A

PROJECT REPORT

On

"Object Recognition with Voice for Blind People"

BACHELORS OF ENGINEERING

BY

Bhor Ajay Sandip

Mhaske Priyanka dattatray

Jadhav Devyani Ranjndra

Kantode Pratik Dhananjay

Under the Guidance of

Prof. Yogeshwari Hardas

Atma Malik Institute of Technology and Research.

Affiliated to

DEPARTMENT OF COMPUTER ENGINEERING

UNIVERSITY OF MUMBAI

Department of Computer Engineering

A.Y:2023-2024

Vishwatmak Jangali Maharaj ashram Trust's

ATMA MALIK INSTITUTE OF TECHNOLOGY AND RESEARCH

DEPARTMENT OF COMPUTER ENGINEERING

This dissertation report entitled **"Object Recognition with Voice for Blind People"** by **Ajay S. Bhor, Priyanka D. Mhaske, Devyani R. Jadhav, Pratik D. Kantode** is approved for the degree of "**Bachelors of Computer Engineering"** academic year 2023 - 2024.

Examiners

1. _

2. _

Supervisor

1. _

Prof.

_____	_____
Head of the Department	**Principal**

Date:

Place:

DECLARATION

I declare that this written submission represents my ideas in my own words and where, others ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Date:

Place:

ACKNOWLEDGEMENT

I would like to take the opportunity to express my heartfelt gratitude to the people whose help and co-ordination has made this project a success. I thank Prof. Yogeshwari Hardas for knowledge, guidance and co-operation in the process of making this project.

I owe project success to my guide and convey my thanks to him. I would like to express my Heartfelt to all the teachers and staff members of Computer Engineering Department of AMRIT for their full support. I would like to thank my principal for conductive environment in the institution.

I am grateful to the library staff of AMRIT for the numerous books, magazines made available for handy reference and use of internet facility. Lastly, I am also indebted to all those who have indirectly contributed in making this project successful.

TABLE OF CONTENT		
CH. NO	**CONTENTS**	**PAGE NO**
	List of Figures	I
	List of Abbreviations	II
	Abstract	III
1	**Introduction**	1
	1.1 Introduction	2
	1.2 Background	2
	1.3 Problem Statement	3
	1.4 Goal and Sub-Objectives	4
	1.5 Final Solution	5
	1.6 Final Approach	7
	1.7 Project Scope	8
2	**Literature Survey**	10
3	**Problem Definition**	15
	3.1 Current Limitations	16
	3.2 Desired Outcomes	16
	3.3 Challenges to Address	16
4	**System Specification**	18
	4.1 Hardware Requirement	19
	4.2 Software Requirement	19
5	**Proposed System**	20
	5.1 Methodology	21
	5.2 Open cv	23
	5.3 Dataset and Packages	23
	5.4 Algorithm	25
6	**System Design**	34
	6.1 Use Case	35
	6.2 Sequence Diagram	36
7	**Implementation And Result**	37
	7.1 Text to speech	38
	7.2 Performance Analysis	39
	7.3 Properties	41
	7.4 Testing	42
	7.5 System Result	43
8	**Conclusion**	46
9	**Reference**	48
	Annexure I: Research Paper	
	Annexure II: Research Paper Certificate	

SR. NO	ABBRIVATION NAME
01	py: python
02	ML: Machine Learning
03	IDE: Integrated Development Environment.
04	CV: Computer Vision
05	YOLO: You Only Look Once
06	CNN: Convolutional Neural Network
07	DFD: Data Flow Diagram
08	ANN: Artificial Neural Network
09	RNN: recurrent neural network

ABSTRACT

Vision is one of the most important human senses and it plays a critical role in understanding the surrounding environment. However, millions of people in the world experience visual impairment. These people face difficulties in their daily navigation since they are unable to see the obstacles in their surroundings. Despite there are many options such as white canes and different advanced technologies to help visually impaired people when navigating, some of the options are unreliable, expensive, and hard to access. Hence, a mobile application is proposed to help visually impaired people to recognize objects in their surroundings using real-time object detection and object recognition techniques. This project also has applied transfer learning on multiple pre-trained models to train the models that are able to classify 40 classes of objects. The performance of the trained models is compared to select a suitable model to be implemented in the mobile application. The Evolutionary Prototyping Model is the development methodology adopted in this project. It involves developing the application in a series of iterations and refining the application based on feedback collected in each iteration. A literature review was conducted on similar existing mobile applications to understand the machine learning framework used for the implementation of object detection and recognition, and also identify the important features and workflow within the application. Finally, an Android-based mobile application was developed successfully and passed all testing. In conclusion, this project has helped visually impaired people to determine the objects in their surrounding in a more cost-effective, accessible and reliable way. They are being informed of the names and directions of the detected objects in the surroundings through voice feedback without requiring any network connection or photo capturing.

Keywords: computer, computers, feedback, object, tasks, innovation, accuracy, advances, vision, videos.

CHAPTER 1

INTRODUCTION

CHAPTER 1

1. INTRODUCTION

1.1 CONCEPTUAL STUDY OF THE PROJECT

Innovation was a substantial word 15 years back however the new age is all forward thinking now. We could see the effect of innovation on our generation and how it has helped us as humans achieve extraordinary things. Innovation obviously has helped a normal human achieve impeccable things, but what about people with defects or a disabled human. They too deserve to see things or feel the beauty this world has to offer. Innovation isn't just to help a normal human do incredible things but also help a disabled person perform better. According to who as of 2019 37.5% of the world's population is a disabled person. This means that every 3rd person you meet is somehow disabled, for example, blind, deaf, colour blind, dyslexia, etc. This is where the need for innovation for disabled people comes in. The significance of this is so high because out of 37.5% disabled humans, 10% of them suffer from some or the other type of blindness. That is 15% of the whole disabled population on earth. This means that 30 crore humans in the world suffer from blindness. This is the main and key driving factor of this project.

1.2 Background

Smartphones have become a very significant device in our lives. They allow us to access various services and information more easily. Nevertheless, there are millions of people unable to see the environment in this world due to visual impairment. Visually impairment hinders the people from carrying out a lot of daily activities. It is a challenge for them to travel independently since they are unable to see the obstacles around them. They always require someone to guide them when navigating around to prevent injuries and accidents. Furthermore, they also face difficulties to complete their daily tasks such as reading and finding an object. They may need help from others to complete these tasks. However, these already increase the burden of family members and friends of a visually impaired person.

With the common use of smartphones, visually impaired people are able obtain benefits from smartphone applications. Several types of mobile applications have been invented to help the visually impaired people, such as text readers that read out text on books and documents, color readers that notify the visually impaired user regarding the colour information of an object, navigation assistance that helps visually impaired users to navigate around by telling the route,

and other applications. These applications allow visually impaired people to do some simple tasks independently without having to seek help from others. The project is proposed to aid the visually impaired people using computer vision, object detection, and object recognition techniques by building a mobile application that detects and recognizes objects in the surroundings and gives audio feedback.

1.3 Problem Statement

Vision is critical in our daily lives. However, according to the World Health Organization (2019), there were more than 2.2 billion people worldwide suffering from visual impairment. These people are unable to view the surrounding objects unlike a person with normal vision. They face challenges in detecting obstacles when navigating (Rajwani et al., 2018; Thakare et al.2017). Although there are several options available for visually impaired people to help them when navigating, such as white canes and advanced technologies, they still encounter a few problems when accessing or using the tools.

1.3.1 Safety Issues When Navigating Using White Canes

In Malaysia, the white cane is accepted as a symbol of blindness (National Council For The Blind Malaysia, 2020). White canes are widely used by visually impaired people in detecting obstacles since they are cheap and easy to get (Khan, Khusro and Ullah, 2018; Santos et al., 2020; Chanana et al.2017). The canes are painted white to make others notice them easily, especially when navigating (Industries For The Blind And Visually Impaired,2020).

But, white canes cannot help them to determine the type of objects in front of them. Therefore, visually impaired people usually identify the object in front of them based on their own experience (Parikh, Shah and Safvan Vahora, 2018). Unfortunately, Santos et al. (2020) and Chanana et al. (2017) states that they may make incorrect expectations, which can cause them injury.

Besides, white canes are also unable to detect the obstacles above the waist level (Khan, Khusro and Ullah, 2018; Santos et al., 2020; Chanana et al.2017). According to Santos et al. (2020), 40% of visually impaired people experienced at least one head accidents every year. Santos et al. (2020) also reported that 23% of accidents had medical consequences. Therefore, white canes expose visually impaired people to the risks of colliding with the obstacles above the waist level, such as tree branches, windows, and floating shelves.

1.3.2 Accessibility and Affordability Issues of Advanced Technologies

Several types of technologies that involve special devices have been developed to help visually impaired people. One of the examples is smart glasses with a camera to capture the user's surroundings and send images to a smartphone for processing (Thakare et al., 2017). Smart stick with ultrasonic, infrared, or laser sensors also has been developed to inform user information of obstacles (Sharma et al., 2017). Moreover, Radio Frequency Identification (RFID) tag and RFID reader is used to assist visually impaired people. RFID tags are attached to the objects so that the user can identify and locate objects more easily (Abdul Malik Shaari and Nur Safwati, 2017). However, although these technologies are able to help the visually impaired users to do their tasks independently and safely, most of these technologies are expensive and hard to access (Anitha, Subalaxmi and Vijayalakshmi, 2019; Awad et al., 2018; Rajwani et al., 2018).

1.4 Goal and Sub-Objectives

This section discusses goal and sub-objectives.

1.4.1 Goal

The goal of this project is to implement an Android-based mobile application that detects and recognizes multiple objects captured by the smartphone's camera in real-time and provides audio feedback to assist visually impaired people identifying surrounding objects more easily. The object detection and recognition are achieved using transfer learning from existing pre-trained object recognition models. The system also incorporates new classes for detection and recognition.

