is called as open source computer vision library. As the name suggests it is a library which is used for processing images. OpenCV is available for free. OpenCV library is programmed using C or C++ because of these programming languages it is fast. This particular library uses less memory space and also portable. To make use of the functions that are available in this library we need a compiler. First the OpenCV must be installed and compiler must be installed and then link must be created between them.

**NumPy:** NumPy is main library function in python. A great-performance, numerous dimensional array object, and for working with those arrays tools are provided by numpy. For the computation of python scientifically numpy is the main package. Using numpy random datatypes can also be declared this would allow to get integrated with numerous databases. Numpy provide good performance related to speed. They require less space. They have some algebra functions built in.

**Scikit-learn:** This is a software library function that is available free in python. This particular library is used for classifying, dimension reduction, clustering and regression. It is a simple and great library for analyzing data.

**Scikit-image:** This function is for processing the images and it contains algorithm collections and some utilities. The charge and the restrictions are available freely. The images to be filtered can be done easily using scikit-image.



**Imutils:** A function that is benefit for basic processing of images functions like translating, rotating, reshaping, images of matplotlib getting displayed, skeletonization, sorting, edges getting detected and many more.

**Matplotlib:** This particular library is used for plotting. This matplotlib is used when the images are getting analyzed. If histograms of images are getting plotted or just the images are getting viewed it is good to have a matplotlib in toolbox as it is a great tool.

**Tensor flow 2.0:** In machine learning peer to peer open source tool is tensorflow. This platform has a complete flexible environment tools and libraries. The tools are powerful and easy to use. It helps to get deployed on any platform. It provide a platform for building datasets that work with 'N' dimensional arrays and constructing basic NN and DL models. For constructing new NN tensor flow can be used directly. For constructing standard NN, tensor flow can be applied along with the front end keras which is a package with tensor flow.

Installing tensor flow:

- pip install tensorflow

**Keras:** This API is related to neural network which is high level and widely used. Keras is programmed using python and has a capability to run above tensorflow, CNTK or Theano. To enable experimentation with large speed this keras was developed. The idea of going to the result from idea with very less time is the important aim of the keras. It is easy and speed is more to use, user friendly and can be extensible. Supports both the CNN and RCN and run on CPU and GPU without any effort.

**Pytts:** Python text to speech is a library function in python. It writes the spoken audio data into file for further usage. Unlimited text length is read. It would correct the pronunciation. It will retrieve automatically all supported languages.

## **Pytts**

Pytts (Python text to speech) is a library which is operating system independent and also a text to audio library that is cross platform. Using this particular library text can be converted to audio message in offline. The python version 2 is supported by pyttsx. For the newly modified versions pyttsx3 is designed which support both the versions 2 and 3 with the same programming.

For installing pytts:

- Pip install pyttsx3 Using the pytts library:
- init (driverName string, debug bool)

The name of the driver that is available in the operating system should be the first argument and for debugging output should be the second argument in the init function.

For speech output:

- say (text Unicode, name string)

Once the initialization is done by passing the arguments with the init function we make use of say function to produce the text output. This particular function contains two arguments: text that should be output and name of that text output.

To execute the speech output:

- runAndWait()

The interpreter first must execute runAndWait function to get the text speech as an output.

The say function would not work unless and until runAndWait function is executed.

## 4.5 Deep Learning

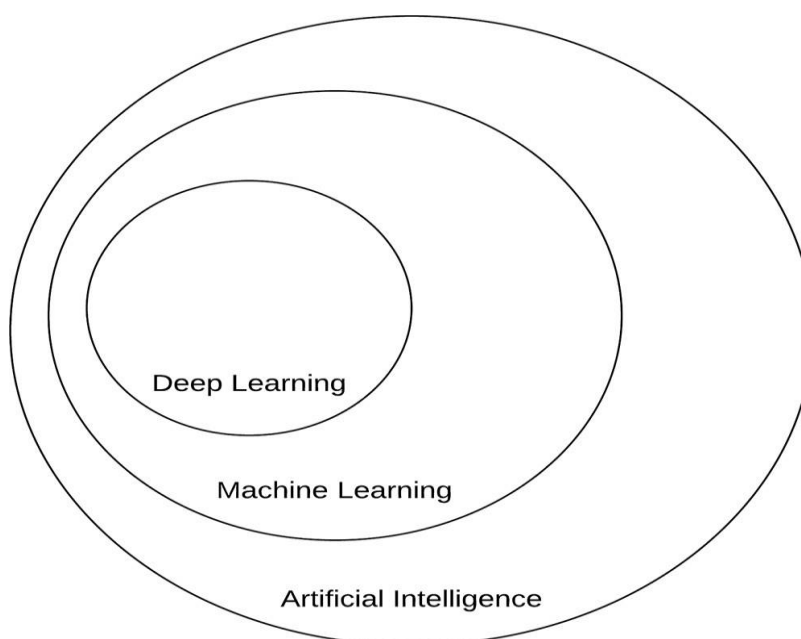
In machine learning, deep learning is a subfield, whereas ML is the subfield of AI (Artificial intelligence). The graphical representation of this relationship is in the below figure 3.1. The main aim of AI is providing certain set of methods and techniques which can be applied in solving problems that can be performed near to human activities automatically or else they would be very difficult for computers.

In the AI for solving problems best example is thinking and reading the stuffs present in the image. This particular work can be done by the people with no difficulty while it is difficult and not the same case for the machines to do that work without effortless. The machine learning field is specially used for recognizing and detecting the pattern and from data studying is also done, whereas AI comprises of certain works like planning, interpretation, reading etc performing automatically.

For learning from data and performing specialization in recognizing patterns comes under ANN (Artificial neural network) which is a subsection of ML and perform operations as the human does and as human brain works.

In the ANN section even a DL (Deep learning) comes under which is used for the same case as of ANN and their names can also be interchanged while using. The technique DL has existed for many years (60 years) but it was used all around the globe by different names depending on the research and the field in which DL was used, based on the datasets, hardware and software and based on researcher opinion.

To know the history of DL, first we will see for a NN (Neural network) what has made to be a deep neural network (DNN) and we will also discuss about the learning concepts based on the hierarchy and how it did DL as a famous technique in this modern century in field of ML and system vision.



**Figure 4.1: Venn diagram displaying relation between DL, ML and AI**

The system or machine will learn in performing tasks by classifying the work from images, words or audio in DL. The accuracy level can be obtained by DL better than a human being accuracy level. NN algorithms which has numerous layers and data with huge set that is labeled is required for training the system.

- **Reason for using Deep Learning**

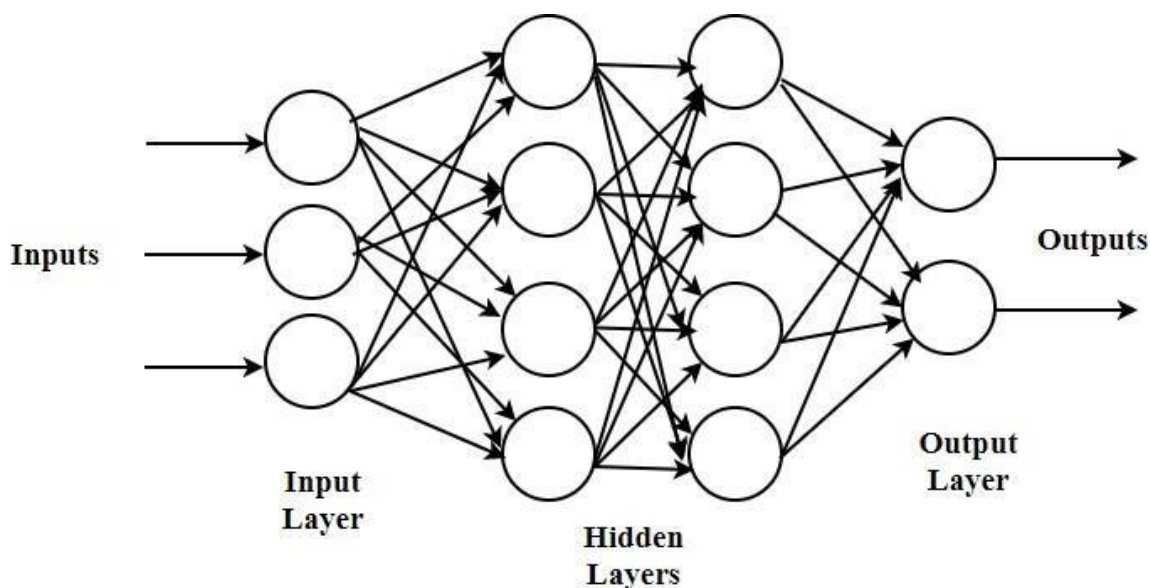
When the issue is about correctness of image recognition it can be done better using DL as it would achieve a great accuracy for recognizing objects. The reason accuracy is more important is example of driverless car, in this application accuracy is main and this would make the consumer to trust the machine and it would meet the consumer expectation for purchasing. The researches that have made in the recent time has proved that DL is better and accurate in image classification than compared to humans. The theory of DL was originated in 1980, nowadays it is popular and used by many people for two reasons:

- For designing and developing a driverless car lakh of images and hundreds of videos with many hours of data is needed. This means huge amount of data that is labeled is needed for DL.
- The computational power should be good performance for DL. GPUs must be designed with an architecture that is parallel which is effective in DL. The training would take lot of time but when it is mixed with cloud computing the training time can be reduced for DL network, it could take just hours for system that was taking weeks.

#### **4.5.1 Working of Deep Learning**

NN (Neural network) designs are used by many DL algorithms that is the reason for systems built by deep learning are called as DNN (Deep Neural Network). In DNN the letter D is 'deep' which means the total of middle layers that are hidden in NN. In old neural network there were only some two or three middle hidden layers whereas in DNN there can be any number of hidden layers as 150. In DL extraction of feature is done by the network as the system will be trained with huge dataset that are labeled and using NN technique which can learn the system features and no need of any manual interference.

In many techniques the regularly used DNN technique is CNN or also called as ConvNet. The 2-dimensional data like images are suitable for processing as CNN uses layers related to 2D input and they are learned about the features by CNN when it convolves with data as an input.



**Figure 4.2 Neural Network with hidden layers**

For classifying the images CNN would be identifying features there is no need of any involvement of manually extracting the features. For this CNN would extract objects from the images. Once the images are collected the system must be trained at that time the objects extracted would be trained here no any pre-training is performed by the network. In classifying the objects under system vision the extraction of features using deep learning network can be done with automation accurately.

There may be numerous hidden layers using which CNN would be detecting multiple objects related to image. The objects of image that has been learned will increase their difficulty in every middle layer. Let us take an example, In the middle layers the 1<sup>st</sup> hidden layer will learn detecting about the edges and the last layer will learn about complicated designs and shapes that are trying to get recognized. For the problems, DL that have been trained previously with the NN models is applied by extracting features and by learning to perform transfer. The features that are related are automatically removed from the images using DL technique. The data is given to the system and work has to be performed like classifying this is done in DL and called as 'peer to peer learning' where the network learns automatically.

## CHAPTER-5

### SYSTEM DESIGN

#### 5.1 System Architecture

The design part includes the system architecture. It explains the workflow of the system proposed. The architecture mainly explains the data is being modified. How it is being used and how the results vary with it.

The scene for vision will be captured at different sampling rates. The images that are captured and acquired would undergo processing and that output would trigger an audio message for the person, the audio message will depend on the object detected. This is shown briefly in below diagram.

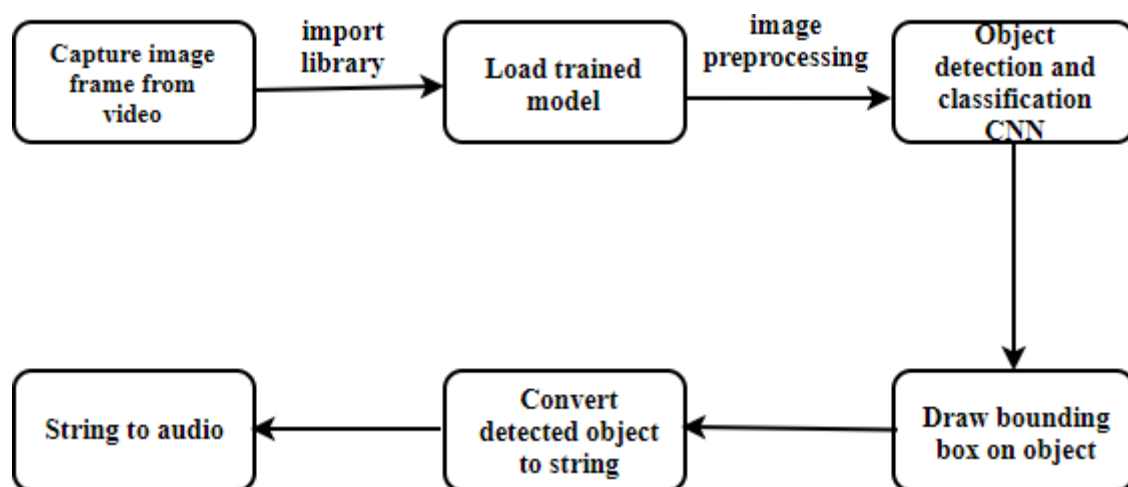


Figure 5.1 System Architecture

- The images are captured as a frame from the video. This is the first step and the respective images may be shades of grey image or combination of color image.
- The model will be trained using the libraries that are imported to the system and this particular model will be loaded to system.
- The images can be of different sizes so they are preprocessed, means the images size are rearranged, rotation is done if required and shape is rearranged if required.
- All images must be maintained with similar size.

