### OBJECT DETECTION WITH VOICE OVER

*A Project report submitted in partial fulfilment of the requirements For the award of the Degree of*

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE ENGINEERING**

By

**R. ALEKHYA**

**(319136410132) G. VANDANA**

**(319136410127) P. JABILI SREE**

**(319136410093)**

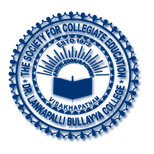
**M. PRATHYUSHA**

**(319136410077)**

Under the esteemed guidance of **Mrs. T. Aruna(PhD)**

Asst Professor

Department of Computer Science Engineering



DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

**Dr. L. BULLAYYA COLLEGE OF ENGINEERING**

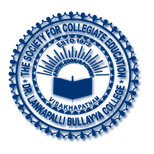
**(Affiliated to Andhra University, Visakhapatnam) New Resapuvanipalem, Visakhapatnam**

Year of submission:2022

# Dr. L. BULLAYYA COLLEGE OF ENGINEERING

***New Resapuvanipalem, Visakhapatnam-530013***

## Department of Computer Science Engineering



### Bonafide Certificate

This is to certify that **Ms**.**R**.**Alekhya, Ms.G.Vandana, Ms.P.Jabili sree, Ms.M.Prathyusha** bearing register numbers **319136410132, 319136410127, 319136410093, 319136410077,** students of Final year B. Tech in Computer Science Engineering, has carried out the project work titled “**OBJECT DETECTION WITH VOICE OVER“** at Dr. L. Bullayya College of Engineering, Visakhapatnam during the academic year **2021-22.**

**Project Supervisor Head of the Department**

Mrs. T. Aruna (PhD) Dr. D. Madhavi

Assistant Professor Professor

Dept. of Computer Dept. of Computer

Science Engineering Science Engineering

### ABSTRACT

The replication of human intelligence is what artificial intelligence (AI) is all about. AI can only mimic, replace, extend, or expand a small portion of human intelligence at this time. In the future, cutting-edge technologies such as brain-computer interface (BCI) research and development, combined with the evolution of the human brain, will usher in a strong AI era, in which AI will be able to simulate and replace human imagination, emotion, intuition, potential, tacit knowledge, and other forms of personalised intelligence. Breakthroughs in cognitive computing algorithms encourage the continued penetration of AI into domains such as education, commerce, and medical care, hence expanding the AI service space.

We want to create an AI-powered device that works similar to a blind stick. We propose an advanced device that uses modern technology to assist vision impaired persons to perceive everything around them with ease. We would be using a camera for object detection. The camera in our proposed idea can be utilised to first detect things in the environment. The camera will collect the relevant photos, and the distance will be measured and recorded in the dataset, with a voice over given by us.

Another feature is that it helps the blind to detect whether the room is dark or light. One additional sophisticated feature of the device is that it can help the blind find their device if they forget where they put it. For this, a wireless rf-based remote is used. As a result, this technology may identify obstacles as well as assist the blind in seeing.The ability to recognize people uniquely and to associate personal attributes such as name and nationality with them has been very important to the fabric of human society. Nowadays, modern societies have an explosion in population growth and increased mobility which necessitated building advanced identity management systems for recording and maintaining people's

#### ACKNOWLEDGEMENT

We would like to express our heartfelt gratitude to **Prof D. Deepak Chowdary**, Principal and to **Sri. Syed Mujib Rahaman**, Vice- Principal, Dr. L. Bullayya College of Engineering for giving us the opportunity for this project.

We would like to thank **Dr. D. Madhavi**, HOD, Department of CSE, Dr. L. Bullayya College of Engineering, Visakhapatnam, for her guidance in producing this work.

We are deeply indebted to our project guide and the head of research cluster “Automated reasoning for human centred interactive system”, **Mrs.T.Aruna** (Ph.D), Assistant Professor, Department of Computer Science Engineering, Dr. Lankapalli. Bullayya College of Engineering, Visakhapatnam, for guiding us throughout the project in spite of her busy schedule.

Apart from our efforts, the success of this project depends largely on the encouragement of other faculty members of CSE, Dr. Lankapalli. Bullayya College of Engineering, Visakhapatnam. We take this opportunity to express my gratitude to the entire faculty who has been instrumental in the successful completion of this project.

Also deserving of thanks for our family and friends, for their support and for their confidence in our achievements.

R.Alekhya(319136410132)

G. Vandana(319136410127)

P. Jabili Sree(319136410093)

M. Prathyusha(319136410077)

#### DECLARATION

This is to declare that the Project work entitled “**OBJECT DETECTION WITH VOICE OVER** ” is a bonafide work done by us under the research cluster group “Automated reasoning for human centred interactive system” with the esteemed guidance of Mrs.T.Aruna(Ph.D), Department of CSE, Dr. L. Bullayya College of Engineering. This project report is being submitted in the partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Computer Science Engineering during the academic year 2021-2022. This project possesses originality as it is not extracted from any source and it has not been submitted to any other institutions and university.