Since only a smartphone is needed to identify surrounding objects without any other hardware, this can solve the accessibility issues of advanced technologies because smartphones are easier to be accessed than these special devices. Furthermore, smartphones are more affordable than these advanced technologies. The price of smart glasses ranges from USD2,950 to USD5,950, while the price of smart sticks ranges between RM579 and RM699, which are more expensive than a low-end smartphone (BAWA, 2020; Wewalk, 2020; IrisVision Global, 2021).

1.4.2 Sub-Objectives

i. To apply transfer learning to multiple pre-trained object detection models then train the models to detect and recognize 40 classes of objects. This also helps to fine-tune the pre-trained models and improve the models performance.

ii. To develop a real-time application that is able to analyse the surrounding scene and detect a maximum of 10 objects within the camera's field-of-view.

iii. To evaluate the training performance and recognition accuracy of different pre-trained

object detection models before and after transfer learning.

iv. To develop an application that uses voice feedback to notify the user of the names and directions of detected objects. This helps to reduce the safety issues encountered when navigating with the white cane.

1.5 Final Solution

With the rapid advancement of technology, smartphones have become a familiar and highly available device (Rajwani et al., 2018; Khan Shishir et al., 2019; Anitha, Subalaxmi and Vijayalakshmi, 2019). According to TheStar (2018), smartphone penetration in Malaysia stood at 70% in the third quarter of 2017. Hence, the final solution proposed an Android mobile application to assist visually impaired people to detect objects around them using real-time object detection and object recognition techniques. Since the smartphone is the only device needed, this solution is more cost-effective and easier to access rather than the technologies that need special devices. Furthermore, it is safer than white canes because the application is able to determine and inform the types of obstacles so that visually impaired people can avoid making wrong assumptions when navigating. The application also helps users to detect and identify obstacles above the waist level. An overall system architecture was designed to describe the solution, as illustrated below.

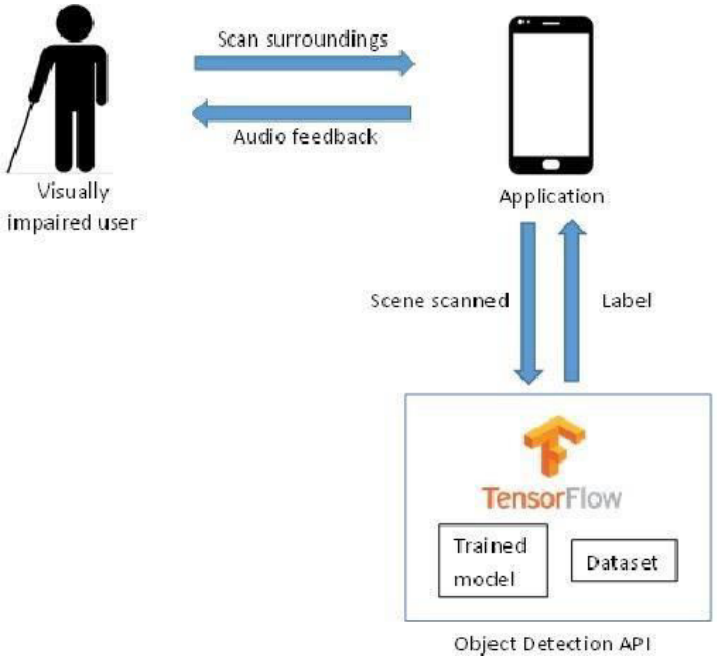

Figure 1.1: Proposed System Overview

In this system, a visually impaired user opens the mobile application and scans his or her surroundings using the smartphone camera. The scene captured through the continuous stream rather than taking a picture of a specific scene. Hence, the visually impaired user does not have difficulty in taking a good quality picture and does not need to capture a picture every time. Furthermore, the scene captured will not be stored in the smartphone memory so that the user does not have to delete the images from time to time.

The scene scanned is then sent to the Tensor flow Object Detection API (Application Programming Interface) for detecting and recognizing objects in the scene. Due to time limitations, an API is utilized in developing the application since it provides a list of operations so that a developer does not need to write code from scratch. The Tensor flow Object Detection API is an open source machine learning framework. It has been developed by Google Brain Team in 2015. It prepares a collection of pre-trained object detection and recognition models for a developer to deploy directly into the application and a developer also can choose to train his or her own model using this framework.

Other than Tensor flow Object Detection API, there are other types of object detection API available, such as Google Cloud Vision API, Microsoft Cognitive Toolkit, and Pytorch. Although Google Cloud Vision API provides more features such as colour recognition, landmark recognition, handwritten text recognition, and others, it is free for the first 1000 units per month only for each feature. Both Microsoft Cognitive Toolkit and Pytorch are open source. However, Microsoft Cognitive Toolkit does not support object detection models for mobile devices (Argawal, 2018). Pytorch provides more pretrained models than Tensorflow, but it has less community compared to Tensor flow so it will be harder to get the tutorials (Lobo, 2017). Moreover, Rane, Patil and Barse (2019) state that using Pytorch, the process is less efficient and is less reliable than Tensor flow. Hence, the main reason for choosing Tensor flow Object Detection API is that it is open source and has the largest community.

Instead of applying the existing pre-trained models into the application directly, transfer learning was applied to retrain the existing pre-trained models to recognize new classes and improve the accuracy of the recognition results of the object detection model. The final object detection model has been trained to detect and recognize the objects in environment context, especially indoor and outdoor obstacles. Its purpose is to make navigation of visually impaired people easier.

The object detection model within the application detects and recognizes an object based on its knowledge trained on the dataset. After that, the object detection model will return the label, coordinate, and confidence score of each object detected. As the visually impaired user is difficult

to see the name and the direction of the object displayed on the mobile screen, the name and direction are spoken out in audio feedback by using Android text-to- speech API. The audio feedback is provided to the user through smartphone speakers or earphones.

1.6 Final Approach

The methodology applied in the development of this application was the evolutionary prototyping model. The development of the complete system was done by undergoing a series of iterations until an acceptable prototype was built. The first phase of the model was requirements gathering. Next, design, prototyping, and user evaluation phases were performed repeatedly until the prototype was accepted by users. The design and development of the prototype were incrementally improved based on the users' feedback in every iteration. After the users was satisfied with the prototype, the final prototype was developed into the complete system. Figure 1.2 below illustrates the overview of the evolutionary prototyping model.

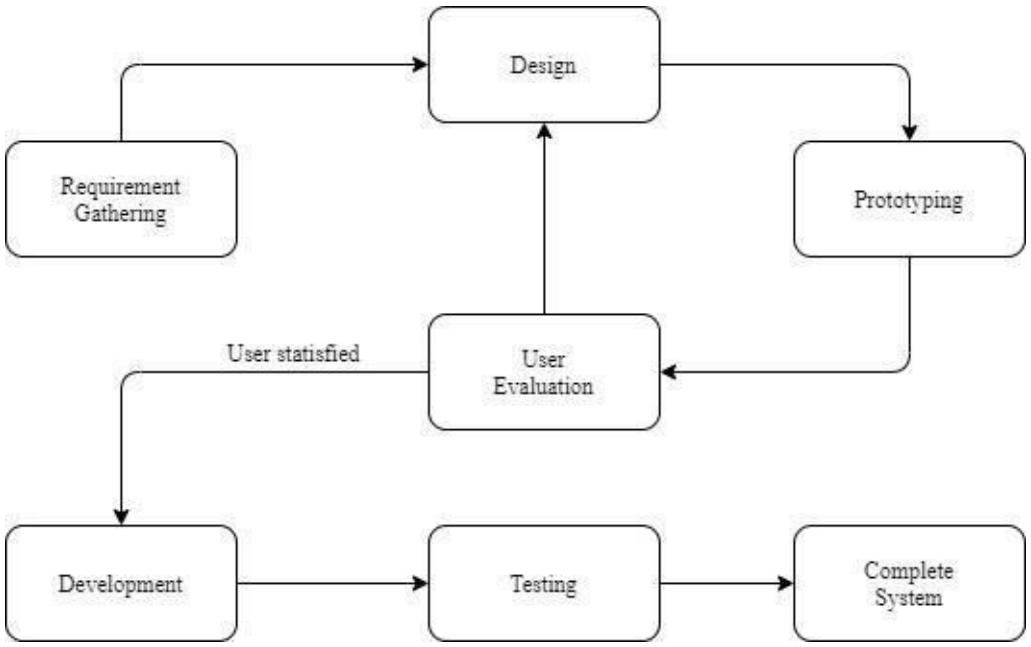

Figure 1.2: Evolutionary Prototyping Model

By applying this methodology, users are involved actively in the software development process. A developer can get feedback from users and find out the limitations, errors, and missing requirements in the prototype early. This is able to reduce the cost of rework since the cost of bugs tends to increase as the project proceeds. Furthermore, the complete system will be more user friendly and able to meet the users' requirements more accurately as the system is improved according to users' opinions.

1.7 Project Scope

The section discusses delimitations, limitations, and assumptions of the project.

1.7.1 Delimitations

This project is aimed to develop an Android-based mobile application that detects and recognizes objects in real-time.

Users can access this application by installing it on their Android smartphone. The main target users of this application are people who have a vision impairment or blindness. By using this application, visually impaired users can scan their surroundings using their smartphone's camera without taking a photo of the objects and saving the photo into the memory. This application is able to recognize at least 90 classes of objects. It will detect, recognize, and locate up to 10 objects within the scene captured by the camera. The location of each object detected is surrounded with a rectangular box (boundary box) and its name is labeled, along with a score representing the confidence of the accuracy of the detection. The detection results of an object are only displayed and informed when the confidence score is above 0.6 to maintain the accuracy of the application. Besides that, audio feedback is provided to inform users about the name of objects detected through headphones or the smartphone's speaker.

1.7.2 Limitations

Certain scope is not covered in this project due to time constraints and wide coverage of the existing scope. The following are the uncovered scope of this project:

i. The development of this application that does not support iOS-based platforms.