- CNN algorithm is used for detecting objects and for classifying objects.
- The objects are converted to a string by drawing a boundary box for those objects that are detected and classified.
- The string that was generated will be further converted in to an audio message using the pytts.
- Later the result would be the audio of the object that was detected through speakers.

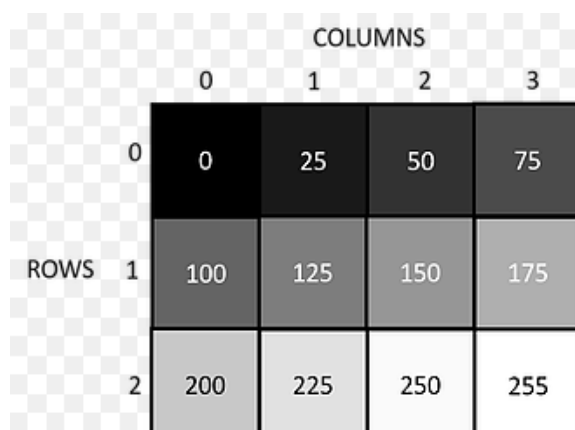
### 5.1.1 Elements of Image

In the memory the storage of images is done in multiple spaces of color. In that color space the most commonly heard by everyone is RGB which is used by the Win Os to the maximum. RGB would require other color system that is suitable for application in order to perform processing of image.

#### ➤ Grayscale Image

There must be an information about the intensity of particular pixel brightness in grayscale image. If the pixel values are more than the intensity of the image would be more. There are total of 0 to 255 shades in gray color system. Each of the pixels are little less brightness than the other one. This can be represented in the below figure 4.2. Each pixel in the grayscale would be occupying 1 byte which is all required and it would store from 0 to 255 pixels which is all the shades.

The grayscale system in the storage is denoted as a 2D array byte. The h and w (height and width) of image would be same as the array size. This array created is a channel where grayscale has one which would denote the white brightness.



		COLUMNS			
		0	1	2	3
ROWS	0	0	25	50	75
	1	100	125	150	175
	2	200	225	250	255

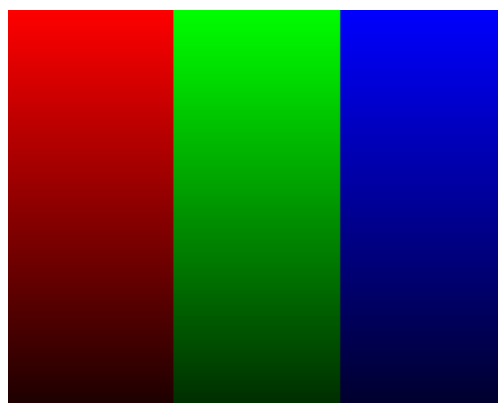
**Figure 5.2: Grayscale Image representation**

### ➤ Color (RGB) Image

Each pixel which is of three bytes are splitted into three parts: each byte for each color (red, green and blue) these colors are the primaries which will allow to get different colors by mixing up with each other in a correct proportion.

RGB color system too has multiple shades of each color which is of 0 to 255 and each byte can be storing these shades values. To obtain the color user wanted to have all the three colors are mixed based on the proportion of the color required. This color system is inbuilt. This is used by everyone without our knowledge.

Each pixel byte is allocated for each color and all these three colors are combined to with their bytes to get required color which is called dedicated. All these dedicated shades of colors are allocated in separate channel. Below figure 4.3 represents color (RGB) image.



**Figure 5.3 Color (RGB) representation**



### ➤ RGB and BGR Ordering

OpenCV stores RGB channels in reverse order. While we normally think in terms of Red, Green, and Blue, OpenCV actually stores the pixel values in Blue, Green, Red order. Why does OpenCV do this? The answer is simply historical reasons. Early developers of the OpenCV library chose the BGR color format as this particular format is very famous in producers of camera and in the time of developing software. It's a small caveat, but an important one to keep in mind when working with OpenCV.

### 5.1.2 Processing of Image

Any image it is comprised of pixels which is 2D array with digits that ranges from 0 to 255. These pixels can be represented as a function of 'f' with the horizontal axis as 'a' and vertical axis as 'b' i.e  $f(a,b)$ . In the place of this  $f(a,b)$  the value given would be the value of the pixel of an image.

For applying preprocessing of image on dataset some steps are applied on the image dataset. The steps applied are as follows:

- Reading the image
- Resizing of image
- Denoise (Noise removing)
- Segmentation
- Edge leveling (Morphology)

#### Step 1: Reading the image

In this step, import all the libraries that are required for reading the images then the path of the dataset of image will be stored into one variable that was generated for loading folder which contains array of images.

#### Step 2: Resizing of image

Two functions are created for displaying the image and for visualization of change. The function created first is used for displaying one image and the other functions is for displaying two images.

Once this is done one more function is created which would receive image as an argument it is called processing function.

### Step 3: Denoise (Noise removing)

For removing the unwanted noise and to level the image within processing function this particular block of code will be added. Unwanted noise may be disadvantage for processing the image.

### Step 4: Segmentation

The images are segmented that is the objects that are present in the background and in the present screen are separated. To improve the segmentation in better way the noise present in the image can be removed.

### Step 5: Edge Leveling (Morphology)

In this step, by the help of markers the objects are separated from the image. The unwanted noise and shape would be smoothened which results in good texture of the image.

## 5.13 Input / Output Design

### ➤ GUI

The system consists of two GUI (output screen). The input is the object of the image which would be captured as soon as the system executes. The captured image name would be the acoustic message and the object name with accuracy would be displayed on the output screen. Below is the figure 4.4 of the output screen which represents a GUI.

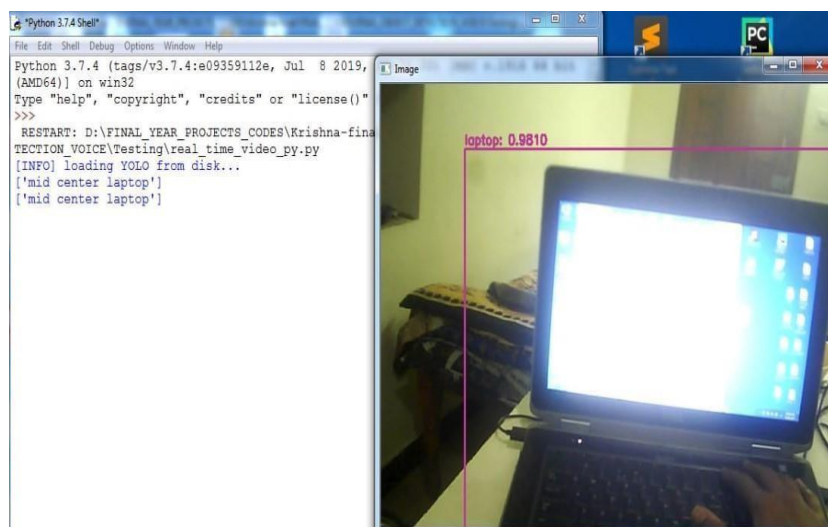
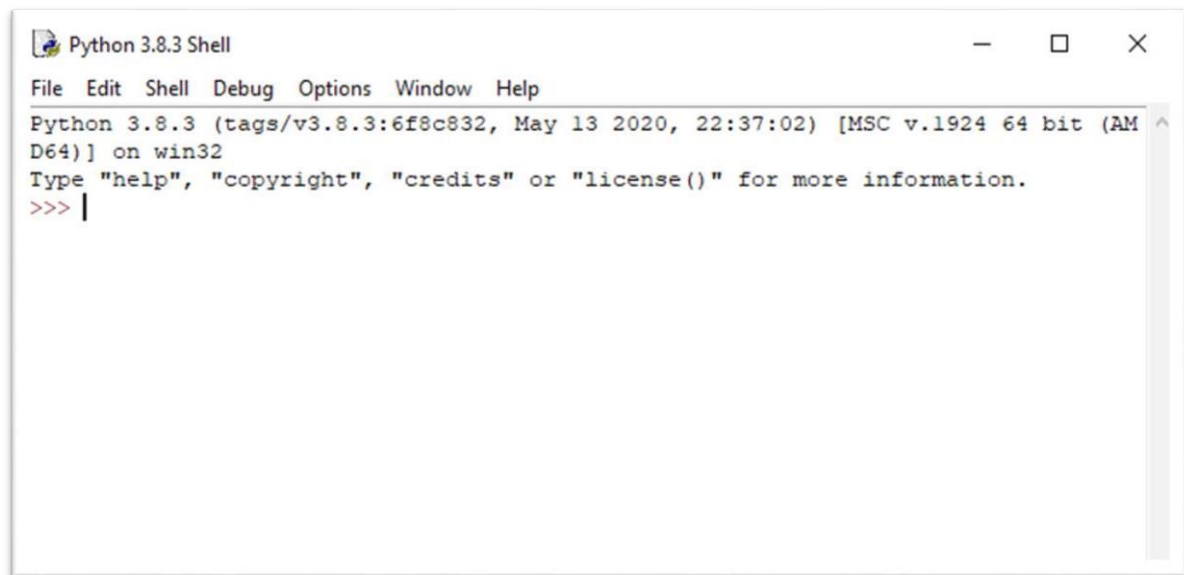


Figure 5.4 Output screen (GUI)

The editor or the GUI that is used for developing the code to build the system is also a GUI. The editor used is python IDLE which is a GUI for the development of the python. This would allow user for editing, executing and debugging the python code in a simple environment. Below is the figure of the python IDLE.



**Figure 5.5 IDLE screen (GUI)**

Most important features of IDLE are:

- It is a text editor with multiple window option for highlighting syntax, indentation and many other.
- Highlighting the syntax using python shell.
- Debugger that is integrated with breakpoints, stepping and visibility of call stack.
- Simple and suitable for beginners.

#### ➤ Non GUI

A block of code that can be reused and can be included in the projects or software is the python library. The python libraries are not related to any particular framework in python as when compared with other programming languages like C and C++. Library is defined as a group of main modules. For installing package library, a package manager is used which is used to install.

These libraries can also be called as a non GUI. Below are the lists of some of the

libraries used for designing the system.

- OpenCV
- Imutils
- Pytts

## 5.2 Object Oriented Design

During the detailed phase, the analysis of the application to be developed during the high- level design is splitted into modules. Every design will be having a login design and will be documented as program specifications.