R. Alekhya

Reg.No : 319136410132 G. Vandana

Reg.No : 319136410127

P. Jabili Sree

Reg.No : 319136410093 M. Prathyusha

Reg.No : 319136410077

Place : Visakhapatnam

Date :

#### LIST OF TABLES

|  |  |  |
| --- | --- | --- |
| **S.No** | **Table** | **Page no** |
| **1.** | **Functional Requirements** | **14** |
| **2.** | **Test Cases** | **33** |

#### LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| **S.No** | **Figures** | **Page no** |
| **1.** | **Proposed system** | **13** |
| **2.** | **System Design** | **15** |
| **3.** | **Use Case Diagram** | **20** |
| **4.** | **Activity Diagram** | **21** |
| **5.** | **Class Diagram** | **22** |
| **6.** | **Sequence Diagram** | **23** |
| **7.** | **State Chart Diagram** | **24** |
| **8.** | **ANN** | **26** |
| **9.** | **Code Output** | **36-46** |

#### TABLE OF CONTENTS

**Chapter No Title Page No.**

Abstract

Acknowledgements

Declaration

List of Tables

List of Figures

List of Symbols

1. Introduction 7

1.1.Background and Motivation

1.2. Problem Statement

1. Requirements Elicitation and Analysis 10
   1. Existing System

2.2.Proposed System

2.3.Feasibility Study

2.4.System Requirements

2.4.1.1.Functional Requirements

2.4.1.2Non Functional Requirements

1. System Design 16

3.1..Object Oriented Analysis and Design

* 1. 1.Use Case diagrams
     1. Activity diagrams
     2. Class diagrams
     3. Object diagrams
     4. Other diagrams that are applicable.
  2. Database Design

1. Implementation Details 23

4.1.Software Environment

* + 1. Software Technologies
    2. Database

4.1.2.1.Tables

1. Testing 28
2. Conclusions 30
3. References 31
4. Appendix 32-39 A – Sample Code 32-38

B– Input/Output Screens 39

#### CHAPTER-1

**INTRODUCTION:**

Human-computer interaction (HCI) is a field of research in the design and the use of computer technology, which studies in detail the interfaces between people (users) and computers. HCI researchers focus on the ways in which humans interact with computers and design technologies allowing humans to interact with computers in novel ways.

It’s a known fact that the estimated number of visually impaired people in the world is about 285 million, approximately equal to 20% of the Indian Population. They suffer regular and constant challenges in Navigation especially when they are on their own. They are mostly dependent on someone for even accessing their basic day-to-day needs. So, it’s quite a challenging task and the technological solution for them is of utmost importance and much needed. One such try from our side is that we came up with an Integrated Machine Learning System which allows the Blind Victims to identify and classify Real Time Based Common day-to-day Objects and generate voice feedback and calculates distance which produces warnings whether he/she is very close or far away from the object. The same system can be used for Obstacle Detection Mechanism.

It's a smart device that helps visually impaired people move around more easily. This project focuses on barrier identification, level detection, and determining position in order to help visually impaired persons navigate more easily.

When we can't rely on our own eyes to navigate an unfamiliar area, it becomes a significant problem.

Traditionally we measure the distance of any object using Ultrasonic sensors such as HC-sr04 or any other high frequency devices which generate sound waves to calculate the distance it traverses. However, when you are working with a embedded device to make a compact design which has functionalities such as 1. Object detection (with camera) and

2. Distance measurement.

The system provides object identification, distance calculation, level detection, and positioning for visually challenged people (GPS).

The overall goal of the system is to provide a smart assistant for the blind that provides a sensation of artificial vision by supplying information about the surrounding environment and objects. The embedded system plays a crucial function in this system.

With the recent rapid development of information technology(IT), a lot of research has been carried out to solve inconveniences in everyday life, and as a result ,various conveniences for people have been provided

Nevertheless, there are still many inconveniences for the visually impaired.They have difficulty recognizing simple objects, and it is not easy to distinguish objects that have similar forms.In this paper, we propose an efficient object detection system to help find objects in a certain space without help from others.

##### 1.1 Background and Motivation

An AI based blind device is an innovative device designed for visually disabled people in order to provide them improved navigation and help them in making smart decisions about the selection of path that has no obstacle till a certain distance. Our search space involves searching a best suited path for a blind friend by using a camera from front, left and right that will search the best path which does not have an obstacle at a certain distance. These sense the obstacles through image recognition and direct blind friends to the direction that is clear of any obstruction to a certain distance. The knowledge acquired through this helps to calculate the distance of obstacles.Automatic object detection can be used to detect small objects in a photo and classify them.It can have security usages when trying to detect enemy units in a satellite images or weapons in security cameras. It can also be used to determine the context in which an image is taken by the objects in it.

##### 1.2 Problem Statement

According to the World Health Organization (WHO). There are over 1.3 billion people who are visually across the globe . Out of which more than 36 million people are blind. India being the second largest population in the world. Contributes 30% of the overall blind population. Although there are enough campaigns being conducted to treat these people, it has been difficult to source all the treat these people. It has been difficult to source all the requirements.it is the era of artificial intelligence and it has gained immense traction due to large amount of data and ease of computation . Using artificial intelligence it is possible to make these people's life much easier. The goal is to provide a "secondary sight" until they have enough resources required to treat them. People with untreatable blindness can use this to make their everyday tasks much easier and simpler.