Although iOS smartphones have a large population, this project takes the Android-based application as initial prototype development due to limited time. The application on iOS-based platforms will be developed in the future.

ii. Color detection and recognition

This application is not able to recognize the colors of the detected objects due to the wide scope. Color detection and recognition is useful because it can assist the visually impaired users to determine the colors of the detected objects instead of their name only. Besides, color detection and recognition also assists people with color blindness to identify colors.

iii. Object detection and recognition in the dark

The accuracy of object detection and recognition results in the dark is affected by the camera quality of the smartphone. The application may not work well if the smartphone's camera has a poor performance when shooting in the dark.

1.7.3 Assumptions

i. Assume that users are not completely blind.

ii. Assume that users are not deaf-blind people.

iii. Assume that users have allowed the system to access their smartphone cameras.

iv. Assume that the smartphone of users has enough battery for them to use this application for navigation.

CHAPTER 2

LITERATURE REVIEW

CHAPTER 2

LITERATURE REVIEW

2.1 LITERATURE REVIEW OF JOURNALS

[1] Liu, Wei, et al. "SSD: Single shot multi box detector." European conference on computer vision. Springer, Cham, 2016

In this paper they explained about a rapid development of deep learning, great breakthroughs have been made in the field of object detection. Deep Learning algorithm is used to detect daily objects. Compared with other traditional object detection algorithms, daily object detection uses deep learning to detect objects faster and accurately.

The main research work of this article:

1. Small data of objects are collected for research purposes

2. Tensor Flow framework is used to build for object detection, and use this data set training model to detect objects.

3. Fine-tuning model is used to improve the efficacy of the training process.

Object detection algorithms usually contain three parts. The first is the design of features, the second is the choice of detection window, and the third is the design of the classifier. Feature design methods contain artificial feature design and neural network feature extraction process. Exhaustive Search, Selective Search, and RPN method based on deep learning are the selection window used. This article adopts deep convolutional neural network (CNN) image feature extraction, using the foremost advanced RPN because of the detection window selection method, the bounding box multivariate analysis, using soft max classification processing, and outputs the detection result.

[2] Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." Proceedings of the IEEE conference on computer vision and pattern recognition 2016

This paper presents YOLO which is an innovative approach to object detection. The work prior to object detection re-purposes classifiers to perform detection. Instead, this paper frame object detection has a regression problem to spatially separated bounding boxes and

Associated class probabilities. A solo neural network predicts the class probabilities from full images in one evaluation. The whole detection pipeline is a single network so it can be completely optimized directly from end-to-end.

The unified architecture is extremely fast therefore the base YOLO model processes images in real-time at the rate of 45 frames per second. Fast YOLO, A smaller network version, processes around an astounding 155 fps while still achieving double the map of other real-time detectors. Compared to state-of-the- art detection systems, the YOLO algorithm makes more localization errors but is far less likely to predict false detentions where nothing exists. Finally, YOLO learns very general representations of objects. When compared to DPM and R- CNN, the algorithm performs better than all other detection methods when generalizing from natural images to artwork on both the Picasso Dataset, the People-Art Data set.

[3] Ren, Shaoqing, et al. "Faster R-CNN: towards real-time object detection with region proposal networks." IEEE transactions on pattern analysis and machine intelligence 39.6 (2016)

In this study it is discovered that Fast R-CNN have reduced the running time of the detection. Here the concept of Region Proposal Network (RPN) is also studied that shares full-image convolution features with the network, thus enabling cost-free region proposals. RPN is a complete convolution network that simultaneously predicts bounds of the object and scores at each position. It is trained end-to-end to generate high-quality region proposals, which are then used by Fast R-CNN for object detection. Later RPN and Fast R-CNN are merged into a single network by sharing their convolution features.

[4] Zhao, Zhong-Qiu, et al. "Object detection with deep learning: A review." IEEE transactions on neural networks and learning systems 30.11 (2019): 3212-3232

This paper explains how the technique uses the frame difference and edge detection for object detection. Intelligent survey is used to detect moving objects. In this paper, an algorithm is improved based on frame difference versus edge detection that are presented for the moving object detection. First, it detects the edges of each two continuous frames and gets the difference between the two edge images. And then, it divides the edge difference image into several small blocks and decides if moving areas are present then comparison between the number of non-zero pixels and the threshold occurs. At last, a block-connected component occurs labelling to get the smallest rectangle that shows the moving object from the frame.

Results show the algorithm improvement overcomes the frame difference method. It has a high recognition rate and a high detection speed, which is more reliable on all fields.

[5] Karayaneva, Yordanka, and Diana Hinter. "Object Recognition in Python and MNIST Data set Modification and Recognition with Five Machine Learning Classifiers." Journal of Image and Graphics 6.1 (2018).

This project uses the humanoid robot NAO which provides object detection and recognition of colors, shapes and handwritten digits and letters. A total of five classifiers are used which include neural networks that are used for the handwritten recognition of digits and letters. The accuracy range of the object detection algorithms are between 82%-92%. The five classifiers used in the algorithm for handwritten digits and letters produce highly accurate results which are within the range of 87%-98%. This project will serve as a promising provision for an affectionate touch for children and young students.

[6] Al Khalid, Farah F., Bashra Kadhim Oleiwi, and M. Abdul Muhsin. "Real Time Blind People Assistive System Based on Open-CV." Journal of University of Babylon for Engineering Sciences 28 (2020): 25-33.

This journal proposes that everyone should deserve to live independently. It is especially for the people who are disabled in the last decade. He suggests using modern technology to give attention to the disabled which makes them control their life as independently as possible. In this, the assistive system for the blind is suggested to let them feel and know what is around him. The project uses YOLO for detecting objects within images and video streams which are based on deep neural networks to make accurate detection, and Open-CV under Python using Raspberry Pi3. The result thus obtained indicated success of the proposed model which gives the users the capability to move around in unfamiliar indoor environments with the help of a user-friendly device through person and object identification model.

[7] Sujeetha, R., et al. "Cyberspace and Its Menaces." 2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN). IEEE, 2019.

In this study the difficulties faced in implementing object detection and recognition in real life is discussed. As this is used widely in security and surveillance these issues must be non-existent and as accurate as possible. These difficulties can be found during different situations like multiple objects in a frame, time object, and climate conditions. Several techniques have

been devised but still lies a lot of scope of improvement, however during this study new techniques of object detection and tracking was found. These new techniques include Tensor Flow and Open-CV library and CNN algorithm where the detected layers will be labelled with accuracy being checked at the same time. Validation is done by live input video where objects will be getting detected and it can be simulated the same for real-time through external hardware added. In the end we could see the properly efficient and optimized algorithm for object tracking and detection.

[8] Geethapriya, S., N. Duraimurugan, and S. P. Chokkalingam. "Real-time object detection with YOLO." International Journal of Engineering and Advanced Technology (IJEAT) 8.3S (2019)

The Objective proposed for this project is to detect objects using the You Only Look Once (YOLO) approach. This algorithm has several advantages as compared to other object detection algorithms. In other algorithms like Fast Convolution Neural Network and Convolution Neural Network (CNN), the algorithm won't look at the image completely. Whereas in the YOLO approach, the algorithm looks at the image completely by predicting the bounding boxes using Convolution Neural Network and the probabilities of class for those boxes and it detects the image much faster as compared to other algorithms.

CHAPTER 3

PROBLEM DEFINITION

CHAPTER 3

PROBLEM DEFINITION

Globally the number of people of all ages visually impaired is estimated to be 285 million, of whom 39 million are blind. People 50 years and older are 82% of all blind, they cannot live their own life without other's help. But with the help of computer vision, we can train the computer to identify the object and give them a voice Feedback So that they can Experience the real-world object with the help of Computer vision.

Blind and visually impaired individuals face significant challenges in navigating their environment and identifying objects due to their lack of sight. This can lead to difficulties with daily tasks, decreased independence, and safety concerns.

3.1 Current Limitations:

Reliance on sighted assistance for object identification.

Difficulty in understanding spatial layout and object placement.

Increased risk of accidents due to unidentified obstacles or hazards.

Limited access to information about surroundings that sighted people take for granted (e.g., clothing labels, food packaging).

3.2 Desired Outcomes:

Develop an assistive technology system that leverages object recognition and voice feedback to:

Increase Independence: Allow blind individuals to identify objects in their surroundings independently, improving their ability to navigate and perform daily tasks.

Enhance Safety: Provide real-time object recognition, including warnings about potential hazards, to promote a safer environment.

Improve Spatial Awareness: Describe the location and orientation of objects, fostering a better understanding of the surrounding layout.

Increase Access to Information: Provide spoken descriptions of objects beyond just their names, including labels and other relevant details.

3.3 Challenges to Address:

Accuracy: Object recognition algorithms can struggle with complex environments, variations in lighting, and similar objects.

Real-Time Performance: The system needs to identify objects quickly and accurately in real-time scenarios for efficient use.

Affordability: The technology should be accessible and affordable for a wide range of users.

Privacy Concerns: The use of cameras raises privacy considerations that need to be addressed.

Usability: The system interface must be user-friendly and accessible for blind individuals, potentially including voice commands and haptic feedback.

Object Description: The system should not only identify objects but also describe them in a way that is informative and helpful for the user (e.g., size, colour, location).

By overcoming these challenges, object recognition with voice has the potential to be a transformative technology for blind and visually impaired people.

CHAPTER 4

SYSTEM SPECIFICATION

CHAPTER 4

SYSTEM SPECIFICATION

4.1 HARDWARE REQUIREMENTS:

- System : AMD Ryzen / intel
- Hard Disk : 156 GB and above
- Monitor : 15 VGA Colour.
- Mouse : any
- Ram : 8 GB.

4.2 SOFTWRE REQUIREMENTS:

- Operating system : Windows 10
- Coding Language : Python / ML
- Data Set : COCO

CHAPTER 5

PROPOSED SYSTEM

CHAPTER 5

PROPOSED SYSTEM

Object detection is a computer vision technique that allows us to identify and locate objects in an image or video. This technique has its roots in the field of computer science. In this project, we will Train our System to identify objects on its own using Python Script and the identified object will be returned as a voice feedback and the result will be implemented on the Raspberry-pi connected with a camera and an earphone or speaker.