### 5.2.1 Flow Chart

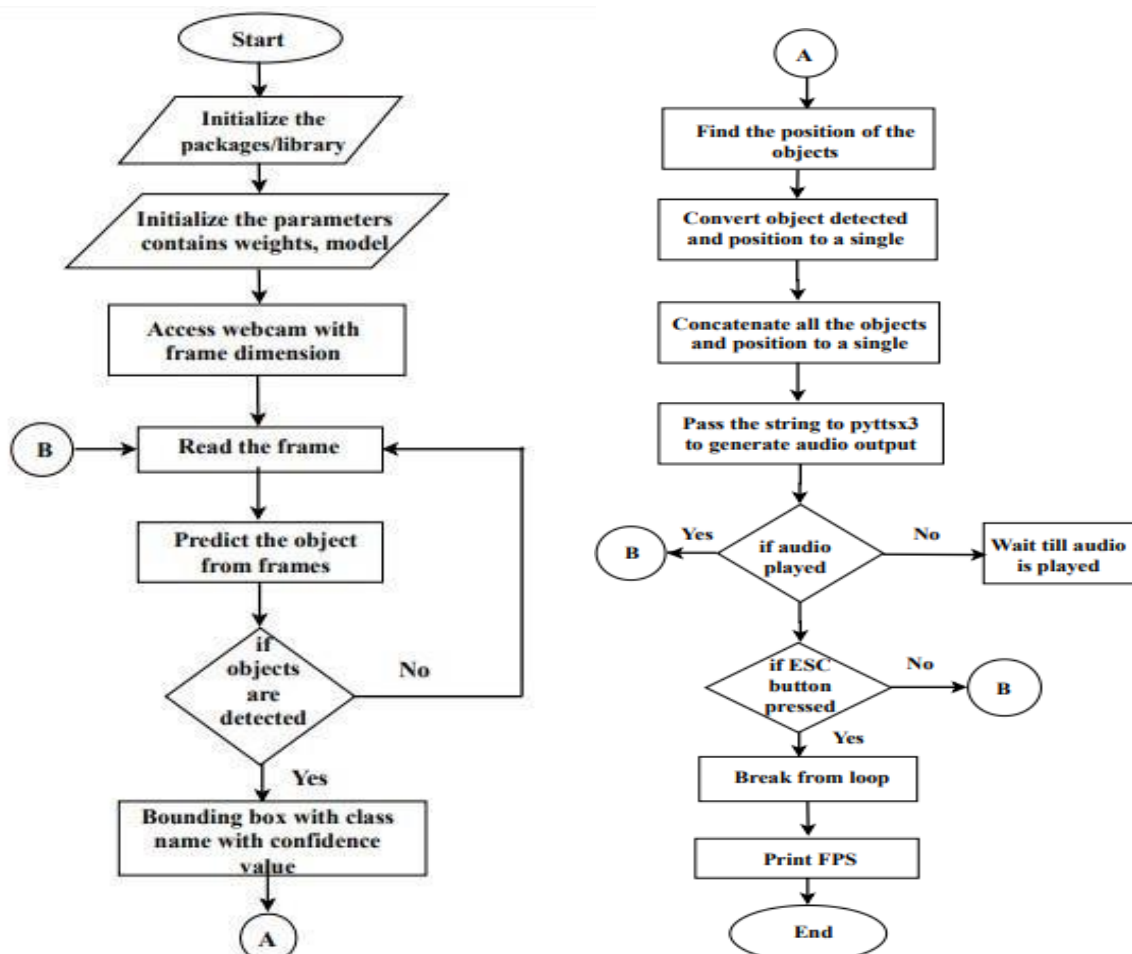


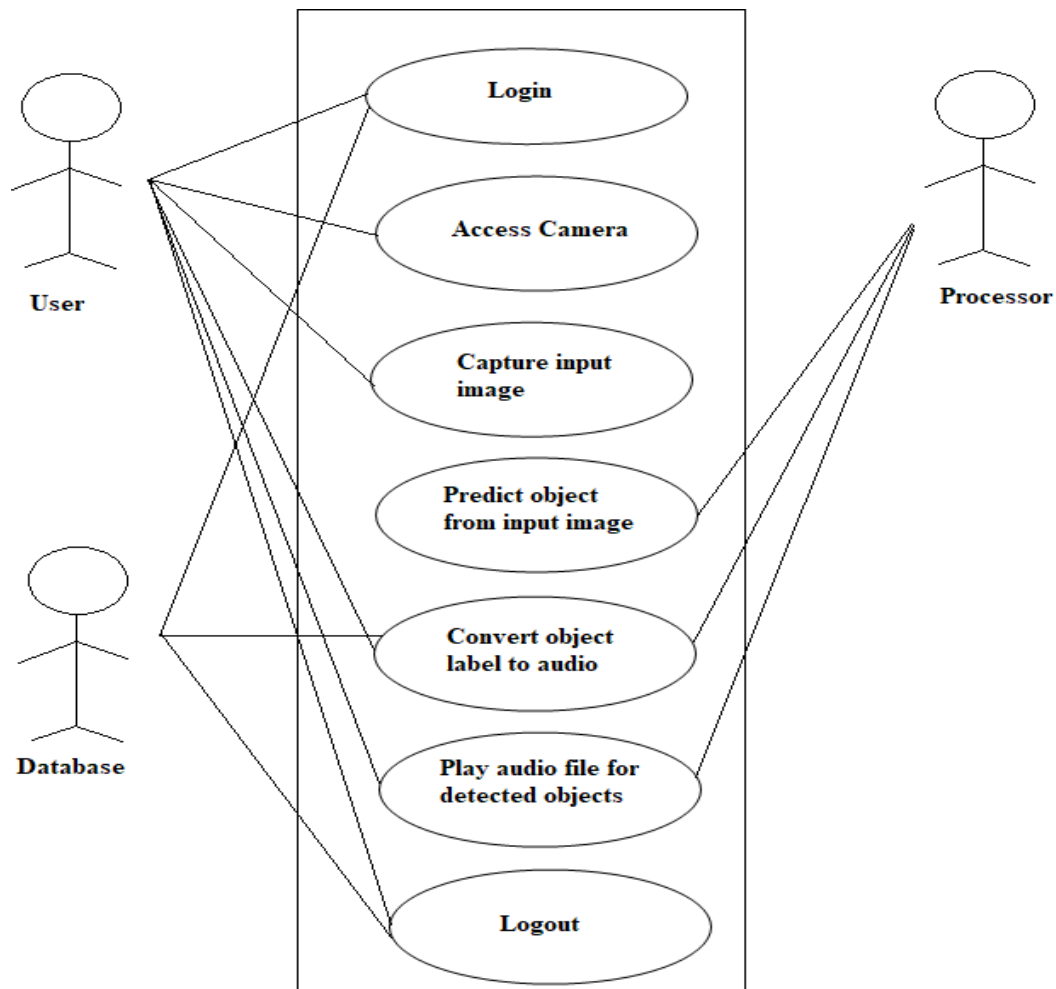
Figure 5.6 Flow Chart of System

The steps that are represented in a graphical manner is flowchart. The algorithms, flow of work and processes are represented in a step by step linear way using this flowchart. Each of the steps are represented in a boxes shape that is different and they are connected using the arrows from one box to another.

For the information to get displayed and for assisting based on the reason's flowchart is the most important part. In visualizing complicated processes or making the structure of jobs and problems explicit it is used. For defining the process and for implementing the project flowchart can be used. The representation of flowchart of system is in figure 4.6

### 5.2.2 Use Case Diagram

If use case is designed once it can denote the representation in text and video way this is only called as use case. If the design is required as per the view of the user or customer this use case diagram would be perfect example for that case. The behavior of the system that are visible would be specified by the . user externally for making the system to communicate and this is an effective method. Below figure 4.7 is the use case diagram of the model.

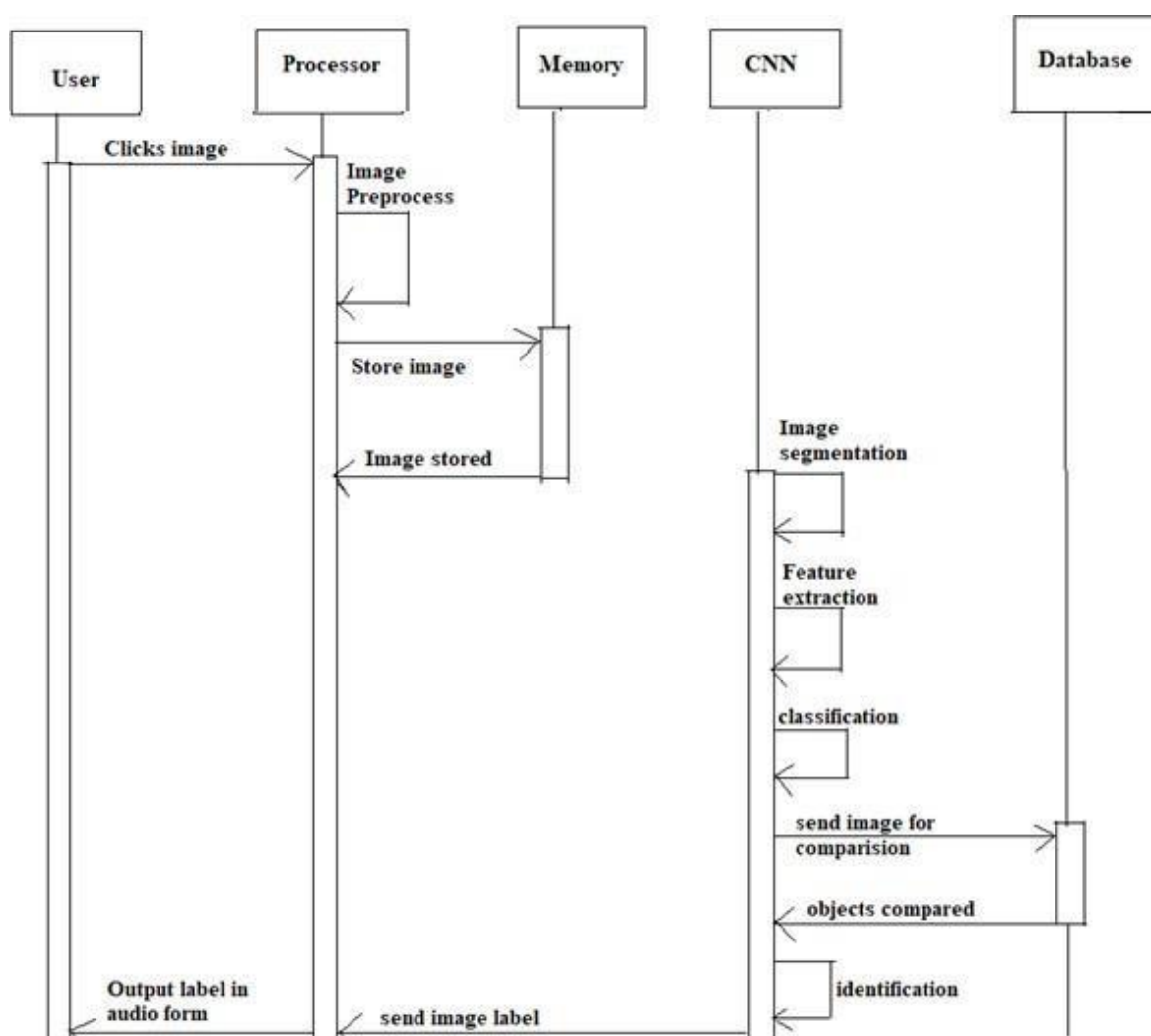


**Figure 5.7 Use Case of System**

- User must be logged in to system to access camera.
- Images are captured by user using webcam available in the laptop.
- Processor would predict the object from the captured image, preprocess the image.
- The label is converted to the audio message and is stored in the folder which can also be called as database which will be used by user to play audio.
- The audio file is played which would read the detected object name.

### 5.2.3 Sequence Diagram

The diagram 4.8 that shows the interaction and details of the operations how it is performed is sequence diagram. For collaborating with the context, the interaction between the objects are also captured clearly. The interaction procedure is displayed visually with straight line and the time is also represented using arrows which represent when and what time messages are transferred.

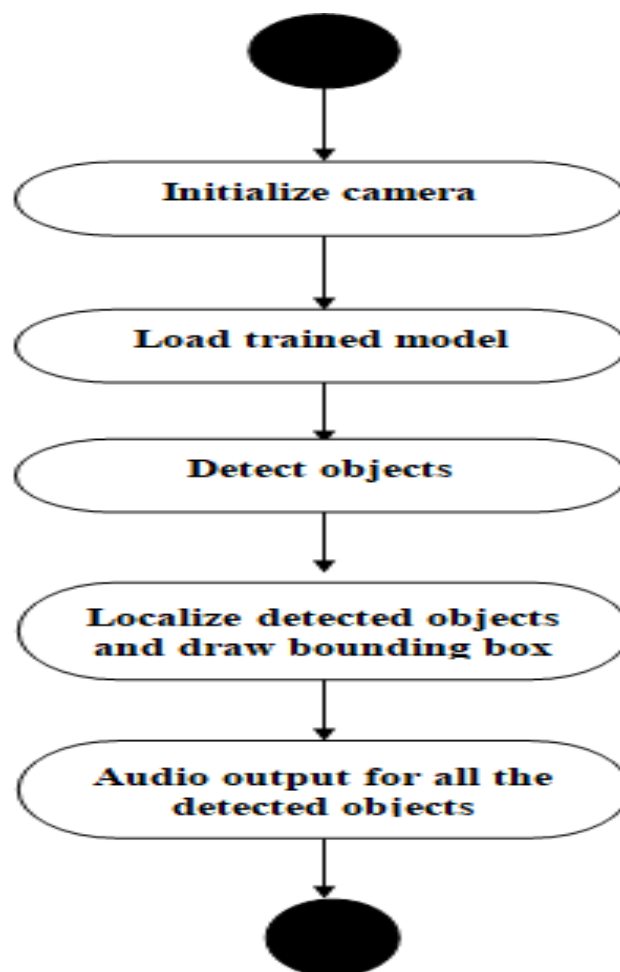


**Figure 5.8 System Sequence Diagram**

From the above figure 4.8 it is seen that the image captured by the user is sent to the processor for preprocessing and the preprocessed image is sent for storing in to the memory Using CNN image segmentation, extracting features and classifying objects are performed. The images are later sent for classification to the database and it is sent back for identifying objects in the image. The label of image is sent as an audio output for the user.