Physical movement is a challenge for visually impaired people. The conventional walking stick used by them is a very important int's range of detection and it is only used to detect the object which is near to the user. The disadvantage of the conventional cane. However, is it a failure to detect obstacles outside of its reach? That is the user has to tap the ground of the object to detect the obstacle.

The visually challenged people can avoid the object better if a better device is produced. Vibrating and sound warning when there is an object in the specific range of distance.

The development of a reliable object recognition system to perform reliable object recognition and tracking is still a challenge today. We can't underestimate the complexity of this problem as recognition is a process that can be easily done by the human eye . However for a computer to model and imitate the human eyes, there are many challenges involved.Some of the challenges faced in object recognition is the variation of viewpoints, illumination, shape and size of the interested objects.

#### CHAPTER-2 REQUIREMENTS ELICITATION & ANALYSIS

##### 2.1 Existing System

Moving object detection based on images obtained in real-time embedded systems where the computational and memory resources are scarce.

Temporal differencing uses the pixel wise differences between two or three consecutive images in an image sequence to extract moving regions and are highly adaptive to the dynamic scene changes.

It accomplishes this task by extracting and modelling each pixel value independently through a mixture of Gaussians of a particular allocation.

The pixels which are determined as belonging to the background category within the current frame are then described in the distribution.

##### 2.2 Proposed System

The proposed system aims to build a visual aid for visually impaired individuals with a voice feedback system in order to help them in their safe mobility. The system captures the real time environment and provides auditory feedback to the user through a headphone. Raspberry Pi 3 Model B+ was chosen as the functional device because of its low cost and high portability. It also offers a multiprocessing capability. To detect the obstacles and to categorise the objects, an ultrasonic sensor and Pi camera module is used. YOLO algorithm is used for detecting the objects and eSpeak, which is a compact open-source speech synthesiser gives the user an auditory feedback of the detected object.

##### 2.3 Feasibility Study

The purpose of this study is to perform the detection and pose estimation of articles for daily use. The main targets are furniture and home electrical appliances, such as a shelf, a refrigerator, or a chair. Conventionally, three-dimensional (3D) geometrical models have been used for such purposes. One drawback of that approach is that manual model creation is a burdensome task. Sim-ilar to other commonly used approaches relevant to ob-ject detection, we can image feature-based and shape feature-based approaches . However, these are problematic for our purpose because our targets gen-erally have no characteristic shape and almost no surface texture. Therefore, we develop a novel method suitable for foundation furniture and home electrical appliances. The pur-pose of this study is to perform detection and pose estima-tion of articles for daily-use.

The method proposed in this paper uses a model con-sisting of 3D surfaces and 3D edges (edge elements). Input Data for generating a model consists of pairs of colour and depth images captured from a 3D range image sensor. In the modelling process, 3D surfaces and edges are directly saved as model data. In contrast, object detection is performed by superimposing input data on the model. That is, surfaces and edges are first extracted from the input data in the same manner as that in the modelling process.Next, the surfaces are used to seek candidate positions of a target object. Then, edges are used to confirm each can-didate. Because the model representation is grounded in statistical approximation, it enables us to cope with sensor measurements that can have errors of several centimetres.

##### 2.4 System Requirements

###### 2.4.1.1 Functional Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.**  **no** | **REQUIREMENTS** | **REQUIREMENTS**  **NUMBERS** | **ESSENTIAL**  **OR**  **DESIRABLE** | **DESCRIPTION** |
| 1. | Camera(on/off) | RS1 | Essential | The camera will identify all the objects. |
| 2. | COCO | RS2 | Essential | It is used for object recognition. |
| 3. | Object Detection | RS3 | Essential | It will say what the object is and identify the object. |
| 4. | Distance calculation | RS4 | Essential | It says the distance between the object and person. |
| 5. | Voice Over | RS5 | Essential | It tells the object name. |

###### 2.4.1.2 Non-Functional Requirements

Hardware Requirements

* Camera: Using object detection methodology and depth images, the goal is to classify the object present in the image and estimate the distance from the camera, that is, what will be the real distance from the camera to the object when those were clicked. The distance between each of the points captured by the camera is then registered, and the image is saved
* Microphone: It captures audio by converting sound waves into an electrical signal, which may be a digital or analog signal. This process can be implemented by a computer or other digital audio devices. The first electronic microphone was based on a liquid mechanism, which used a diaphragm that was connected to a current- charged needle in a diluted sulfuric acid solution. It was not able to reproduce the intelligible speech.

**Software Requirements**

* TTS(Text To Speech):

Text-to-speech (TTS) is an assistive technology meant to read text aloud. The sound we listen to through TTS solutions is computer-generated, and we can control the reading speed by speeding it up or slowing it down.A majority of text to speech solutions work similarly. Users upload either a text file or type in the text they must convert to voice. After that, they kiselect from the voices available and see which sound is perfect for the voiceover.

Many TTS solutions rely on some variant of OCR technology. OCR helps us to recognize written and digital text and extract it from documents and images. For instance, if you click a picture of a street sign, the tool will read the words written on it.

* The You Only Look Once (YOLO) system is an object detection system designed for real-time processing:

YOLO is an abbreviation for the term ‘You Only Look Once’. This is an algorithm that detects and recognizes various objects in a picture (in real-time). Object detection in YOLO is done as a regression problem and provides the class probabilities of the detected images.