5.1. METHODOLOGY

Object Detection using computer vision In this project we use a Computer Vision Algorithm known as YOLO to identify objects and Open CV to capture the image. Our system will detect the objects stored in a partially per-trained model and label them with a rectangular box around them. Then the identified objects will be returned to the original frame.

YOLO:

YOLO is the most used algorithm for object detection. As the name of the algorithm suggests the algorithm will look only once and detect the object.

YOLO will use 3 attributes to detect the object.

1. Centre of a bounding box

2. Width, Height

3. Value c is corresponding to a class of an object

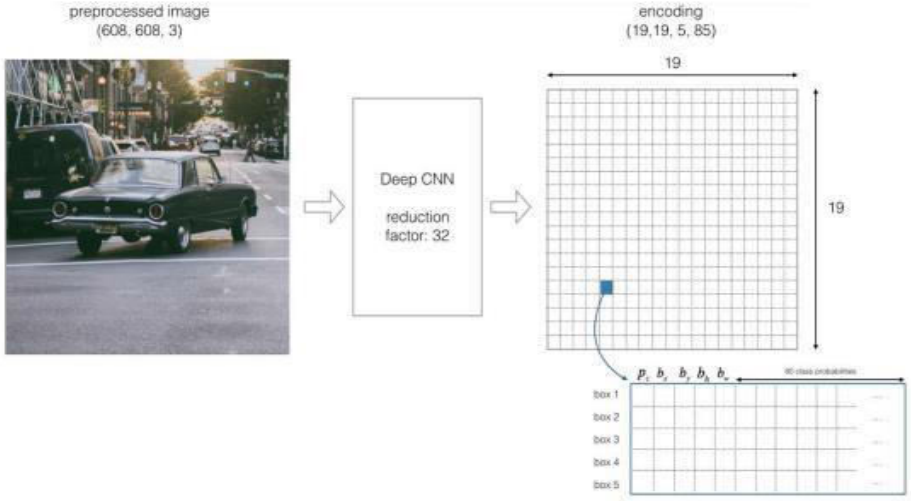

Fig 5.1.1 Representing YOLO grids and bounding boxes.

YOLO will divide the image into a 19x19 grid and each cell has 5 bouncing boxes so in total there will be 1805 bouncing boxes in a single whole image. (i.e) more bouncing box is equals to faster and more reliable prediction. After the bouncing boxes identify the object in the image it uses a technique called non-max suppression to identify and combine the identified grids.

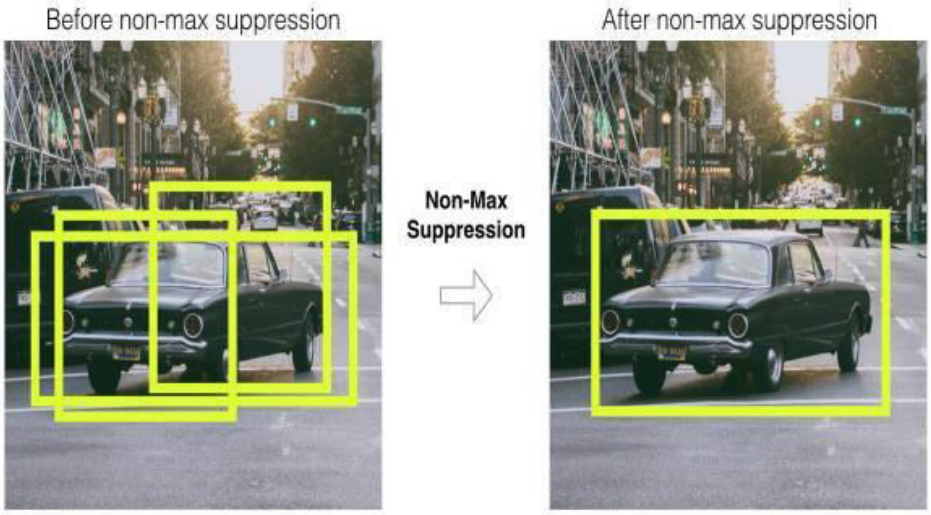

Fig 5.1.2 Representing YOLO applying Non-max Suppression

5.2 OPENCV:

Open CV is an open-source tool for computer vision and ML library. It is the most used library for computer vision and application that runs on machine learning algorithms. It has more than 2500 libraries.

Open CV is mostly used in:

- ❖ 2D and 3D feature tools
- ❖ Ego-motion identifier
- ❖ Facial Detection
- ❖ Gesture Recognition
- ❖ Human–computer interaction
- ❖ Mobile robotics
- ❖ Motion understanding
- ❖ Object identification
- ❖ Segmentation and recognition
- ❖ Stereo-phis stereo vision: depth perception from 2 cameras
- ❖ Structure from motion (SFM)

5.3 DATASET AND PACKAGES

5.3.1 DATASET

COCO Dataset stands for Common Objects in Context. It is a pre-defined dataset by Microsoft. The COCO dataset is a collection of demanding, high-quality datasets for computer vision, mostly using state-of-the-art neural networks. This name is also used to refer to the format in which the datasets are stored. It is an object detection, segmentation, and captioning dataset. It contains around 330k images in which more than 200k images are

labelled, which makes it even easier to recognize the class (category) of detected object. It has around 1.5 million object instances and 80 object categories. COCO annotations employ the JSON file format, which has a top value of dictionary (key-value pairs inside braces). It can also have nested dictionaries or lists (ordered collections of objects inside brackets), as shown below:

{

"info": {…},

"licenses": […],

"images": […],

"categories": […],

"annotations": […]

}

Info Section: It contains metadata about the dataset like description, url, version etc.

Licenses section: It contains the links to the licenses for the images present in the dataset. All the license contains the id field which is used to recognize the license.

Image: It is the second most important dictionary of the dataset. It has the fields like licence, file_name, coco_url, height, width and date captured.

Categories Section: It contains classes of the objects that may be detected on images.

Annotations Section: This is the most important section of the dataset, which contains information vital for each task for specific COCO dataset.

5.3.2 PACKAGES

The following packages has been used for building the model:

(i) NumPy: NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python. NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using Numpy which allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

(ii) OpenCV: Open Source Computer Vision. It is one of the most widely used tools for computer vision and image processing tasks. It is used in various applications such as face detection, video capturing, tracking moving objects, object disclosure. now it plays a major role in real-time operation which is very important in today's systems. By using it, one can process images and videos to identify objects, faces, or even handwriting of a human.

(iii) gTTs: gTTS is a very easy to use tool which converts the text entered, into audio which can be saved as a mp3 file. The gTTS API supports several languages including English, Hindi, Tamil, French, German and many more. The speech can be delivered in any one of the two available audio speeds, fast or slow. However, as of the latest update, it is not possible to change the voice of the generated audio.

5.4 ALGORITHM

In order to build the model, we have cross validated various ML classification algorithms.

However, the high accuracy rated algorithms that we have further used to fit the model shall be discussed henceforth.

5.4.1 YOLO v3:

The term 'You Only Look Once' is abbreviated as YOLO. This is an algorithm for detecting and recognising different items in a photograph (in real-time). Object detection in YOLO is done as a regression problem, and the identified photos' class probabilities are provided. Convolutional neural networks (CNN) are used in the YOLO method to recognise objects in real time. To detect objects, the approach just takes a single forward propagation through a neural network, as the name suggests. This indicates that a single algorithm run is used to forecast the entire image. The CNN is used to forecast multiple bounding boxes and class probabilities at the same time. It is an instantaneous object identification algorithm that has a COCO test-dev mAP of 57.9% while analyzing images at 30 frames per second rate. The main features of YOLOv3 lie in it being very fast and accurate, which can easily be traded off by simply customizing the size of the model, thereby requiring no retraining whatsoever.

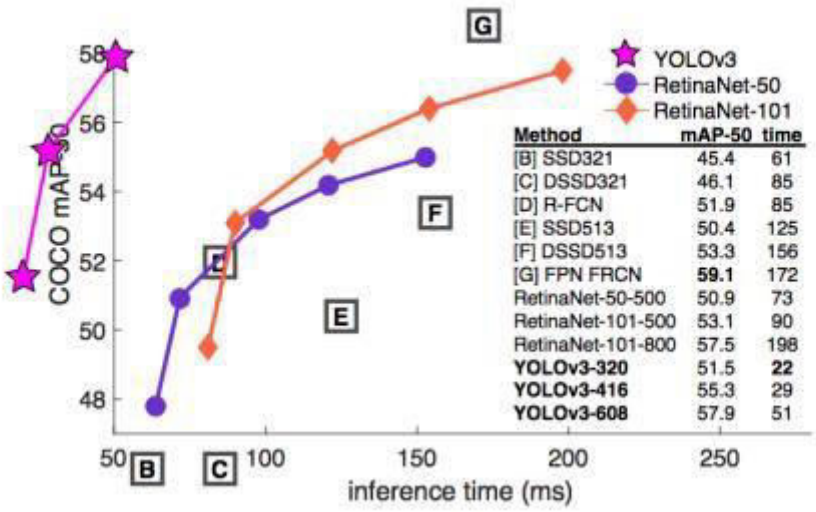

Fig5.4.1: YOLOv3 comparison

5.4.1.1 Working of YOLO algorithm

YOLO is regression-based. Initially it takes the video input and segments the video into 24 frames. Each frame is then divided into cells. Image classification and localization are applied on each grid. YOLO then predicts the bounding boxes and their corresponding class probabilities for objects. To break down it into simpler terms, the labelled data let's say is divided into 3x3 grids and there are total of 3 classes in which we want it to be classified. So, for each grid cell, label y will be defined as eight dimensional vector.

Y = pc, bx, by, bh, bw, c1, c2, c3

Here,

pc is the probability of whether the object is present in the grid or not.

bx, by, bh, bw specify the bounding box if there's an object, and c1, c2, c3 are the classes of the detected objects.