### 5.2.4 Activity Diagram

The activities that are at various level of abstraction can be provided with a service with the activities that are coordinated, this is shown in activity diagram. The event which has to be attained using some procedures is proposed for achieving numerous things for which coordination is required or should describe the relation of one event with the other event and case where the activities would intersect and need management. These functions are performed by the activity diagram.



**Figure 5.9 Activity Diagram of System**

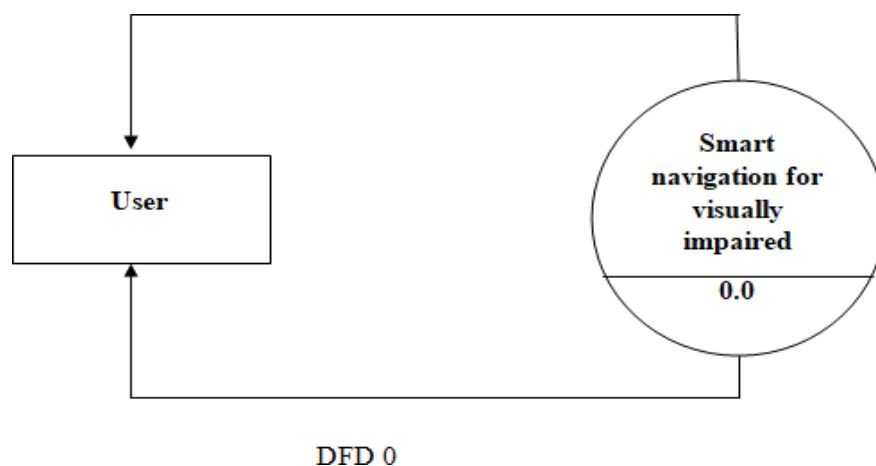
The above figure display the flow of activities held in the system. The camera is initialized and the images that are trained are loaded into the system. Objects are detected from the images that are loaded onto the system. Bounding box is drawn for the detected objects which display the label of the image detected and finally the name would be the output as an audio message.



### 5.2.5 Data Flow Diagram

The flow of data or information from one process or model to other is represented in DFD (Data flow diagram). For representing incoming, outgoing and storing points circle, rectangle symbol is used and also for the route from source to destination will be displayed using symbols like arrows along with these small text labels are also used. The flowcharts of data can be either simple with written or drawn from hand representing overview of program or can be complex with multiple levels which represent deep about how data is maintained and used. For analyzing the system that is existing or proposed one DFD can be used. DFD is same as other diagrams and flowchart which would tell that is difficult to explain this would be worked out by technical (developer) and non-technical (CEO) also. The DFD are less used for some of the system like based on database and real time software. There must contain at least one incoming and outgoing data in DFD. For one data that is stored it must contain at least one flow in both the directions. Data must pass through process and process must pass through storage or other process.

#### ➤ DFD Level 0 Diagram



**Figure 5.10 Level 0 DFD**

The context figure or the level 0 DFD is same and it is the figure which would explain the overview of the network that is being designed. This is designed in order to make the system understand easily as explaining the system would be difficult. Many people related to technical and non tech would be referring to this figure for the better understanding of the system or model as in figure .

### ➤ DFD Level 2 Diagram

The context figure is further divided into more level such as level 1. This particular Level 1 DFD defines how the functions are flowing from one system to other system. Some of the functionalities and information is provided by this level 1 DFD as in figure 4.11.

- Context figure will be explained in more detailed way.

The most important procedures performed by the network is emphasized and it will be divided into sub function.

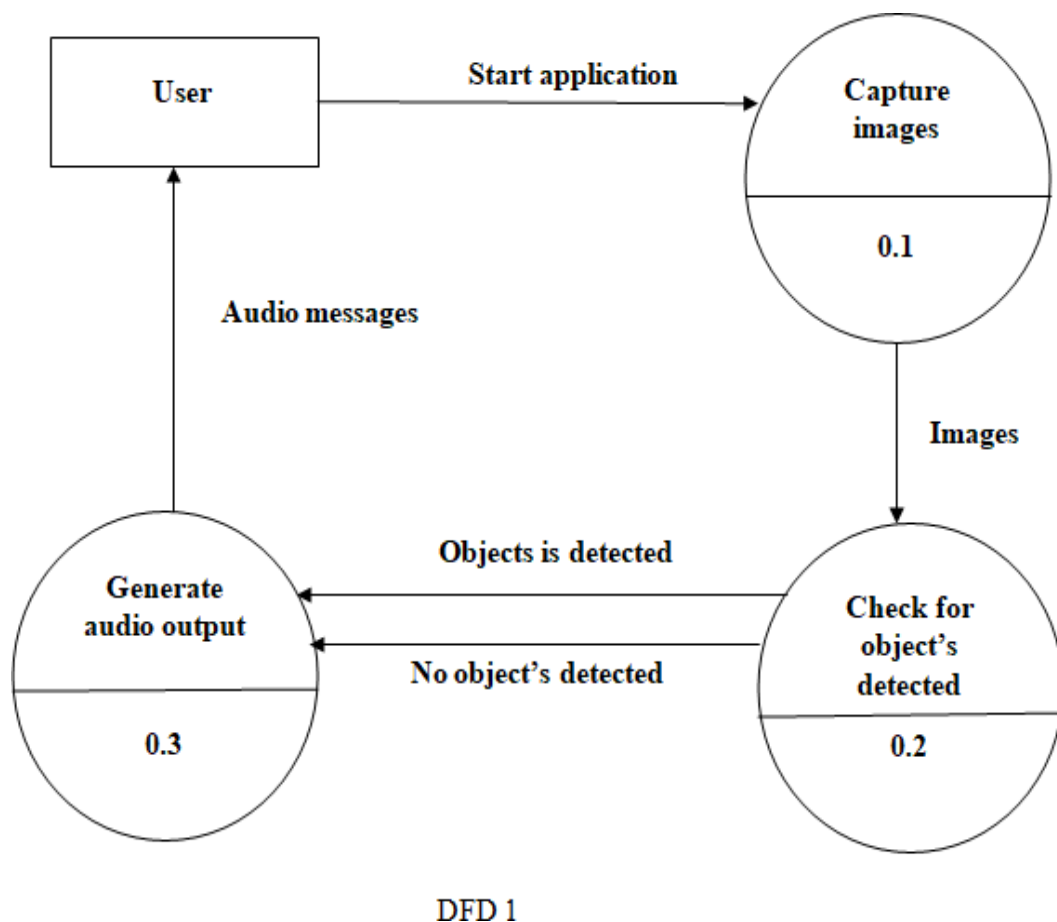


Figure 5.11 Level 1 DFD

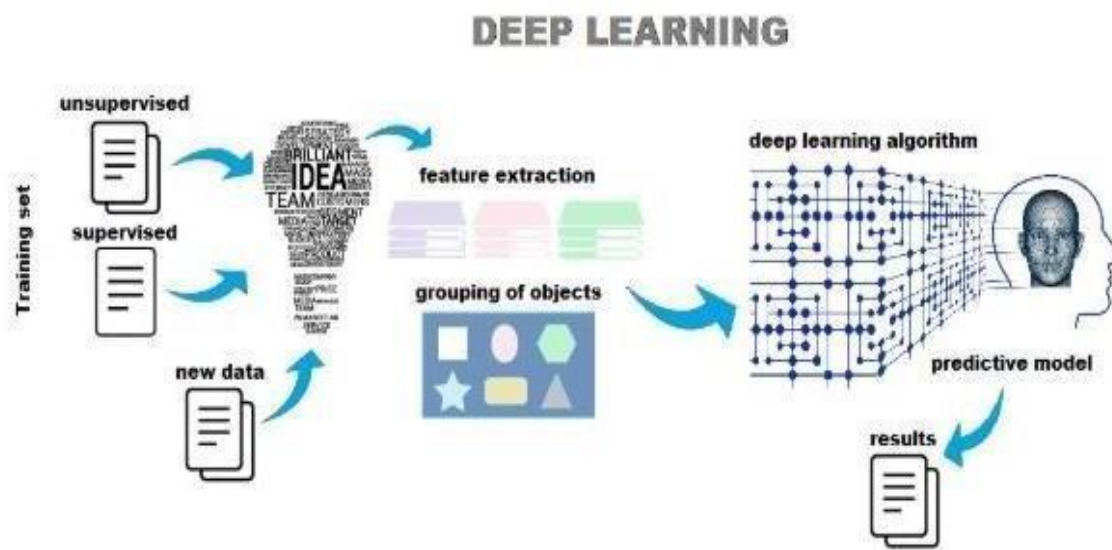
### 5.3 Algorithm Used

To build the system certain algorithms are applied which are explained as below:

### 5.3.1 Deep Learning Algorithm

Depending on the data input the information is decrypted on different layers which was encrypted by using layers of NN technique done by DL. Let us take an example of recognizing an image app in which the features like edges that are sharp or less contrast is identified by one of the layers, other layers can be used in identifying separate shapes.

After the first two layers then 3<sup>rd</sup> layer can be used to decrypt to see what is in image. By learning from the previous layers by differentiating objects these layers can be achieved. The architecture based on DL that has been using nowadays are mainly depended on ANN which would use many layers of non-sequential processing for extracting feature and alteration. Below is the figure 4.12 of deep learning.

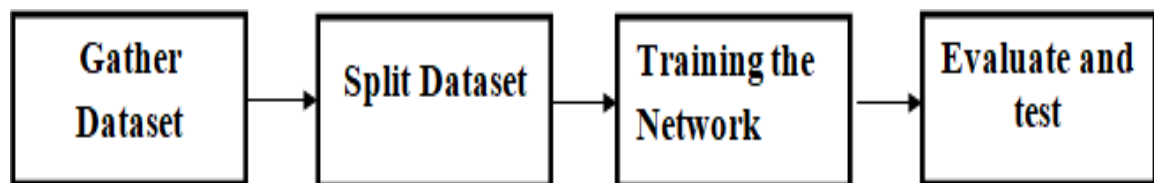


### Figure 5.12 Deep Learning

The representation of learning on own is DL technique feature which depends on ANN as it would represent same as human brain calculate data. While distributing the input for extracting features and patterns the elements are used that are not known when the training is going on.

Same as machines that are getting trained to learn by themselves would arrive at many levels which uses the techniques to construct the network. DL techniques uses numerous algorithms.

But none of them are perfect, while few are better and well suited in performing tasks. For choosing correct one understand the techniques and methods.



**Figure 5.13 Algorithm Deep learning**

First the dataset is collected either from the website available or captured from the webcam of the laptop (or any camera). The dataset would be splitted into training and testing dataset with a specific percentage 75% for training and 25% for testing in the system. Then the system is trained to detect the objects that are collected and splitted for the training. The trained objects are later tested and evaluated whether the output obtained is valid or not valid.

### **5.3.2 Convolutional Neural Network**

In CNN every layer would apply certain filters kind of tens or hundreds or may be even thousands and the result would be combined producing the result for further layer in the system. The numbers for filters would be learnt by CNN in the time of training automatically. For classification of images CNN should learn for:

- Detecting edges in the 1<sup>st</sup> layer from raw pixel of data.
- Using these edges for detecting shapes in the 2<sup>nd</sup> layer.
- Using these shapes for detecting features that ae in high level like face feature, car parts etc in the next further layers.