The YOLO algorithm employs convolutional neural networks (CNN) to detect objects in real-time. As the name suggests, the algorithm requires only a single forward propagation through a neural network to detect objects. This means that prediction in the entire image is done in a single algorithm run. The CNN is used to predict various class probabilities and bounding boxes simultaneously.The YOLO algorithm consists of various variants. Some of the common ones include tiny YOLO and YOLOv3

* OpenCV 4.2.0 :

OpenCV stands for Open Source Computer Vision. To put it simply, it is a library used for image processing. In fact, it is a huge open-source library used for computer vision applications, in areas powered by Artificial Intelligence or Machine Learning algorithms, and for completing tasks that need image processing. As a result, it assumes significance today in real-time operations in today’s systems. Using OpenCV, one can process images and videos to identify objects, faces, or even the handwriting of a human.

* Jupyter Notebook:

IPython notebook was developed by Fernando Perez as a web based front end to IPython kernel. As an effort to make an integrated interactive computing environment for multiple languages, the Notebook project was shifted under Project Jupyter providing front end for programming environments Juila and R in addition to Python.

A notebook document consists of rich text elements with HTML formatted text, figures, mathematical equations etc. The notebook is also an executable document consisting of code blocks in Python or other supporting languages.Jupyter notebook is a client-server application. The application starts the server on a local machine and opens the notebook interface in the web browser where it can be edited and run from. The notebook is saved as an ipynb file and can be exported as html, pdf and LaTex files.

* Anaconda: Anaconda is a [distribution](https://en.wikipedia.org/wiki/Software_distribution) of the [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) and [R](https://en.wikipedia.org/wiki/R_(programming_language)) [programming](https://en.wikipedia.org/wiki/Programming_language) [languages](https://en.wikipedia.org/wiki/Programming_language) for [scientific](https://en.wikipedia.org/wiki/Scientific_computing) [computing](https://en.wikipedia.org/wiki/Scientific_computing) ([data](https://en.wikipedia.org/wiki/Data_science) [science](https://en.wikipedia.org/wiki/Data_science), [machine](https://en.wikipedia.org/wiki/Machine_learning) [learning](https://en.wikipedia.org/wiki/Machine_learning) applications, large-scale [data processing](https://en.wikipedia.org/wiki/Data_processing), [predictive](https://en.wikipedia.org/wiki/Predictive_analytics) [analytics](https://en.wikipedia.org/wiki/Predictive_analytics), etc.), that aims to simplify [package](https://en.wikipedia.org/wiki/Package_management) [management](https://en.wikipedia.org/wiki/Package_management) and [deployment](https://en.wikipedia.org/wiki/Deployment_environment). The distribution includes data-science packages suitable for [Windows](https://en.wikipedia.org/wiki/Microsoft_Windows), [Linux](https://en.wikipedia.org/wiki/Linux), and [macOS](https://en.wikipedia.org/wiki/MacOS). It is developed and maintained by Anaconda, Inc., which was founded by Peter Wang and [Travis](https://en.wikipedia.org/wiki/Travis_Oliphant) [Oliphant](https://en.wikipedia.org/wiki/Travis_Oliphant) in 2012.As an Anaconda, Inc. product, it is also known as Anaconda Distribution or Anaconda Individual Edition, while other products from the company are Anaconda Team Edition and Anaconda Enterprise Edition, both of which are not free.Package versions in Anaconda are managed by the package management system [*conda*](https://en.wikipedia.org/wiki/Conda_(package_manager)). This package manager was spun out as a separate [open-source](https://en.wikipedia.org/wiki/Open_source) package as it ended up being useful on its own and for things other than Python. There is also a small, [bootstrap](https://en.wikipedia.org/wiki/Bootstrapping) version of Anaconda called Miniconda, which includes only conda, Python, the packages they depend on, and a small number of other packages.
* Tensorflow:

TensorFlow is an open source machine learning framework for all developers. It is used for implementing machine learning and deep learning applications. To develop and research fascinating ideas on artificial intelligence, the Google team created TensorFlow. TensorFlow is designed in the Python programming language, hence it is considered an easy to understand framework.TensorFlow is well-documented and includes plenty of machine learning libraries. It offers a few important functionalities and methods for the same.TensorFlow is also called a “Google” product. It includes a variety of machine learning and deep learning algorithms. TensorFlow can train and run deep neural networks for handwritten digit classification, image recognition, word embedding and creation of various sequence models.

* Google Colab:

Google is quite aggressive in AI research. Over many years, Google developed an AI framework called TensorFlow and a development tool called Colaboratory. Today TensorFlow is open-sourced and since 2017, Google made Colaboratory free for public use. Colaboratory is now known as Google Colab or simply Colab.Another attractive feature that Google offers to the developers is the use of GPU. Colab supports GPU and it is totally free. The reasons for making it free for the public could be to make its software a standard in the academics for teaching machine learning and data science. It may also have a long term perspective of building a customer base for Google Cloud APIs which are sold on a per-use basis.