Bounding boxes i.e bx, by, bh and bw are calculated relative to the grid cell it is a dealing with. bx and by are the x and y coordinates of the midpoint of the object with respect to this grid.bh is the ratio of the height of the bounding box to the height of the corresponding grid cell. bw is the ratio of the width of the bounding box to the width of the grid cell. bx and by will always range between 0 and 1 as the midpoint will always lie within the grid. Whereas bh and bw can be more than 1 in case the dimensions of the bounding box are more than the dimension of the grid

5.4.1.2. Intersection over Union and Non-Max Suppression

How can we tell if the anticipated bounding box is producing a good (or terrible) result? This is where the concept of Intersection over Union comes into play. It computes the intersection of the actual bounding box and the predicted bonding box over their union. IoU, or Intersection over Union, will calculate the area of the intersection over union of these two boxes. That area will be:

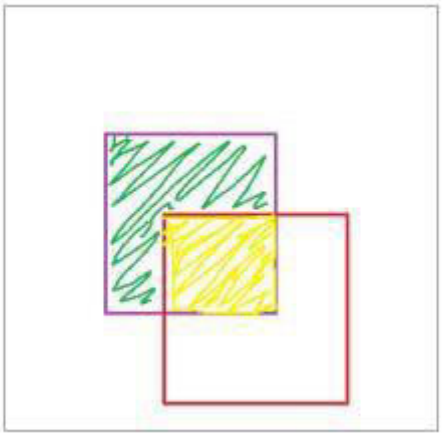

Fig 5.4.2: Intersection over Union visualization

IoU = Area of the intersection / Area of the union,

i,e, IoU = Area of yellow box / Area of green box

If IoU is greater than 0.5, we can say that the prediction is good enough. 0.5 is an arbitrary threshold we have taken here, but it can be changed according to the specific problem. Intuitively, the more we increase the threshold, the better the predictions become.

There is one more technique that can improve the output of YOLO significantly – Non-Max Suppression. To avoid the multiple detection of object in the single frame, Non-Max Suppression technique is used. It first looks at the probabilities associated with each detection and takes the largest one. Then, it looks at all the other boxes in the image. The boxes which have high IoU with the current box are suppressed. After the boxes have been suppressed, it selects the next box from all the boxes with the highest probability, this process is repeated and all the boxes have either been selected or compressed and the final bounding boxes are calculated.

5.4.1.3. Training and Testing:

The input for training our model will obviously be images and their corresponding y labels.

Consider the scenario where we are using a 3 X 3 grid with two anchors per grid, and there are 3 different object classes. So the corresponding y labels will have a shape of 3 X 3 X 16.

Now, suppose we use 5 anchor boxes (multiple bounding boxes in a single grid) per grid and the number of classes has been increased to 5. So the target will be 3 X 3 X 10 X 5 = 3 X 3 X 50. This is how the training process is done – taking an image of a particular shape and mapping it with a 3 X 3 X 16 target (this may change as per the grid size, number of anchor boxes and the number of classes).

The new image will be divided into the same number of grids which we have chosen during the training period. For each grid, the model will predict an output of shape 3 X 3 X 16 (assuming this is the shape of the target during training time).

The 16 values in this prediction will be in the same format as that of the training label. The first 8 values will correspond to anchor box 1, where the first value will be the probability of an object in that grid. Values 2-5 will be the bounding box coordinates for that object, and the last three values will tell us which class the object belongs to. The next 8 values will be for anchor box 2 and in the same format, i.e., first the probability, then the bounding box coordinates, and finally the classes.

Finally, the Non-Max Suppression technique will be applied on the predicted boxes to obtain a single prediction per object.

5.4.1.4. DARKNET ARCHITECTURE

Yolo V3 is an improvement over the previous two YOLO versions where it is more robust but a little slower than its previous versions. This model features multi-scale detection, a stronger feature extraction network, and a few changes in the loss function. For understanding the network architecture on a high-level, let's divide the entire architecture into two major components: Feature Extractor and Feature Detector (Multi-scale Detector). The image is first given to the Feature extractor which extracts feature embedding and then is passed on to the feature detector part of the network that spits out the processed image with bounding boxes around the detected classes.

5.4.1.5 Feature Extractor

Darknet-19 (a custom neural network architecture developed in C and CUDA) was utilised as a feature extractor in prior YOLO versions, with 19 layers as the name suggests. Darknet-19 now has a total of 30 layers thanks to YOLO v2, which adds 11 extra layers. However, because to the down sampling of the input image and the loss of fine-grained characteristics, the system had difficulty detecting small objects.

The feature extractor utilised in YOLO V3 was a combination of YOLO v2, Darknet-53 (an ImageNet-trained network), and Residual networks, which resulted in a better architecture (ResNet). The network is formed with consecutive 3x3 and 1x1 convolution layers followed by a skip connection, resulting in a total of 53 convolution layers (thus the name Darknet-53) (introduced by ResNet to help the activations propagate through deeper layers without gradient diminishing).

The darknet's 53 layers are piled on top of another 53 for the detection head, giving YOLO v3 a total of 106 layers of fully convolutional underlying architecture. As a result, it has a huge architecture, which makes it a little slower than YOLO v2, but improves accuracy at the same time.

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	

Fig 4.3: Darknet-53 architecture taken from YOLO: An Incremental Improvement

If the aim was to perform classification as in the ImageNet, then the Average pool layer, 1000 fully connected layers, and a SoftMax activation function would be added as shown in the image, but in our case, we would like to detect the classes along with the locations, so we would be appending a detection head to the extractor. The detection head is a multi-scale detection head hence, we would need to extract features at multiple scales as well.

5.4.1.6 Multi-Scale Detector

For visualizing how the multi-scale extractor would look like, I'm taking an example of a 416x416 image. A stride of a layer is defined as the ratio by which it down samples the input, and hence the three scales in our case would be 52x52, 26x26, and 13x13 where 13x13 would be used for larger objects and 26x26 and 52x52 would be used for medium and smaller objects.

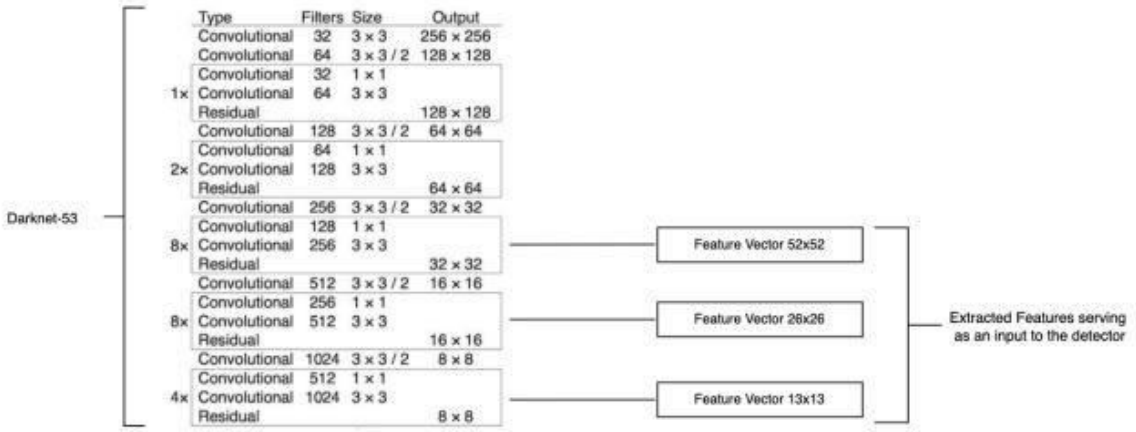

Fig 4.4: Multi scale Feature Extractor for 416x416 image

An important feature of the YOLO v3 model is its multi-scale detector, which means that the detection for an eventual output of a fully convolutional network is done by applying 1x1 detection kernels on feature maps of three different sizes at three different places. The shape of the kernel is 1x1x(B*(5+C)). The complete network architecture can be explained as in below.

5.4.1.7 Complete Network Architecture

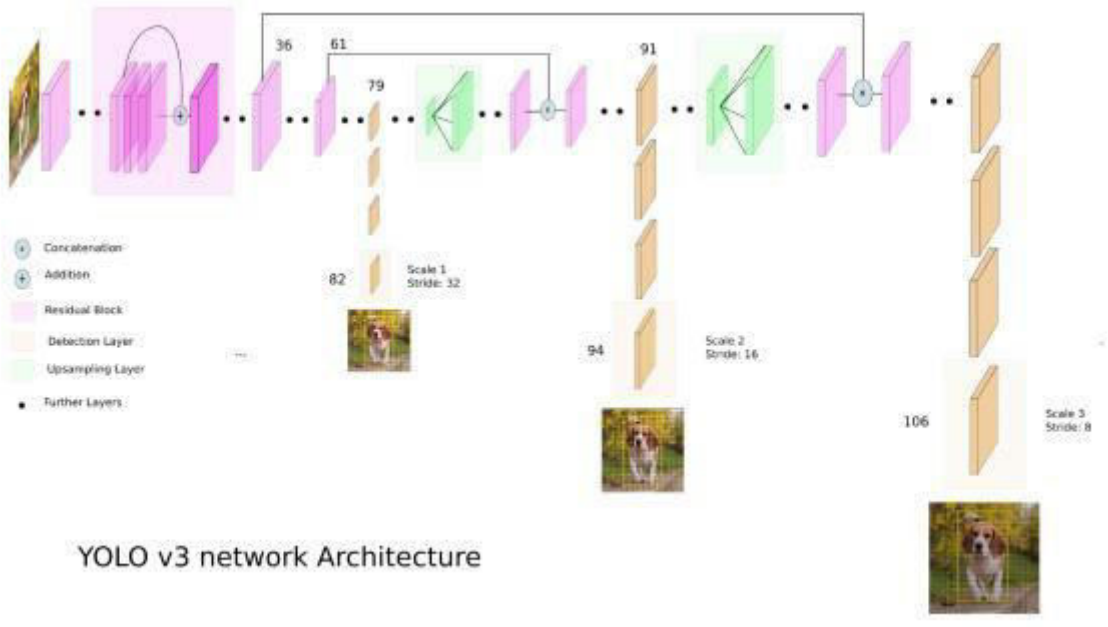

Fig 4.1.7.1: Complete architecture of YOLO v3 combining both the extractor and the detector

The three scales where the detections are produced are at the 82nd layer, 94th layer, and 106th layer, as seen in the above image of a 416x416 image.