### VGG16 and VGG19

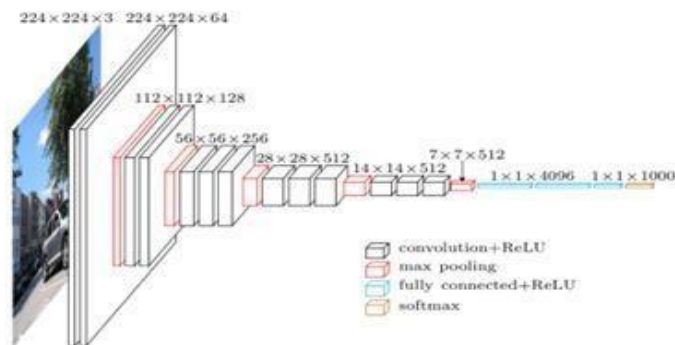


Figure 5.14 VGG Architecture Visualization

The features that are in high level would be used by CNN in final layer for making predictions based on the image contents.

In terms of DL, Conv is an image which is the product of two matrix which is followed by the addition.

Two matrices with similar dimension.

- Number by number perform multiplication.
- Add them together.

#### 5.3.2.1 Kernels

Considering image to be a huge matrix and kernel to be a small matrix as shown in below figure 4.15

From the above figure it can be explained that the small matrix is slide horizontally and vertically with the real image. Once the x and y axis are passed the neighbor elements are examined which are resided at the middle of the small image.

131	162	232	84	91	207
104	<del>91</del>	<del>109</del>	<del>+11</del>	237	109
243	<del>-2</del>	<del>202</del>	<del>+23</del>	135	26
185	<del>-15</del>	<del>200</del>	<del>+13</del>	61	225
157	124	25	14	102	108
5	155	16	218	232	249

**Figure 5.15 Kernel as a tiny matrix visualized**

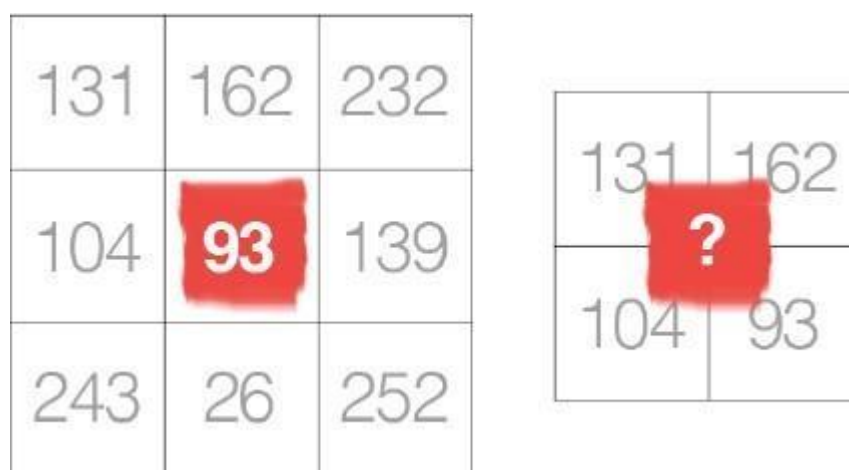
This neighbor elements are later taken and they are convolved with small matrix to get result as a single result. The value of the result obtained will be stored in the image for output in the same x and y axis in middle of the small matrix (kernel).

There is another example which explains the same procedure but in different manner. In this the kernel looks different as shown below.

$$K = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

For ensuring about the valid number of x and y axis at the middle of the image size of the kernel used is in odd number as shown is figure.

3\_3 matrix is represented on the left image while the middle of matrix is at 1 and 1 based on x and y, the origin is represented at the left side on top corner of matrix and axis is mentioned as 0. 2\_2 matrix is represented on the right image while the middle of matrix is at 0.5 and 0.5 based on x and y axis. For locating a pixel on the matrix first interpolation has to be performed and the pixel axis must be in digits.



**Figure 5.16 3\*3 kernel with pixel at centre (Right), 2\*2 kernel centre is what? (Left)**

As of this reason the size of the kernel is in odd size which would make sure that the axis x and y is having valid values at the point of the small matrix.

Convolution Example:

The kernel matrix discussed is done and now the discussion is the actual convolution operation and see an example of it actually being applied to help us solidify our knowledge. In processing of image, a convolution needs three elements:

Image that is input.

- For applying to image of input a small matrix is required.
- The result of the image that is convolved with the small matrix is stored in an output image.

Conv (i.e., cross-correlation) is really very easy to do so:

- From real image the x and y axis is selected.
- The point of the small matrix is placed at that x and y axis.

- The numbers in the region of image and in the small matrix are multiplied and are added to obtain a single number. The addition of these products are kernel result.
- From 1<sup>st</sup> step the same x and y axis are used and the result of small matrix is stored at that x and y axis of the result of image.
- The convolved matrix that is 3\_3 of image along with 3\_3 small matrix which is used for distorting:
- Once the conv is applied the pixel that is at the axis of i and j of the result image would be set into R to  $R_{ij} = 132$ .

$$O_{i,j} = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \star \begin{bmatrix} 93 & 139 & 101 \\ 26 & 252 & 196 \\ 135 & 230 & 18 \end{bmatrix} = \begin{bmatrix} 1/9 \times 93 & 1/9 \times 139 & 1/9 \times 101 \\ 1/9 \times 26 & 1/9 \times 252 & 1/9 \times 196 \\ 1/9 \times 135 & 1/9 \times 230 & 1/9 \times 18 \end{bmatrix}$$

Therefore,

$$O_{i,j} = \sum \begin{bmatrix} 10.3 & 15.4 & 11.2 \\ 2.8 & 28.0 & 21.7 \\ 15.0 & 25.5 & 2.0 \end{bmatrix} \approx 132.$$

The examples explained would justify that convolution is the addition of numbers in the matrix that is the product of the small matrix and the neighbor of the small matrix which covers the image that is input.

## 5.4 Layer Types

For constructing CNN many kinds of layers are used but the most commonly and regularly used layers are:

- Conv (Convolution)
- Act (Activation or RELU)
- Pool (Pooling)
- Fully\_Connected (FC)
- Batch norm (BN)
- DO (Dropout)



Stacking a series of all six layers in a certain method would results in a CNN. We often use simple text diagrams for describing a CNN:

Image input => Conv => Act => FC => Softmax

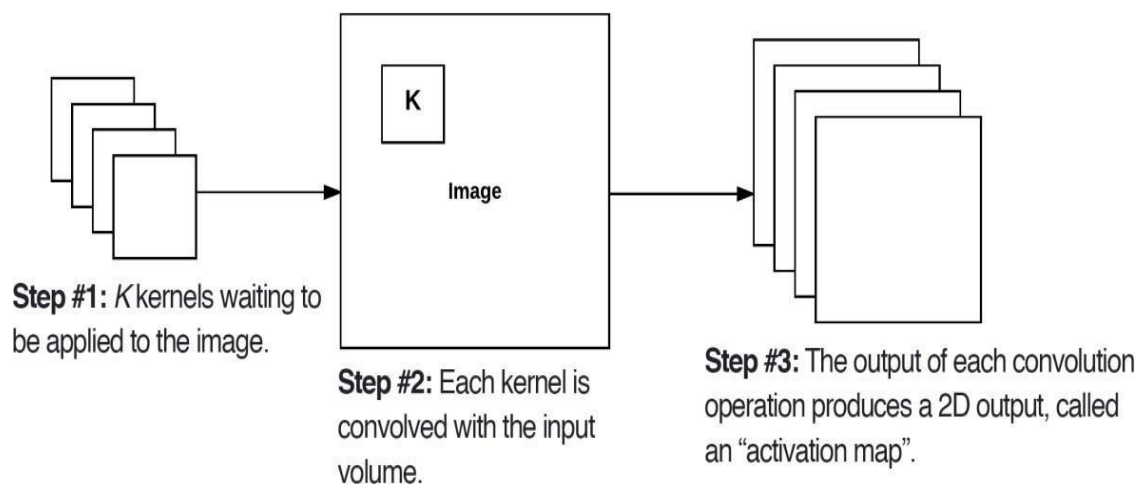
A simple CNN which would receive an input and for this input conv layer is applied which is followed by act and then Fc and finally classifier using softmax function for obtaining the probabilities of the classifying the input. When the softmax function in the act layer follow the last FC layer then the diagram related to network is obtained.

When there is a training process going on the layers Conv, Batch norm and FC are learnt as they have parameters for learning. Whereas the act and do layers are not said to be a valid layer but they are considered for system diagrams for making clarity in the architecture. The images that have extent in the dimensions have more impact and for these images pool layer is also used with the same importance as given to the conv and FC layers and it is used in the system diagrams as it passes through the Conv.

Pool, Conv, Act and FC are considered to be the main layers when the real system architecture id getting defined. It doest not mean that other layers are not difficult but when compared these layers are difficult as they are used for defining the technique.

#### ➤ **Convolutional Layers**

In CNN main layer for constructing architecture is the Conv layer. There is some 'k' set of filters related to small matrix with Conv layer arguments in these filters there is a length and width which form square. When compared to the dimension these filters are little but when compared to extent, they are complete in the depth. In CNN the input is given as the image that contain depth in the channel with any numbers (i.e., a depth of three when working with RGB images, in this one depth for every channel). The volume that are cavernous in the system, this would depend on the filters that are used in the earlier layers. In CNN 'k' filters are convolved from length and height of the volume of the input which is done in further pass of CNN. To make it simpler each 'k' small matrix is slide towards region of input by performing computation on numbers as a product and adding then finally storing the result in 2D act map as shown in figure 4.17.

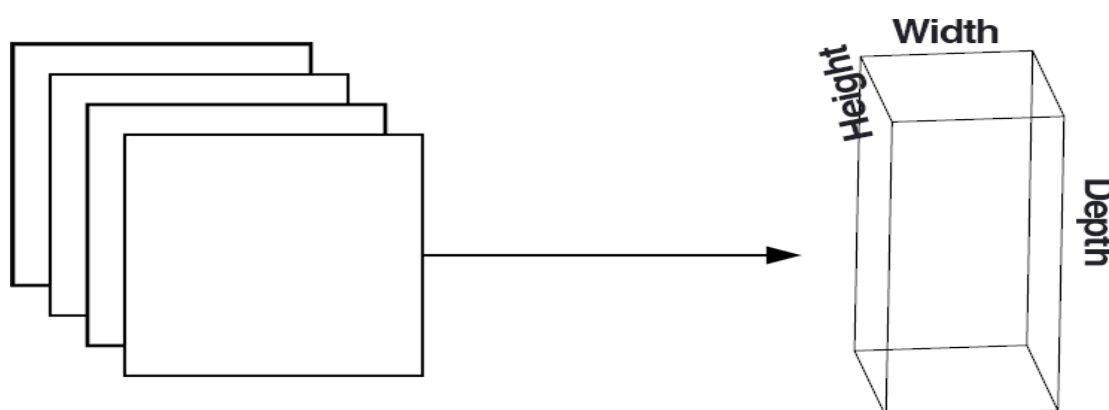


**Figure 5.17 2D act map**

4.17 In CNN at every layer  $k$  Kernel applied (Left), Each Kernel  $k$  with input convolved (Middle) and 2D result of every  $k$  Kernel (Right)

Once filter with ' $k$ ' is applied on the input 2D act map with ' $k$ ' layers is obtained. For the final result volume, the  $k$  act map is tacked with array of dimension as displayed in the figure 4.18. The result that has entry is the output of the volume of the input. The system would learn filtering in the input. When filters look at the edge or side region the layers that are low level would be activated to add filters for network.

Where there are features of high level the layers that are deeper would get activated for objects like cat ear, dog paw etc. This activation thought is from the theory of neural network. When there is a particular object in an image the layers would become activated for training them.



**Figure 5.18 Result of  $k$  Kernel stacked to produce input for next layer**

The concept of convolving a small filter with a large(r) input volume has special meaning in Convolutional Neural Networks – specifically, the local connectivity and the receptive field of a neuron. When working with images, it's often not practical for connecting neurons in the present volume to all earlier neurons volume – there are simply too many connections and too many weights, making it impossible to train deep networks on images with large spatial dimensions.

Instead, when utilizing CNNs, we choose to connect each neuron to only a local region of the input volume – we call the size of this local region the receptive field (or simply, the variable  $F$ ) of the neuron.

To make this point clear, let's return to our CIFAR-10 dataset where the input volume as an input size of  $32\_32\_3$ . Each image thus has a width of 32 pixels, a height of 32 pixels, and a depth of 3 (one for each RGB channel). If our accessible area is of  $3\_3$ , then every neuron would connect to a  $3\_3$  local region of the image in the conv layer to obtain a total of  $3\_3\_3 = 27$  weights (remember, the depth of the filters is three because they extend through the full depth of the input image, in this case, three channels).