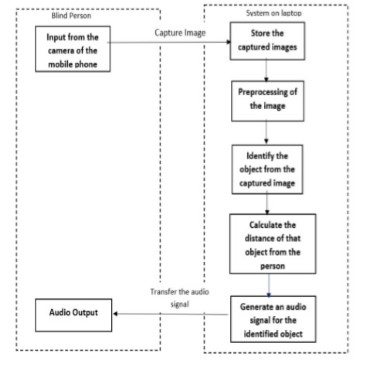
Irrespective of the reasons, the introduction of Colab has eased the learning and development of machine learning applications.

**CHAPTER-3**

#### SYSTEM DESIGN

System Design is a method for approaching the creation of a system. System design is the process of creating the architecture, components, modules, and interfaces for a system so that it meets the needs of the end user. System design could be defined as the application of system theory to product development.There is some overlap with the disciplines of system analysis, system architecture and system engineering.System modelling is the interdisciplinary study of the use of mobile to conceptualise and construct systems in business and IT development.This critical phase provides the understanding and procedural details required for implementing the system recommended in the feasibility study.The design step generates a data design, an architectural design, and a procedural design.The data design transforms the information domain model created during analysis.

##### Block Diagram



###### 3.1 Object Orientation Analysis and Design

* In the system analysis or object-oriented analysis phase of software development, the system requirements are determined, the classes are identified and the relationships among classes are identified.
* The three analysis techniques that are used in conjunction with each other for object-oriented analysis are object modelling, dynamic modelling, and functional modelling.
* Object modelling develops the static structure of the software system in terms of objects. It identifies the objects, the classes into which the objects can be grouped into and the relationships between the objects. It also identifies the main attributes and operations that characterise each class.

###### 1.Blind device can capture images via stereo camera

Stereo camera enables us to get 3D images because it consists of two cameras put in specific positions to act like the human eye. We don’t want to get a 2D image because we need to calculate the width between the person and the object in the next stages.

###### 2.Compute the distance between the person and each object

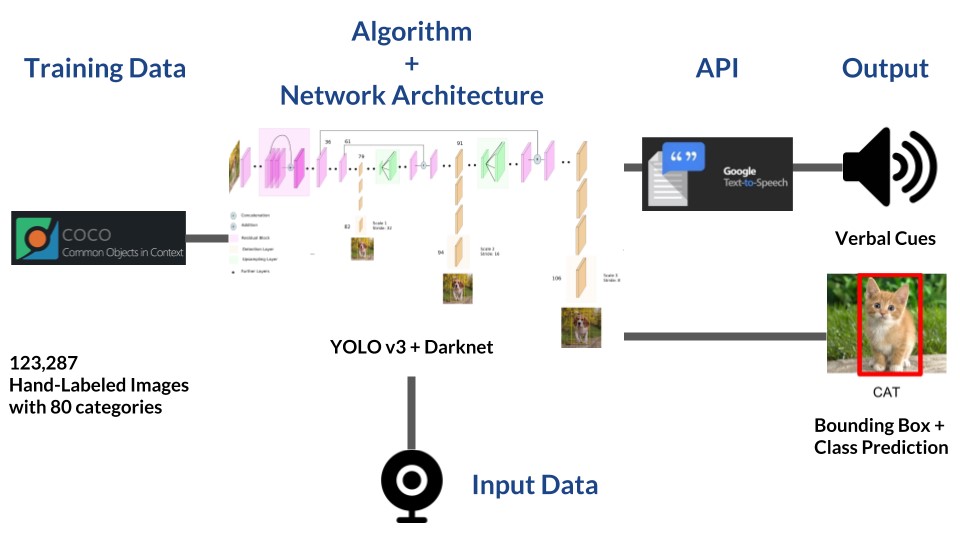
According to the 3D images captured from the camera we can compute the distance using the depth map in which each point will be as its real point in the real world.

###### 3.Convert the information to voice to be understood by blind people via headphone

After recognizing each object, objects will be expressed with text then using Speech Synthesis system will convert the text to sound using Text-To-Speech (TTS) technique and the person can hear it using the headphone connected to his ears.

4.Notify them if they are very close to an object

We can use a little loud voice when the person comes too close to an object or we can use alert buzzers.



**1.Training Data:** The model is trained with the Common Objects In Context (COCO) dataset.

**2.Model:** The model here is the You Only Look Once (YOLO) algorithm that runs through a variation of an extremely complex Convolutional Neural Network architecture called the Darknet. Even though we are using a more enhanced and complex YOLO v3 model, I will explain the original YOLO algorithm. Also, the python cv2 package has a method to setup

Darknet from our configurations in the yolov3.cfg file.

**3.Input Data:** We will be using our webcam to feed images at 30 frames-per-second to this trained model and we can set it to only process every other frame to speed things up.