For the first detection, the first 81 layers are down sampled until the 81st layer has a stride of 32 (a layer's stride is defined as the ratio by which it down samples the input), resulting in a 13x13 feature map and a 1x1 kernel, resulting in a 13x13x255 detection 3D tensor.

For the second detection, convolutional layers are used from the 79th layer onwards before up sampling to 26x26 dimensions. This feature map is then depth concatenated with layer 61's feature map to create a new feature map, which is then fused with the 61st layer using 1x1 convolution layers. The second detection layer, with a 3D tensor of dimension 26x26x255, is located at the 94th layer.

The feature map of the 91st layer is subjected to convolution layers before being depth concatenated and fused with a feature map from the 36th layer for the final(third) detection layer, which follows the same process as the second detection layer. With a feature map of size 52x52x255, the final detection is made at the 106th layer.

The multi-scale detector is used to make certain that the small gadgets also are being detected not like in YOLO v2, in which there has been steady grievance concerning the same. Up sampled layers concatenated with the preceding layers grow to be retaining the fine-grained capabilities which assist in detecting small gadgets

5.5 FLOW DIAGRAM

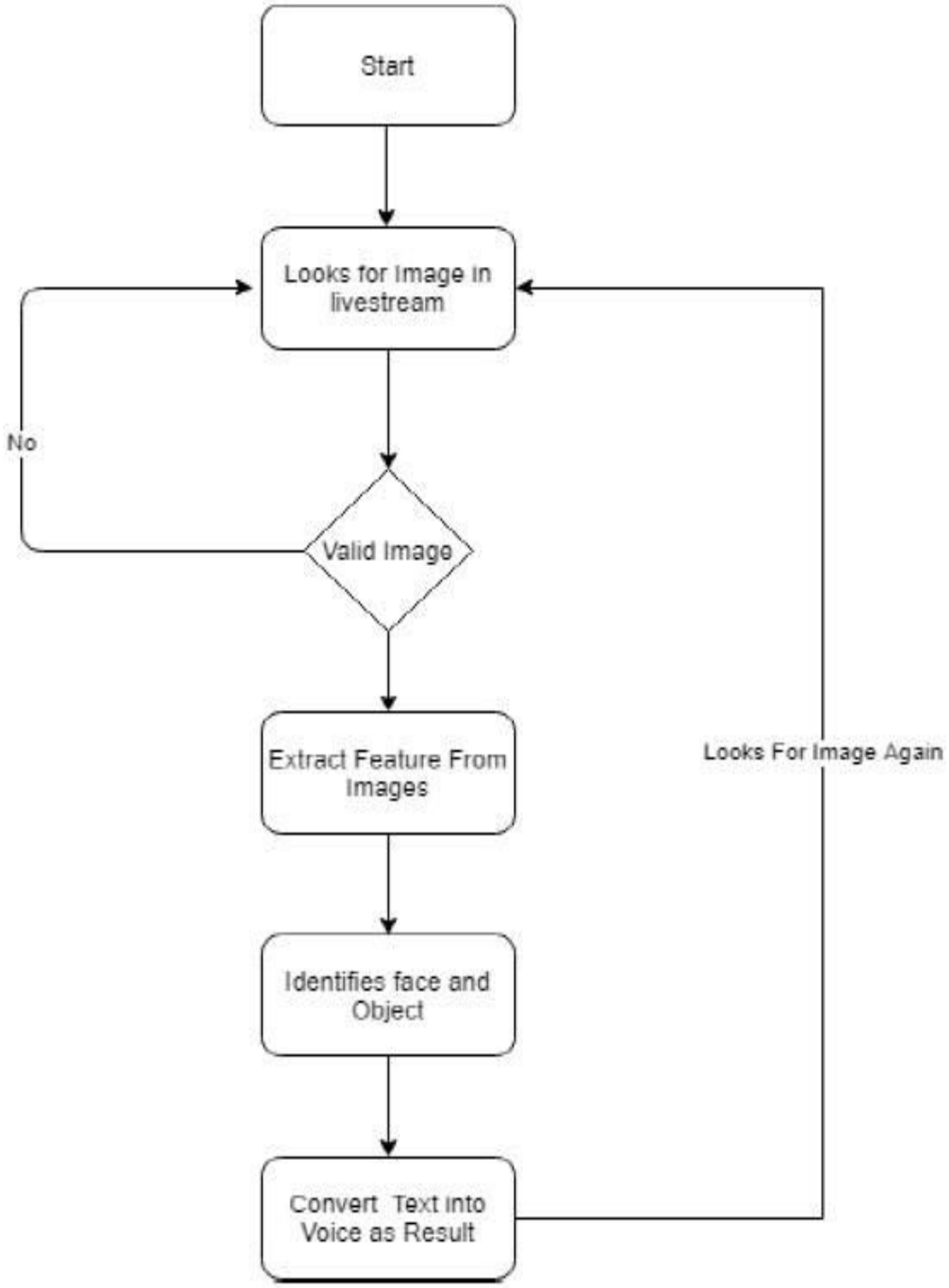

Fig 5.5.1 Flowchart explaining how the project works

CHAPTER 6

SYSTEM DESIGN

CHAPTER 6

SYSTEM DESIGN

6.1 Usecase

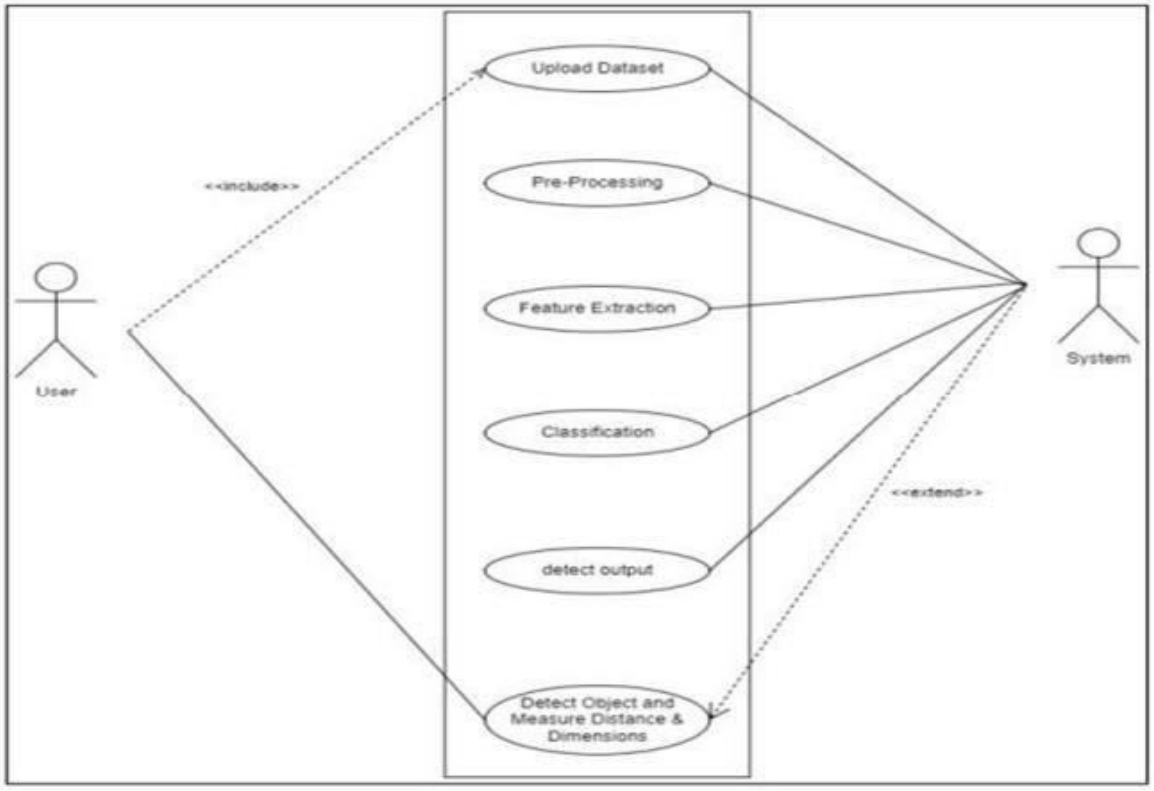

Fig. 6.1 use case Diagram

6.2 Sequence Diagram

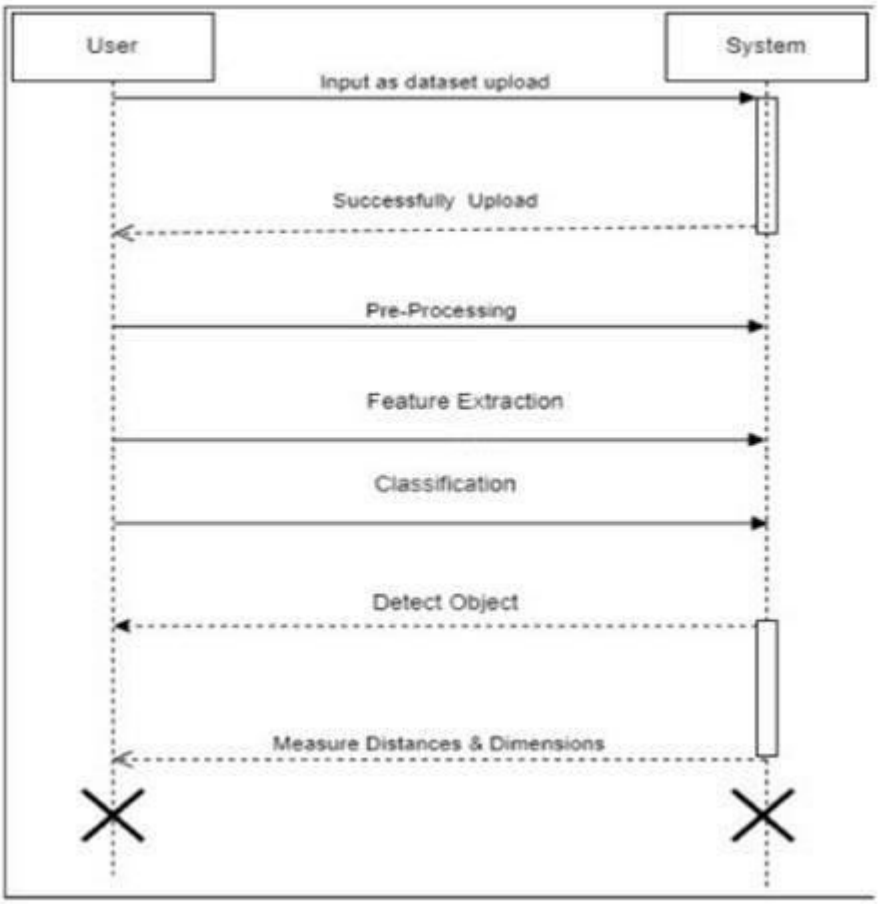

Fig. 6.2 Sequence Diagram

CHAPTER 7

IMPLEMENTATION AND RESULT

CHAPTER 7

IMPLEMENTATION AND RESULT

There are 2 main components included in this project

1. Camera module (hardware)

2. Earphone

7.1 Text to speech

Text-to-speech (TTS) is a type of speech synthesis application that is used to create a spoken sound version of the text in a computer document, such as a help file or a Web page. TTS can enable the reading of computer display information for the visually challenged person, or may simply be used to augment the reading of a text message. Current TTS applications include voice-enabled e-mail and spoken prompts in voice response systems. TTS is often used with voice recognition programs. Like other modules the process has got its own relevance on being interfaced with, where Raspberry Pi finds its own operations based on image processing schemes. So once image gets converted to text and thereby it could be converted from text to speech. Character recognition process ends with the conversion of text to speech and it could be applied at anywhere.

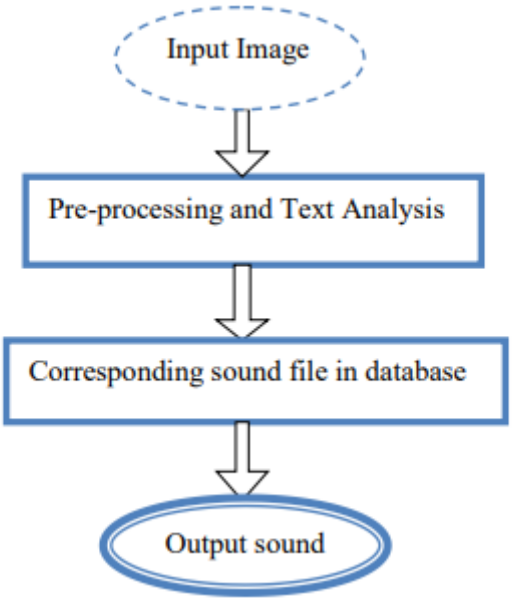

Fig. 7.1- Text to Sound Conversion

7.2 Performance Analysis

To analyze the performance of YOLO, it compared with algorithms like RCNN, fast RCNN, faster R-CNN on various performance measures like time taken, accuracy and the frames per second. When analysis was done based on time taken by the algorithm to detect the objects as listed in table 1, it is found that R-CNN takes around 40 to 50 seconds, fast R-CNN takes 2 seconds, faster R-CNN takes 0.2 seconds, and YOLO takes just 0.02 seconds. From this analysis it can be inferred that, YOLO performs 10 times quicker that faster R-CNN, 100 times quicker than fast RCNN and more than 1000 times quicker than R-CNN.