Now, let's assume that the spatial dimensions of our input volume have been reduced to a smaller size, but our depth is now larger, due to utilizing more filters deeper in the network, such that the volume size is now  $16\_16\_94$ . Again, if we assume a receptive field of size  $3\_3$ , then each neuron will be having a total of  $3\_3\_94=846$  connections in the conv layer to the input.

Simply put, the receptive field  $F$  is the size of the filter, yielding an  $F\_F$  kernel along with the input it is convolved.

At this point we have explained the connectivity regarding the input based on the network system, but not the arrangement or size of the output. For controlling the size of the output three constraints are used: depth, tread, and padding (zero).

#### ➤ **Depth**

The network system which would get connected to the local area of input is controlled at the result volume in conv layer. An Act map is generated by every neuron with the presence of concerned sides, color and spots.

The act map deepness would be of  $k$  size for the conv layer or the total neurons that are learnt in the present layer. The neurons that are at the position of  $x$  and  $y$  in the input is the column of deepness.

➤ **Zero-padding**

The borders of image have to be padded to recollect the size of the image that is the real one during the conv layer, this is repeated for the neurons present inside the CNN. With the help of padding, we “pad” our input with the margins so that the result volume must be matching with the size of the input. The quantity of pad applied for input is measured by the constraint  $C$ .

This particular method is difficult when compared with the deep CNN structure that apply multiple CONV filters on top of each other.

To visualize zero-padding, applied a  $3_3$  Laplacian kernel to a  $5_5$  input image with a stride of  $S = 1$ . The output volume is smaller ( $3_3$ ) than the input volume ( $5_5$ ) due to the nature of the convolution operation. If we instead set  $P = 1$ , we can pad our input volume with zeros to create a  $7_7$  volume and then apply the convolution operation, leading to a size of the result volume which would get matched with the size of the input which is of  $5_5$ . The dimensions that are spatial in the volume of the input will be decreased very fast if padding is not performed and we wouldn't be able to train deep networks (as the input volumes would be too small for learning any beneficial patterns).

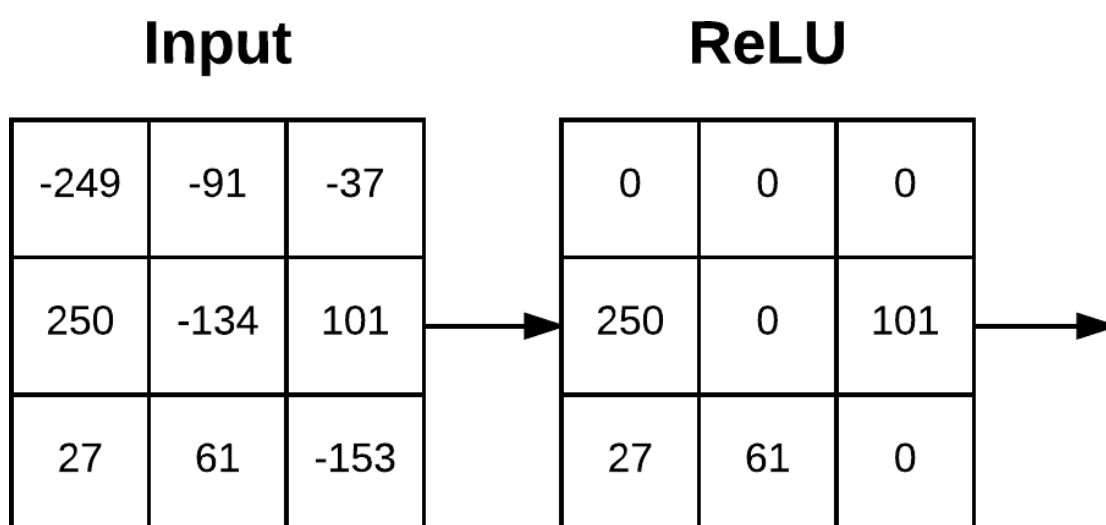
Putting all these parameters together, the size of the result volume can be calculated as a method or procedure of the size of the volume of the input ( $W$ , assuming the input images are square, which they nearly always are), the size of field that is accessible is  $F$ , the Tread is  $T$  and the padding (zero) is  $Z$  for constructing a Conv that is correct and valid.

➤ **Act Layers**

In CNN after every conv layer, we apply act procedure that is not linear like Relu, ELU, or any of the other Leaky ReLU variants. We typically denote activation layers as RELU in network diagrams as since ReLU activations are most commonly used, we may also simply state ACT – in either case, we are making it clear that an activation function is being applied inside the network architecture.

Activation layers are not technically “layers” (due to the fact that no constraints and weights inside an act layer is learnt) and they are left out from the diagrams related to the network sometimes as it’s assumed that an activation immediately follows a convolution. In this case, authors of publications will mention which activation function they are using after each CONV layer somewhere in their paper. As an example, consider the following network architecture: INPUT => CONV => RELU => FC.

An activation layer accepts an input volume of size  $W_{input} * H_{input} * D_{input}$  and then applies the given activation function as displayed in below figure 4.19.



**Figure 5.19 Inputs passing to ReLU**

Since the activation function is applied in an element-wise manner, the output of an activation layer is always the similar as the dimensions of the input,

$$W_{in} = W_{out} \quad H_{in} = H_{out} \quad D_{in} = D_{out} \quad \textbf{Pool Layers}$$

There are two methods for reducing the volume of the input and its size - CONV layers with a tread that is greater than 1 and pool layers. For inserting pool layer between its successive is common.

#### ➤ **CONV layers:**

Image => Conv => Act => Pool => Conv=> Act => Pool => Fully-connected

The reduction of the dimension's length and height gradually of the volume of input is the main aim of the pool layer. By performing this reduction, it would help us in reducing the constraints and calculation in the system, pooling would also help in controlling overfitting.

With the usage of the maximum or regular procedure the deepness of the slice can be operated by the pool layers. In between of CNN the maximum pooling is performed for reducing the size of the dimensions i.e. length and height and regular pooling is performed as the last layer in the system.

The most common type of POOL layer is max pooling, although this trend is changing with the introduction of more exotic micro-architectures.

Typically we'll use a pool size of 2\_2, although deeper CNNs that use larger input images (> 200 pixels) may use a 3\_3 pool size early in the network architecture. We also commonly set the stride to either  $S = 1$  or  $S = 2$  as displayed in below figure 4.20.

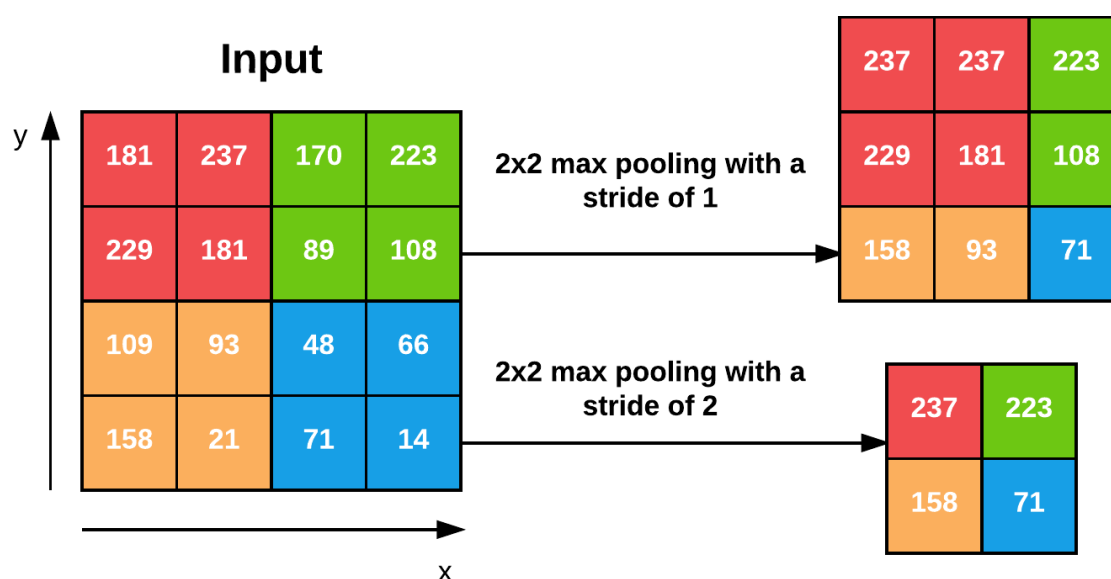


Figure 5.20 Result of 4\*4 input (Left) Max pooling on 2\*2 with step size 1 (Right)

For every 2\_2 block, we keep only the largest value, take a single step (like a sliding window), and apply the operation again – thus producing an output volume size of 3\_3.

We can further decrease the size of our output volume by increasing the stride – here we apply  $S = 2$  to the same input. For every  $2 \times 2$  block in the input, we keep only the largest value, then take a step of two pixels, and apply the operation again.

In brief, for accepting the size of the volume of input pool layer is used  $W_{in} * H_{in} * D_{in}$ .

They then require two parameters:

- The receptive field size  $F$  (also called the “pool size”).
- The stride  $S$ .

Applying the POOL operation yields an output volume of size multiple of all  $W_{out}$ ,  $H_{out}$  and also  $D_{out}$ , where:

- $W_{out} = ((W_{input} - F + 2Z)/S) + 1$
- $H_{out} = ((H_{input} - F + 2Z)/S) + 1$
- $D_{out} = D_{input}$

### ➤ Fully-connected Layers

In Fc layer the networks are connected fully compared with the previous layers for all activations and this is the main layer for feed-forwarding the NN. The Conv layer, Fc and other Conv layer are not applied whereas Fc layer would be applied always during the finish of the system. One or two full connected process can be applied for the classifier of the softmax function and it is common also for demonstrating the networks:

Image  $\Rightarrow$  Conv  $\Rightarrow$  Act  $\Rightarrow$  Pool  $\Rightarrow$  Conv  $\Rightarrow$  Act  $\Rightarrow$  Pool  $\Rightarrow$  Full-connected  $\Rightarrow$  FC

Here we apply two fully-connected layers before our (implied) classifier for the softmax function is computed for the final result of every class.

### ➤ Batch Normalization

Batch normalization layers (or BN for short), this is used for the normalization of the volume of the input given for activating before it is forwarded into the next layer of the system. At testing time, we replace the mini-batch  $mb$  and  $sb$  by the average of the  $mb$  and  $sb$  running that is calculated at the time of training.

By this it is known that the image can be passed through the system and the predictions that are obtained can be correct even if the mb and sb is not influenced at the time of training the system. BN is effective to an extent for the reduction of epochs that is taken for the training of the NN.

➤ **Dropout**

The final layer type that is used is dropout. DO layer would help in preventing the overfitting. This is done by increasing the correctness in the testing and expense in the correctness of training. The inputs are casually disconnected with the probability of 'p' in the DO layer for forwarding it to the next layer in the system for every batch of dataset that is under training. It is most common to place dropout layers with  $p = 0.5$  in-between FC layers of an architecture where the final FC layer is assumed to be our softmax classifier.

**Advantages**

- Considering to other modules working distance is more.
- Accommodation of many hardware device modules is not required.

**Expected Outcome**

- Live object recognition system using convolutional neural networks is developed.
- This helps the life of visually impaired people easier.
- The system that is implemented would demonstrate a network such that it can be used for identifying the obstacles and helping the purblind people.
- This helps visually impaired people less dependent on others.



## Chapter-6

# SYSTEM IMPLEMENTATION

### 6.1 Modules

- Collecting Dataset
- Splitting Dataset
- Training Network
- Evaluate/Testing Network

### 6.2 Module Description

#### ➤ Collecting Dataset

The first step in designing a network in deep learning is to collect the dataset initially. The images with the label associated with the images are required for designing the system. Categories that are with finite number sets produce the labels for the images (ex: cat, wood, flowers etc).

Furthermore, the images selected for each category should be having balanced number of images (i.e dog 1000 images then cat also 1000 images). If the images selected are not equal means, if the flower images are selected as two times greater than the cat image then the image of table is selected thrice more than flower image this would make the system unbiased. This is a common problem in machine learning when we are designing the system to make it work as a human brain. The class that is not balanced problem can be overcome using many techniques but the easy way to overwhelmed and is used as a balanced data or class while designing the network.

### Splitting Dataset

The dataset would be splitted into two methods:

- Training set.
- Testing set.

The set of images in the system has to undergo training Once after the training is done on the dataset the system has to undergo testing on certain dataset. To perform this process the dataset has to be kept separate which is different from the trained dataset. During development stage, having vast amount of data is not very easy. For this kind of situation, the solution is splitting the set of data into 2 sets, out of two – one is used for the train process and the other is used for the test process this must be performed before the network training is getting started. The splitting of dataset is done in many ways like 25% for testing and 75% for training, 30% for testing and 70% for testing and finally 10% for testing and 90% for training it is displayed in below figure 5.1.