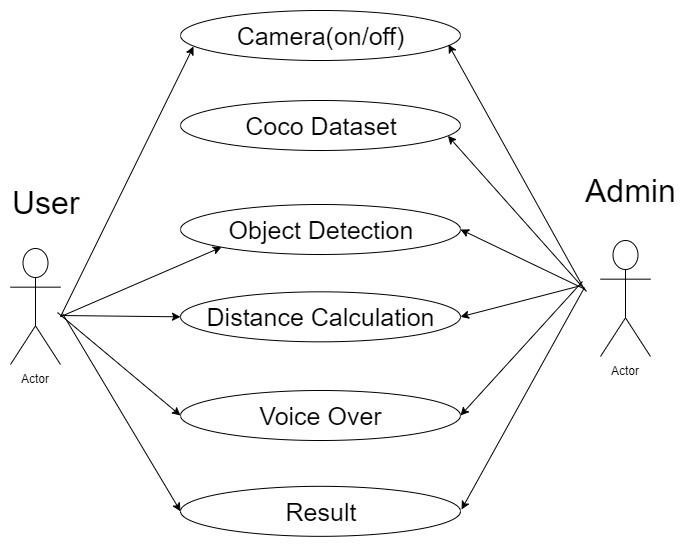
**4.API:** The class prediction of the objects detected in every frame will be a string e.g. “cat”. We will also obtain the coordinates of the objects in the image and append the position “top”/“mid”/“bottom” & “left”/“centre”/“right” to the class prediction “cat”. We can then send the text description to the Google Text-to-Speech API using the gTTS package.

**5.Output:** We will also obtain the coordinates of the bounding box of every object detected in our frames, overlay the boxes on the objects detected and return the stream of frames as a video playback.

3.1.1 Use Case diagram

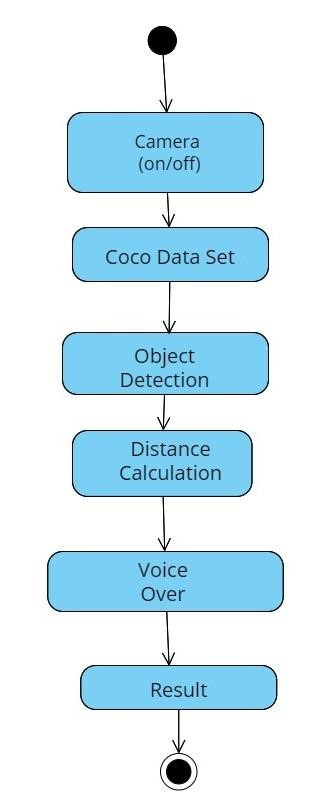
A use case diagram is a Unified Modelling Language (UML) is a type of behavioural diagram defined by and created from Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted. A use case is a set of scenarios that describes an interaction between a user and a system. A use case diagram displays the relationship among actors and use cases. The two main components of a use case diagram are use cases and actors. The purpose of a use case diagram is to capture the dynamic aspect of a system. But this definition is too generic to describe the purpose. Use case diagrams are used to gather the requirements are mostly design requirements. So, when a system is analysed to gather its -functionalities use cases are prepared and actors are identified.So, in brief, the purpose of use case diagrams:

1. Used to gather requirements of a system.
2. Used to get an outside view of a system.
3. Identify external and internal factors influencing the system.
4. Show the interacting among the requirements are actors.



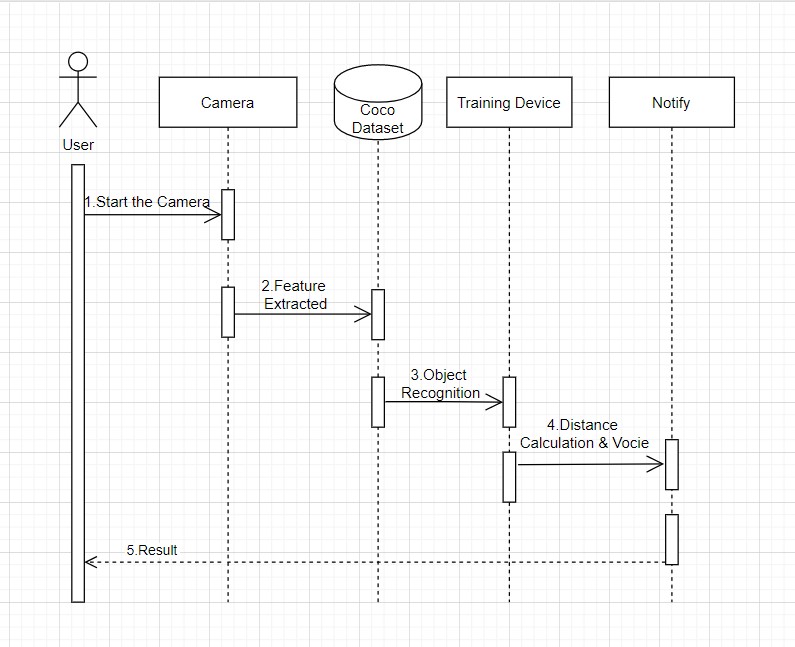
###### State Chart Diagram

A State chart diagram describes a state machine. State machine can be defined as a machine which defines different states of an object and these states are controlled by external or internal events. As the State chart diagram defines the states, it is used to model the lifetime of an object. State chart diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered. The most important purpose of a state chart diagram is to model the lifetime of an object from creation to termination