Algorithm	Time taken (in sec)
R-CNN	40-50
Fast R-CNN	2
Faster R-CNN	0.2

YOLO	0.02

Table 7.2.1: Performance Evaluation Based on Time Taken

When analysis was done based on the number of frames per second, YOLO performs far better than all other algorithms as shown in Fig. 11, with 48 fps whereas, RCNN processes 2 fps, fast R-CNN processes 5 fps and faster R-CNN processes 8 fps.

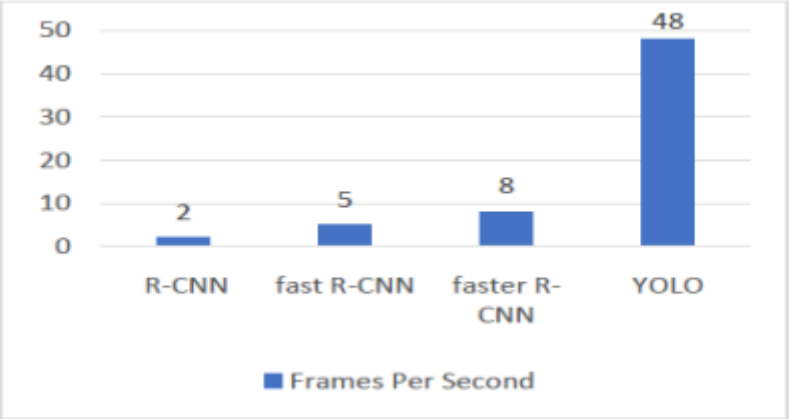

Fig. 7.2.2: -Performance Analysis based on frames per second

When analysis was done based on the accuracy it is found that YOLO has lesser accuracy than the other three algorithms as shown in fig.12. So, it is not recommended to use YOLO for applications in which accuracy is the major concern.

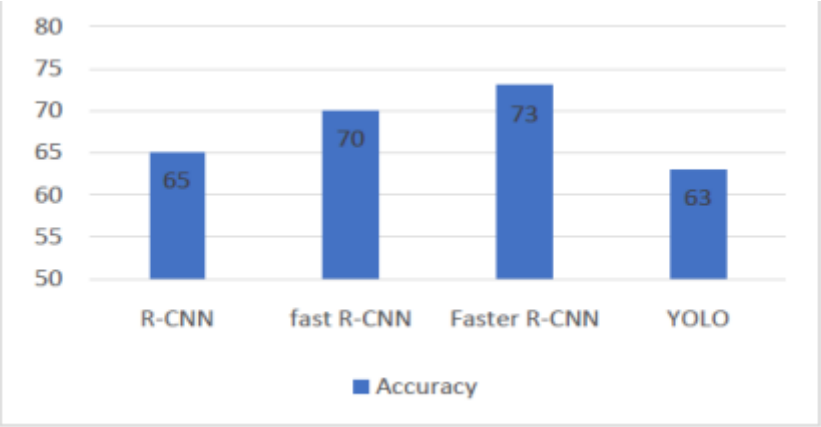

Fig. 7.2.3: -Performance Analysis based on Accuracy

The model can be used in tracking objects for example tracking a ball during a football match, tracking movement of a cricket bat, tracking a person in a video, video surveillance, Smart Class for students, Instructor for blind people to get details about unknown objects. It is also used in Pedestrian detection.

7.3 Properties

5.5.1 Face detection:

An example of object detection in daily life is that when we upload a new picture in Facebook or Instagram it detects our face using this method.

• People Counting:

Object detection can be also used for people counting,it means that it is used for analyzing store performance or crowd statistics during festivals where the people spend a limited amount of time and other details .This type of analysis is little difficult as people move away from frame.

• Vehicle detection:

When the object is a vehicle such as a bicycle or car or bus, object detection with tracking can prove effective in estimating the speed of the object. The type of ship entering a port can be determined by object detection based on the shape, size etc. This method of detecting ships has been developed in certain European Countries.

• Manufacturing Industry:

Object detection is also used in industrial processes to identify products. If we want our machine to detect products which are only circular, we can use Hough circle detection transform can be used for detection

• Online images:

Apart from these object detections can be used for classifying images found online. Obscene images are usually filtered out using object detection.

7.4Testing:

Object Name	Number of Tries	Detection Ratio	Pass	Failure	Percentage Error
person	**50**	**91**	**49**	**1**	**2%**
backpack	**50**	**93**	**49**	**1**	**2%**
bottle	**50**	**94**	**49**	**1**	**2%**
cup	**50**	**91**	**48**	**1**	**2%**
banana	**50**	**93**	**49**	**1**	**2%**
apple	**50**	**93**	**49**	**1**	**2%**
spoon	**50**	**94**	**49**	**1**	**2%**
bowl	**50**	**91**	**47**	**2**	**3%**
chair	**50**	**91**	**47**	**2**	**3%**
laptop	**50**	**94**	**48**	**1**	**3%**
tv	**50**	**95**	**49**	**1**	**3%**
mouse	**50**	**99**	**49**	**1**	**3%**
keyboard	**50**	**91**	**46**	**3**	**6%**
cell phone	**50**	**91**	**47**	**3**	**6%**
book	**50**	**89**	**45**	**3**	**3%**
clock	**50**	**92**	**48**	**1**	**3%**
scissors	**50**	**91**	**48**	**1**	**3%**
remote	**50**	**94**	**48**	**1**	**3%**
toothbrush	**50**	**91**	**47**	**2**	**3%**