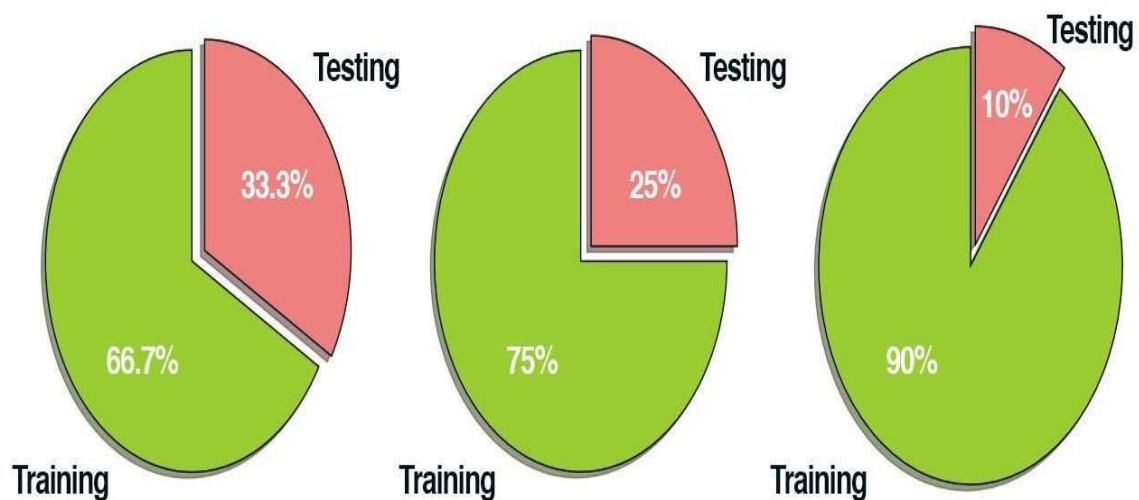


Figure 6.1 Dataset splitted to test and train as an example

For splitting the dataset into train and test process randomly scikit library can be used. For splitting the dataset in different proportions a library can be used. Once the testing and training is completed on the dataset one of the method is done which is overfitting or underfitting. By performing overfit the system will be trained very well and will be having complex model. On the trained data this method would be more accurate and on the data that is not trained this method is not accurate. If the model is underfit it is not suitable for the data trained.

This method would be the output of a simple system. The ability of the system for predicting objects would also be poor. The difference is as shown below figure 5.2

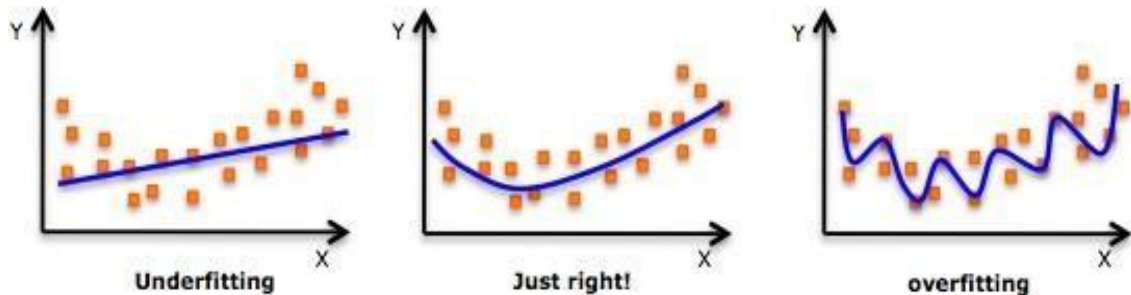


Figure 6.2 Example for right, overfit and underfit

### ➤ Training Network

The image set is taken for the process of training where the system can be started to train. The aim here is if the system must be known recognizing the objects the in the image based on the label that is on data the system must be trained. If the system performs any mistake, the system would learn from that mistake and get improve in recognizing image. So, how does the actual “learning” work? In general, we apply a form of gradient descent.

#### Steps to perform CNN algorithm:

1. To save the figures in background, use matplotlib as a backend.
2. Import necessary packages
3. To parse the arguments, create argument parse.
4. Initializing number of epochs for training, batch size, learning rate and dimensions of image.
5. Initializing labels and data
6. Take the path of image and mix them randomly.
7. Looping input images.
8. Loading image, preprocessing, and storing in the list of data.
9. From the path of the image collect the class label and update the list of labels.
10. Intensities of the pixel of the raw data should be reduced to range 0 and 1.

11. Binarize the labels.
12. Separate the data into testing and training splits using 20% for testing and 80% of the data for training.
13. Construct the image generator for data augmentation.
14. Initialize model.
15. Training network.
16. Saving model to disk.
17. Save the label binarizer to disk.
18. Plotting training loss/error and accuracy.

➤ **Evaluate/ Testing Network**

Finally, the system that has trained must be evaluated or tested. The each and every images that have been trained in the previous step would be collected and given to the system for testing and it is asked to find out the label of the image. Based on the answer given by the system the model accuracy is tested and predicted and that would be tabulated. Later, the predictions performed by the network is checked with the label of the image that has been given for the testing from the previous step. The ground-truth labels represent what the image category actually is. From there, we can compute the number of predictions our classifier got correct and compute aggregate reports such as precision, recall, and f- measure, which are used to quantify the performance of our network as a whole.

## 6.3 Functions

➤ **Numpy**

The full form of NumPy is Numerical Python, for the many dimensional array and collecting arrays for processing is done using this library. Some of the operations related to mathematics and logics are performed using this library. There are certain procedures for constructing array and indexes also depends. This library would come under the package with python and also called as number python.

Packages like scientific python – scipy and matplotlib library allow this library to use their packages. This library would replace the matlab which is most powerful in the field of computing the tech.

### ➤ **OpenCV**

The library that is a cross platform for which it is used as developing system vision apps in the real time environment. The main aim of this particular library is focusing on the processing of the image, capturing the video and analyzing the featured that are included in detecting face and detecting the object.

System vision is called as a field that explain the constructing again and interrupting or understanding the 3 dimen image from the 2 dimen image based on the structure and way it is presented in the view. With the help of system software and devices that human view is duplicated and modeled by this library.

System Vision overlays expressively with the following areas –

- **Processing of image** – Manipulating the image is mainly focused.
- **Recognizing pattern** – For classifying pattern various methods are explained.
- **Photogrammetry** – The measurements related to the image is obtained accurately. The transformation of one image to other image is the process called as processing of image. For processing images input and output both would be images only.

System vision is the creation of clear, expressive metaphors of objects that are present physically from dataset. The result of system vision is a explanation or an clarification of constructions in 3 dimen view.

### **Steps to set matplotlib:**

1. First step is to import the matplotlib: `import matplotlib`
2. For plotting backend agg is used. `matplotlib.use('agg')`

### Importing packages

All the necessary packages are imported:

1. Keras for preprocessing images.
2. Image search package for smaller images.
3. Numpy, random, cv2, os and argparse libraries are also imported.

### Constructing 'argparse' for argument parsing.

1. Assigning the argument parser function.
2. Path of the dataset input is assigned that is in directory.
3. The path for the output of model is defined.
4. The accuracy output path is also defined

### Initializing rate of forward and backward parsing for dataset, size of batch and dimensions of image

Set the values for all a parameters epochs = 100

bs = 32

img\_dim = (96,96,3)

### Data and labels defining

Initialize an array for data and labels.

### Path of images collected to shuffle randomly

1. Path of images is collected and sorted

```
img_path = sorted (list(paths.list_images(args['dataset'])))
```

2. Path of images are shuffled random.shuffle( img\_path)

**For storing image in data list – load and preprocess**

```
img = cv2.imread(img_path) img = img_to_array(img)
```

**Updating list of label from class label**

```
Split and append the label label = imagePath.split() labels.append(label)
```

**Setting range for the intensity of raw pixel**

```
numpy (object, dtype='datatype')/255.0 labels= numpy.array(labels)
```

**Separating data for training and testing 80% (training) and 20% (testing)**

```
train_x, test_x, train_y, test_y = train_test_split(data, labels, test_size=0.2)
```

## CHAPTER-7

### SYSTEM TESTING

The goal of the testing is to find out the defects and faults by performing testing on each element individually. These elements can be called as functional, modules or units. When performing testing for system all these elements are deployed together as a result of full one system.

To ensure that system behavior can be examined in all possible combinations of conditions, the test cases are selected. System testing includes bringing together all the modules and reviewing all of the applications. Its useful to check whether or not we get desired output for the given input as a result. Unit research should be primarily practical in nature. This would focus on the correct and incorrect cases, values based on boundary and the special term.

#### 7.1 Testing

Levels of software testing are the different stages of the SDLC (software development life cycle) where research is carried out. There are 4 levels of testing the software as shown in below figure 6.1.

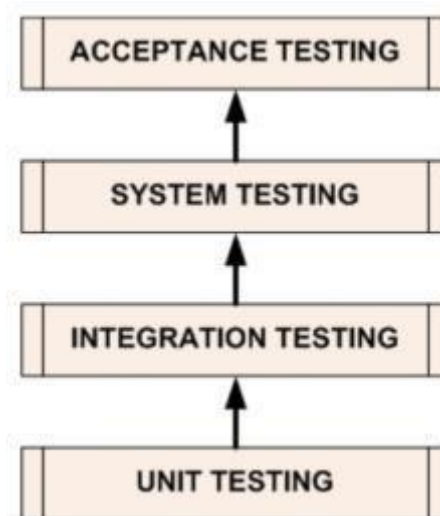


Figure 7.1 Testing levels



Tester describes three separate datasets viz for evaluating a ML (machine learning) technique. Training the dataset, testing the dataset and evaluating the dataset (a research dataset subset). Tester would define 3 datasets, set of training data (65 percent) set of validation of data (20 percent) and set of test data (15 percent). Until breaking, please randomize the dataset and do not use the validation or test data set in your training dataset. Once the dataset is specified by the tester, he will start training the models with the training dataset. When this training model is completed, the tester then conducts with the validation dataset to validate the models. This is iterative and will accept any tweaks or changes that are required for a model based on results that can be achieved and reassessed. This means the test dataset stays unused and can be used to check a model that has been tested.

When the system is evaluated the tam would predict the best model that they get confident on the system which would produce less error and good predictions will be chosen and given for testing with the dataset that is available to make sure that the system is working perfectly and it would match with all the results of the dataset that has been validated before. The validation and testing of dataset is not leaked into the training this must be ensured if the system accuracy is good.

## **7.2 Manual and Automation Testing**

The testing with relation to the ML is used to perform testing based on the correctness of the system. Based on the software the testing is different and based on ML it is different but the aim of both testing is same to make sure the system works accurately.

As the programs are tested in the tech field even the ML systems must be tested based on the quality and correctness view. Some of the testing procedure like blackbox and whitebox are applied on ML to perform check on the quality of the network. Testing that is performed manually is used to check the modules of project without having any data or information about the project that has implemented even the program in the network or system. In manual testing based on the modules requirements the test cases are prepared and called as requirement based testing.

In the field of ML testing means as same as in the program testing where the tester would not be having knowledge about the program same as the tester will not be having any knowledge about the internal structure of the machine designed like the technique used.

This would be a challenge for the tester as he should try and test the result with the actual result. The input and output of the implemented system would be specified by the developer team the tester should check with the result section as it is matching as specified by the programming team. If the result matches the system is correctly programmed and does not have any defects.

Functional Test Automation should be done only once the feature / product is stable. Once the team knows what now needed to automate at the top / UI layer, we should automate those tests. To make the once-passing test pass again, the Functional Test Automation Tool Framework should make updating / evolving the existing test as easy as possible. The changes may be required in the locators, or in the flow - it does not matter. If this process is easy, then team members will get huge value from the tool / framework and the tests automated / executed via the same. The most important aspect to me of Test Automation - understanding what has been automated - and does it indicate value of the test as opposed to a series of UI actions.

### **7.3 Unit Testing**

Although TF2.0 and Keras provide a set of well-implemented algorithms. In many cases, we still need to make our hands dirty and implement our own models and layers. Implementations involve a set of mathematical transformations. This makes correctness testing necessary. Unit testing is suitable in this when we want to test the correctness of our own implementation. TF 2.0's unit test is very similar to the python unit test. Everything starts with a TestCase class.