###### Sequence diagram

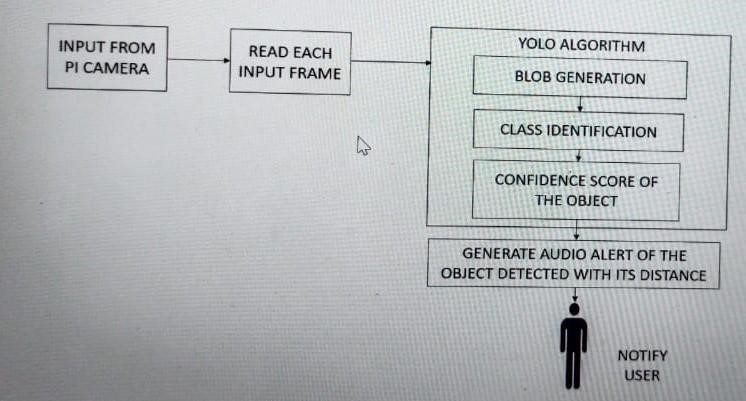
Sequence diagrams are probably the most important UML diagrams among not only the computer science community but also design - level models for business application development. Lately they have become popular in depicting business processes, because of their visually self- explanatory nature. As the name suggests, sequence diagrams describe the sequence of messages and interactions that happen between actors and objects. Actors or objects can be active when needed or when another object wants to communicate with them. All communication is represented in a chronological manner.A sequence diagram is a type of interaction diagram because it describes how and in what order a group of objects works together. These diagrams are used by software developers and business professionals to understand requirements for a new system or document an existing process. Sequence diagrams are sometimes known as event diagrams or event scenarios. The sequence diagram shows the interaction between two lifelines (parallel vertical lines) as a time ordered sequence of events.



#### CHAPTER-4

##### IMPLEMENTATION DETAILS

The system is implemented in the camera, headphone and power supply. Each part of the system was evaluated individually and the overall system is tested after assembling all the parts. The developed system is evaluated both for obstacle and object detection. The system is tested in real time to evaluate the performance of the system.



The Pi camera will capture a real-time video stream, then the video is split into frames and each input frame is read to get its width and height. Object detection is done using the YOLO Object Detection Algorithm in this system. To get the correct prediction the OpenCV function blobFromImage() is used to get the blob of the input frame, then it is sent to the YOLO pre-trained model. For each detection from each output layer, we will get the class label and confidence score. Then ignore the object whose confidence score is less than 0.6, and apply non-max suppression to determine the final detection. The class probability and confidence score together will help in determining the object accurately.

We have taken data and calculated the distance of these data. Standard deviation and variance are two closely associated measures of deviation. The variance is the measure of how much each value varies from the mean. The value of the variance represents the greater data range in the overall system. From the table data it can be observed that the lower values of standard deviation and variance are achieved when the hindrances are very near to the users. These values are increased with the increase of obstacles’ distance.

The audio feedback is generated to inform the user about his or her distance from

the obstacle, and the object type, which will help the visually impaired to move safely in their environment without depending on others.

**4.1 Software Environment**

###### 4.2 Software Technologies

A conda environment is a directory that contains a specific collection of conda packages that you have installed. A virtual environment is a tool that helps to keep dependencies required by different projects separate by creating isolated spaces for them that contain per-project dependencies for them. Users can create virtual environments using one of several tools such as Pipenv or Poetry, or a conda virtual environment. Pipenv and Poetry are based around Python's built-in venv library, whereas conda has its own notion of virtual environments that is lower-level (Python itself is a dependency provided in conda environments).

**Python** : It is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasises readability and therefore reduces the cost of program maintenance.

**NumPy** : is a Python library used for working with arrays. It also has functions for working in the domain of linear algebra, fourier transform, and matrices.

**Matplotlib** : is designed to be as usable , with the ability to use Python and the advantage of being free and open-source. Each pyplot function makes some change to a figure: e.g., creates a figure, creates a plotting area in a figure, plots some lines in a plotting area, decorates the plot with labels, etc.

**TensorFlow** : is a Python library for fast numerical computing created and released by Google. It is a foundation library that can be used to create machine Learning models directly or by using wrapper libraries that simplify the process built on top of TensorFlow.

4.1.2 Dataset Explanation

The dataset must be an image dataset containing object/objects as well as the distance between all the objects seen in the image, which should be registered and will serve as ground reality. To keep track of all these documents, an excel file with the image name, an entity present in that image, and the distance corresponding to that object should be kept.

Sample data set(fig(a)):

|  |  |  |  |
| --- | --- | --- | --- |
| **S.NO** | **IMAGE NAME** | **OBJECT** | **DISTANCE** |
| 1 | door.jpg | Door | 99 |
| 2 | ac.jpg | AC | 162 |
| 3 | window.jpg | Window | 76.5 |
| 4 | chair.jpg | Chair | 152 |
| 5 | cupboard.jpg | Cupboard | 120.5 |
| 6 | fridge.jpg | Fridge | 196 |
| 7 | sofa.jpg | Sofa | 110.5 |
| 8 | table.jpg | Table | 186 |
| 9 | tv.jpg | TV | 74.5 |

The names of the columns in the preceding table are :

1. image name, which comprises the names of the images.
2. Objects in the image, which is a list of the objects that were caught in the image.
3. Distance refers to the true distance between a certain object and the camera as measured at the time of the object's capture in the camera.

The dataset was developed by grabbing two or three objects and placing them at different distances from the camera, while still attempting to maintain relative distances between the objects so that it is clear which object is closer to the camera and which is farther away. The distance between each of the points captured by the camera is then registered, and the image is saved.