Table 7.4.1: Test of Objects

7.4 System Result

Fig 7.4.1: Represents YOLO detecting the whole room

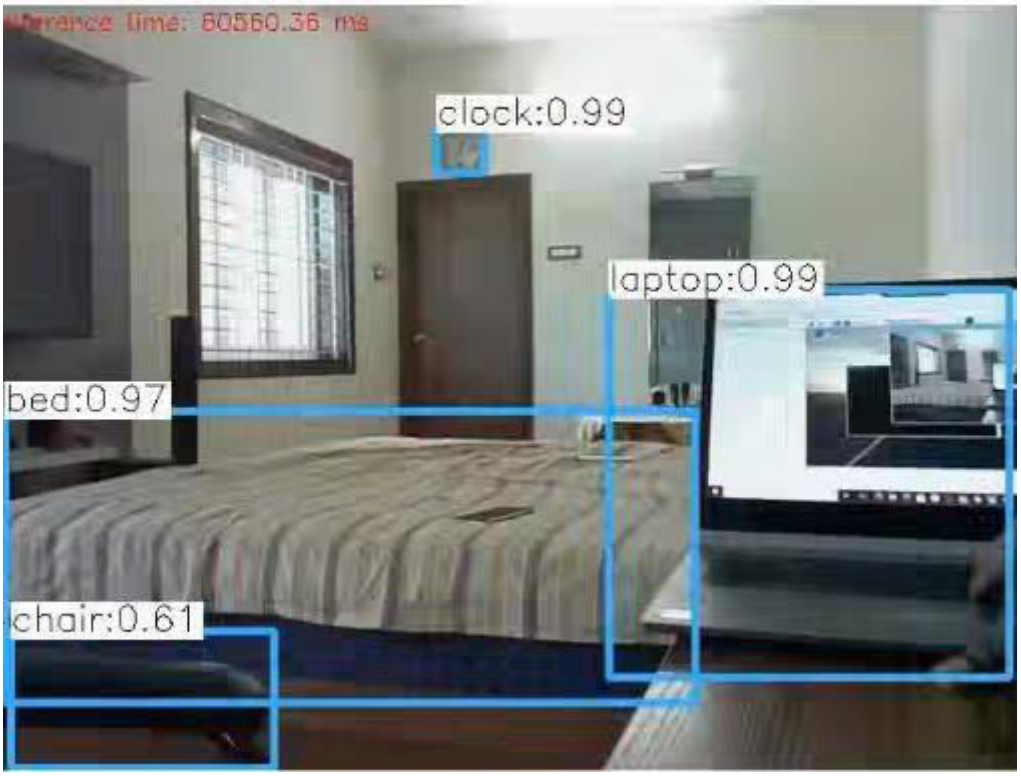

Fig7.4.2 Represents Real time object detection

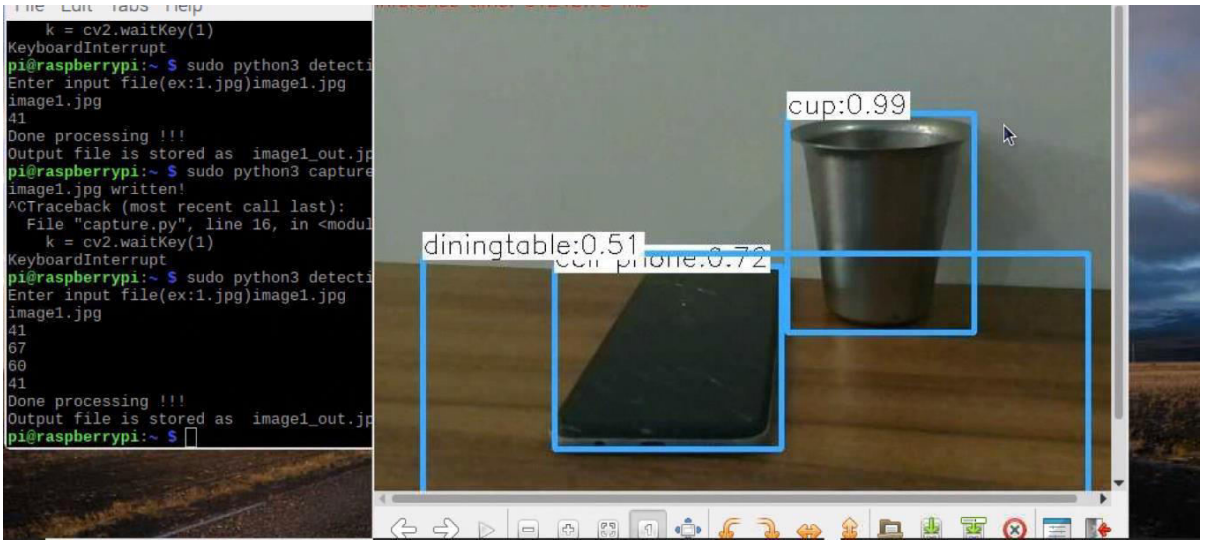

Fig 7.4.3 Represents YOLO capturing multiple object at the same time

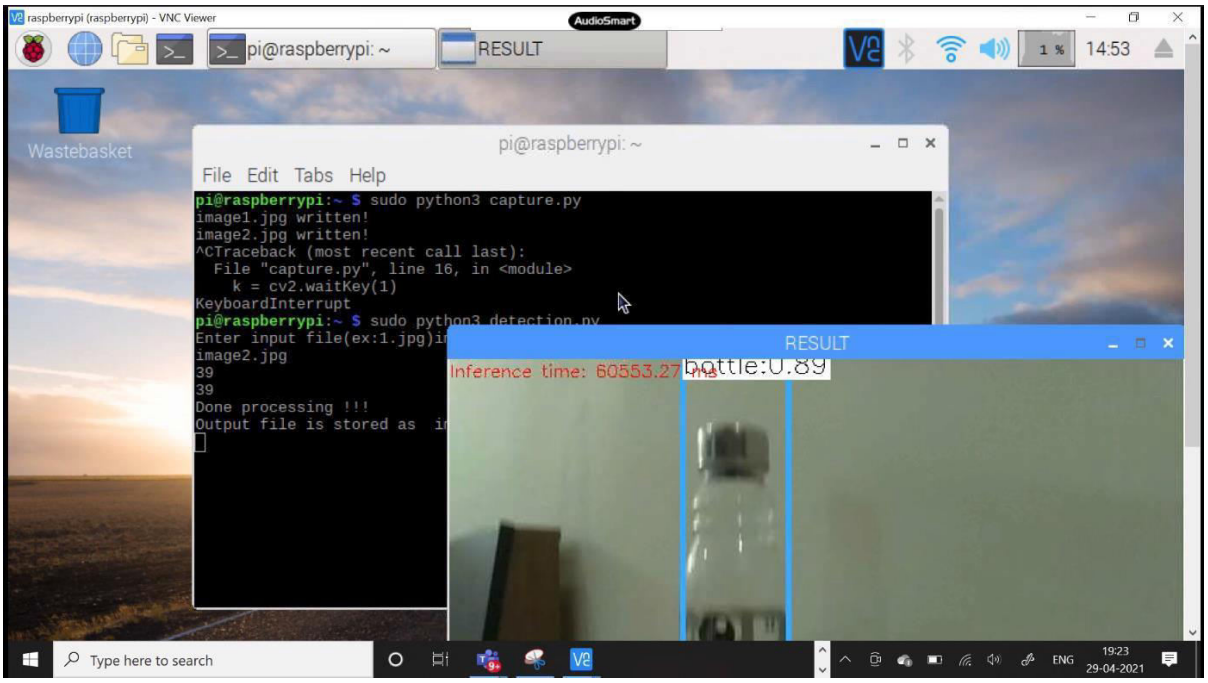

Fig 7.4.4 Detecting partially detected object

CHAPTER 8

CONCLUSION

CHAPTER 8

CONCLUSION

We introduce YOLO, a unified model for object detection. Our model is simple to construct and can be trained directly on full images. Unlike classifier-based approaches, YOLO is trained on a loss function that directly corresponds to detection performance and the entire model is trained jointly. Fast YOLO is the fastest general-purpose object detect-tor in the literature and YOLO pushes the state-of-the-art in real-time object detection. YOLO also generalizes well to new domains making it ideal for applications that rely on fast, robust object detection and text to speech is also done.

Future Scope

Object detection is a key ability for most computer and robot vision system. Although great progress has been observed in the last years, and some existing techniques are now part of many consumer electronics (e.g., face detection for auto-focus in smartphones) or have been integrated in assistant driving technologies, we are still far from achieving human-level performance, in particular in terms of open-world learning. It should be noted that object detection has not been used much in many areas where it could be of great help. As mobile robots, and in general autonomous machines, are starting to be more widely deployed (e.g., quad-copters, drones and soon service robots), the need of object detection systems is gaining more importance. Finally, we need to consider that we will need object detection systems for Nano robots or for robots that will explore areas that have not been seen by humans, such as depth parts of the sea or other planets, and the detection systems will have to learn to new object classes as they are encountered. In such cases, a real-time open-world learning ability will be critical.

CHAPTER 9

REFERENCES

CHAPTER 9
REFERENCES

[1] Liu, Wei, et al. "Ssd: Single shot multibox detector." European conference on computer vision. Springer, Cham, 2016.

[2] Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[3] Ren, Shaoqing, et al. "Faster R-CNN: towards real-time object detection with region proposal networks." IEEE transactions on pattern analysis and machine intelligence 39.6 (2016): 1137-1149.

[4] Zhao, Zhong-Qiu, et al. "Object detection with deep learning: A review." IEEE transactions on neural networks and learning systems 30.11 (2019): 3212-3232.

[5] Karayaneva, Yordanka, and Diana Hintea. "Object Recognition in Python and MNIST Dataset Modification and Recognition with Five Machine Learning Classifiers." Journal of Image and Graphics 6.1 (2018).

[6] Alkhalid, Farah F., Bashra Kadhim Oleiwi, and M. Abdul Muhsin. "Real Time Blind People Assistive System Based on OpenCV." Journal of University of Babylon for Engineering Sciences 28 (2020).

[7] Sujeetha, R., et al. "Cyber-Space and Its Menaces." 2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN). IEEE, 2019.

[8] Geethapriya, S., N. Duraimurugan, and S. P. Chokkalingam. "Real-time object detection with Yolo." International Journal of Engineering and Advanced Technology (IJEAT) 8.3S (2019).