We implement one test method called testDenseLayerOutput. This test method checks whether the dense layer gives us the correct output. The assertAllEqual is provided in tf.test.TestCase, this method checks whether the expected output equal to the computed output.

In OOP, procedure is the smallest unit and it will belong to the class that is base or super or abstract or child. The module of the app will also be treated as a functional from some of the people and this will discourage may people individually inside those units. The framework for testing unit by unit dummy objects are used for assisting the unit during the time of testing.

## 7.4 Integration Testing

Integration testing comes under the program testing in which separate units are formed and deployed together to perform testing. The errors would be exposed in between the functions that are integrated this is the aim of this testing. For assisting drivers and stubs are used in this particular test.

A technique that is performed in integrated testing is bottom up test in which the elements that are present at the below would be tested initially and the elements above level would be tested. This would continue until all the elements from bottom to top are tested. The procedures and modules are deployed and tested. This approach helps in making the report of the testing conducted in easy way.

The other approach in integrated testing is the top down testing in this all the units that are present at the top level would be tested initially then followed by the units that are the bottom level. Testing is performed until all the units are tested step by step from top to the bottom of the system.

This particular testing is done after the functional testing and before the testing based on validation. The units that are tested would be taken as an input by the integration testing and all of them are grouped to apply testing and the result would be delivered as the system ready for working.

## 7.5 Acceptance Testing

In any field of technically related this particular acceptance testing is performed to test whether all the functionalities are fulfilled. Some tests are performed chemically, physically and based on performance too. Based on the requirements of the customer the testing is performed formally and the processes and applied to check whether the network satisfy all the requirements specified by the customer and they would specify if the system is acceptable or not.

- UAT is performed on the system to determine if the network is working in a correct way or not. The requirements that are used initially is the main functionalities that are generally used by the users so called as testing based on end user.

- For determining if the product can meet purpose and aim of the business or not BAT is used. It would focus mainly on the profits as the business today is changing from profit to loss and vice versa because of emerging of new technologies. The changes in the network may require extra cost.
- If the system goes for use it has to be performed with the testing based on acceptance to make sure that all the cases are pass this is CAT. The contract that is specified here is SLA the condition include is that the payment is done when the system is in the service in live with all the functionalities getting passed.
- If in case the product breaks the rule and protocols that are specified by the government which is getting released to the country this particular RAT is used. This would be not done intentionally but it would give an immense impact on business in a negative way.
- For determining whether the system can be read properly and understood is done by OAT. This would involve with the recovering, compatible, maintaining and reliable testing and so on.
- For determining the development of the system based on the surroundings tested by an experienced testing team is Alpha testing.
- Assessing the system to customers by exposing it to them is done by the beta testing based on the environment and the feedback would be taken from the customers to fix if any bugs are detected or any changes.

## 7.6 Test Cases

A collection of statements that has the procedure and details about the functionalities underwent for testing is called test case. By writing or developing test case it would help in finding out the faults that the system has and no need of remembering the errors as everything would be noted in the test case.

To make sure that the system is ready for release and can be used by customers for every functional requirements atleast there must be around 2 to 3 test cases based on valid and invalid input. The requirement containing sub fields must also be tested with positive and negative values to make sure that the system is suitable for real time.

For some of the simple applications no need of mentioning with the test cases all the time. The test case that has created must be understandable by the user and the tester and also to the development team. The developers would refer with these test cases for repairing their bugs so it must be as much as easy and understandable. The requirements that are mentioned are tested based on the test plan and scenario and the test cases are noted with the result status and defect with their severity.

Below are the test cases of the system:

**Video capture:** The images are captured from the webcam and the video has to be displayed from the webcam as specified by the user. The input would be video from the webcam with good resolution. Output expected is to display video from webcam, resolution as specified by the user. This test case has executed and successfully passed as per the expected output as shown in below table 7.1

Test Case	1
Name of Test	Video capture
Input	Webcam, Resolution(width, height )
Expected output	Display video specified by user resolution from webcam
Actual output	User specified video is displayed
Result	Successful

**Table 7.1 Test Case for Video capture**

**Loading trained model:** The models that are trained are loaded to check if any error exists in the model. The input would be video yolo trained weight, cfg and the names of the model. Output expected is to model getting loaded without displaying any error. This test case has been executed and successfully passed as per the expected output as shown in below table

Test Case	2
Name of Test	Load trained model
Input	Yolo trained weight, cfg and names
Expected output	Model loading without any error's
Actual output	Model Loaded without any error's
Result	Successful

**Table 7.2 Test case for load trained model**

**Classify objects:** The images are captured from the webcam are resized or preprocessed for classifying the objects from the image. The input would be image that is resized from the webcam. Output expected is to classify the objects from the input image specified by the user. This test case has been executed and successfully passed as per the expected output as shown in below table

Test Case	3
Name of Test	Classify Object's from input image
Input	Resized input from webcam
Expected output	To classify Object's from user input
Actual output	Objects' classified from user input
Result	Successful

**Table 7.3 Test case for classifying objects**

**Localize object location:** The images are captured from the webcam and the image that are resized or preprocessed will be used to localize the objects from the image input that was classified. The input would be image resized from the webcam. Output expected is to localize objects from the input image that is classified. This test case has been executed and successfully passed as per the expected output as shown in below table

Test Case	4
Name of Test	<b>Localize Object's location</b>
Input	Resized image from webcam
Expected output	To localize object's from classified input image
Actual output	Classified object's from input image is localized
Result	Successful

**Table 7.4 Test case for localize object location**

**Display detected objects:** The images are captured from the webcam and the object that is detected has to get plotted with the bounding box. The input would be image from the webcam with good resolution. Output expected is to display plot the bounding box around the object detected. This test case has been executed and successfully passed as per the expected output as shown in below table

Test Case	5
Name of Test	<b>Display detected Object's</b>
Input	Input image from webcam
Expected output	Plot bounding box for all the detected objects
Actual output	Bounding box for all the objects detected are plotted
Result	Successful

**Table 7.5 Test case for displaying detected object**

**Audio message:** The images are captured from the webcam would be localized and that particular image with the detected objects has marked with the bounding box and generates the output with the audio message as name of the object detected. The input would be localized and bounded image with detected object. Output expected is in the form of the audio message. The audio would be produced with the name of the detected object as a result. This test case has been executed and successfully passed as per the expected output as shown in below table

Test Case	6
Name of Test	<b>Converted detected object's to audio</b>
Input	Localized image with bounding box
Expected output	Audio output for all the detected objects
Actual output	Audio generated for all the detected objects
Result	Successful

**Table 7.6 Test case for converted detected object to audio**



## **CHAPTER-8**

### **RESULT AND DISCUSSION**

Result represents that program for recognition of object is implemented successfully using CNN (Convolutional Neural Network). The aim is to help purblind people in making their life better by detecting and assisting them with the obstacle or object detected. Proposed model tells us that this program can be used for distinguishing between artifacts and supporting impaired people.

## CONCLUSION

A system based assisting network has been proposed in order to assist the purblind people and completely blind people. The template that are matching the procedures conducted by experimenting using OpenCV has formed a successful method that is multiscale and useful for the applications used inside the surroundings. The constraints that are based on time and the range of detection are the optimum numbers which need to be founded depending on the values of the factors based on the scaling and the length and width of the image.

The objects detected are finally output as an acoustic message with the name of the object detected. The accuracy will be depended upon the clarity of the image captured by the user. If the image looks similar to other objects there may exist an ambiguity which would reduce the accuracy of the object detected. Model is trained to detect 78 objects with a maximum of accuracy. The distance of the image getting captured depends on the camera. The vision of the system for the accuracy it can be made better by improving the constraints that are adapted for illuminating and changing for real life surroundings.

## REFERENCES

- [1] K Sarthak, Sanjay K, Ronak S, Samarth G, "Object Detection in Foggy Conditions by Fusion of Saliency Map and YOLO," 12th International Conference on Sensing Technology (ICST), IEEE 2018 Dec 4.
- [2] Arakeri MP, Keerthana NS, Madhura M, Sankar A, Munnavar T, "Assistive Technology for the Visually Impaired Using Computer Vision," in 2018 Sep19, International Conference on Advances in Computing, Communication and Informatics (ICACCI), IEEE.
- [3] Kun S, Hayat S, Tengtao Z, Tu T, Y Du, Yu Y, "A Deep Learning Framework Using Convolutional Neural Network for Multi-class Object Recognition," International Conference on Image, Vision and Computing (ICIVC) in 2018 3rd Jun 27 (pp. 194-198) IEEE.
- [4] Aishwarya S, Kaiwant S, Anandji C, Tatwadarshi N, "An Innovative Machine Learning Approach for Object Detection and Recognition," in 2018 2nd International Conference on Inventive Communication and Computational Technologies (ICICCT), Apr 20, IEEE.
- [5] Jeong H J, Park K S, Ha Y G, "Image Preprocessing for Efficient Training of YOLO Deep Learning Networks," IEEE International Conference on Big Data and Smart Computing, (pp 635-637), 2018 Jan 15.
- [6] Yawei H, Huailin Z, "Handwritten Digit Recognition Based on Depth Neural Network," IEEE, ICIIBMS Track2 2017 Nov 24.
- [7] Malay S, Rupal K, "Object Detection Using Deep Neural Networks," International Conference on Intelligent Computing and Control System, 2017 ICICCS IEEE. June15.
- [8] Xinyi Z, Wei G, Wenlong F, Fengtong D, "Application of Deep Learning in Object Detection," IEEE, 16th International Conference on Computer and Information Science, ACIS May 24 2017.
- [9] Ani R, E Maria, J.J, V Sakkaravarthy and M A Raja, "Smart Specs: Voice assisted text reading system for visually impaired persons using TTS method", IGEHT, IEEE, Mar 16 2017, pp. 1-6.
- [10] Tianmet G, Jiwen D, Henjian L, Yunxing G, "Simple Convolutional Neural Network on Image Classification," IEEE in 2017 2nd International Conference on Big Data Analysis (ICBDA), (pp. 721-724) Mar 10.
- [11] Hassan E A, Tang TB, "Smart glasses for the visually impaired people", Jul 13 2016, in International Conference on Computers Helping People with Special Needs, pp. 579-

- [12] Khade, Sanket and H D Yogesh, "Hardware Implementation of Obstacle Detection for Assisting Visually Impaired People in an Unfamiliar Environment by Using Raspberry Pi", 2016 Aug, International Conference on Smart Trends for Information Technology and Computer Communications, pp. 889-895, Singapore.
- [13] Michael T, Ahmed A and Kumar Y, "A Smart Wearable Navigation System for Visually Impaired", International Conference on Computers Helping People with Special Needs, Nov 30 2016, pp. 333-341, Springer, Cham.
- [14] Liu Shuying, Deng Weihong, "Very Deep Convolutional Neural Network Based Image Classification Using Small Training Sample Size," IEEE in 2015 Asian Conference on Pattern Recognition (ACPR), 3rd IAPR 3 Nov.
- [15] S Christian, Wei L, Yangqing J, Pierre S, Scott R, Dragomir A, Dumitru E, Vincent V, Andrew R, "Going Deeper with Convolutions," 2015 June Conference on Computer Vision and Pattern Recognition (CVPR) IEEE.
- [16] Schmiduber J "Deep Learning in Neural Networks: An Overview," Neural Networks Jan 1 2015, 61:85-117.
- [17] Rashidan MA, Mustafah YM, Abidin ZZ, Zainuddin NA, Aziz NN, "Analysis of Artificial Neural Network and Viola – Jones Algorithm based Moving Object Detection," in 2014 Sep 23, International Conference on Computer and Communication Engineering pp. 251-254 IEEE.
- [18] Tepelea L, Tiponur V, Szolgay P, Gacsadi A "Multicore Portable System for Assisting Visually Impaired People," in 14th 2014 International Workshop on Cellular Nanoscale Networks and their Applications (CNNA) pp. 1-2 Jul 29 IEEE.
- [19] CJ Sung, DK Lim and YN Shin, "Design and Implementation of Voice Based Navigation for Visually Impaired Persons," in 2013 International Journal of Bio- Science and Bio- Technology, 5.3 June, 61-68.
- [20] L Tepelea, G Alexandru, G Ioan and Virgil T, "A CNN based correlation algorithm to assist visually impaired persons", in ISSCS, IEEE, Jun 30 2011, pp 1-4.
- [21] Virgil T, Daniel Lanchis, Zoltan Harazy, "Assisted Movement of Visually Impaired in Outdoor Environments – Word Directions and New Results," WSEAS 13th International Conference on SYSTEMS 2009.