Various steps were taken when constructing the dataset including

1. Attempting to collect at most three data points from a single image.
2. Attempting to preserve the greatest possible relative distance between the objects so that the depth image would easily discern the collected different pixel values for different objects.

The distance between different objects was observed, and the names of images and their depth were supplied numerically in order to ensure one-to-one correspondence.

1. The height of objects was also retained in order to include it as one of the input features.
2. The backdrop was maintained as a white wall, and all objects were kept on the ground.

(v)The camera configuration was not considered,

(vi) all photographs were taken with the camera in a spot on the floor.

Evaluation matrix

If the expected values are greater than zero, this matrix is often used in many research papers to calculate the value. Its value ranges from 0 to 1, and model performance is determined by how close it is to zero. If rmsle is close to zero, model performance is better; if rmsle is close to one, model performance is worse. It is given by



Mean of region of interest all bounding box coordinates and object height as input:

Region of interest is the bounding box region which was obtained using YOLOv4 . To use this section of the image, it was first cropped, and then the mean of that cropped image was computed, yielding the mean value of r,g,b, which was then used as input features. So, for the first experiment, we took the mean of the pixel values produced from the depth map after adding bounding box coordinates to it. Apart from that, the entity height and width, as well as the bounding box height, are considered input features. As it turns out, there is a link between bounding box width (w) and bounding box height (h) . Which is used in the triangular similarity system. As a result, our input features are r,g,b,w,h, and entity height. The output function is the object’s true distance from the camera. The data set looks like as shown below(fig(b)).

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**S.NO**](http://s.no/) | X | Y | W | H | IMAGE | DISTAN  CE | r | g | b |
| **1** | 36 | 92 | 33 | 61 | data/IMG\_20210104  \_230447291.jpg | 99 | 120.5166  42 | 53.99751  6 | 107.3661  2 |
| **2** | 183 | 88 | 93 | 60 | data/IMG\_20210104  \_230447291.jpg | 162 | 134.3639  78 | 52.65465  9 | 172.5802  87 |
| **3** | 47 | 75 | 44 | 86 | data/IMG\_20210104  \_230800476.jpg | 76.5 | 130.7336  15 | 27.19186 | 72.43340  4 |
| **4** | 192 | 80 | 104 | 62 | data/IMG\_20210104  \_230800476.jpg | 152 | 141.7090  57 | 42.41826  9 | 162.7931  14 |
| **5** | 83 | 95 | 25 | 48 | data/IMG\_20210104  \_231127922.jpg | 120.5 | 133.5066  67 | 32.98 | 102.135 |
| **6** | 153 | 91 | 66 | 45 | data/IMG\_20210104  \_231127922.jpg | 196 | 93.11313  1 | 116.6013  47 | 217.9690  24 |
| **7** | 53 | 86 | 30 | 56 | data/IMG\_20210104  \_231314528.jpg | 110.5 | 127.4339  29 | 38.80059  5 | 99.775 |
| **8** | 134 | 87 | 72 | 48 | data/IMG\_20210104  \_231314528.jpg | 186 | 104.7245  37 | 93.15191 | 211.9956  6 |
| **9** | 16 | 65 | 43 | 90 | data/IMG\_20210104  \_231330048.jpg | 74.5 | 123.4413  44 | 41.06072  4 | 70.63746  8 |
| **10** | 154 | 77 | 88 | 57 | data/IMG\_20210104  \_231330048.jpg | 150 | 151.0574  16 | 21.68441 | 123.1200  16 |
| **11** | 4 | 73 | 57 | 107 | data/IMG\_20210104  \_231346592.jpg | 59.5 | 146.1939  66 | 28.80078  7 | 133.4940  15 |
| **12** | 186 | 91 | 104 | 64 | data/IMG\_20210104  \_231346592.jpg | 135 | 145.4224  76 | 35.48287  3 | 157.3460  04 |
| **13** | 137 | 77 | 69 | 48 | data/IMG\_20210104  \_231401065.jpg | 188.1 | 83.32518  1 | 127.9966  79 | 227.5428  74 |
| **14** | 58 | 79 | 27 | 54 | data/IMG\_20210104  \_231401065.jpg | 112.6 | 150.6117  97 | 10.97256  5 | 80.85253  8 |
| **15** | 141 | 75 | 74 | 53 | data/IMG\_20210104  \_231416685.jpg | 175.3 | 90.5436 | 121.1137  17 | 219.5237  12 |

fig(b)

**CHAPTER-5**

#### TESTING

We usually perform system testing to find errors resulting from the interaction between the sub-system and system components,Software must be tested to detect and rectify all possible errors once the source code is generated before delivering it to the customers.For finding errors,series of test cases must be developed which ultimately uncover all the possibly existing errors.Different software techniques provide systematic guidance for designing test that

* Exercise the internal logic of the software components.
* Exercise the input and output domains of a program to uncover errors in program function,behaviour and performance.

We test the software using two methods:

##### White Box testing

White-box testing is the detailed investigation of internal logic and structure of the code. White-box testing is also called glass testing or open-box testing. In order to perform white-box testing on an application, a tester needs to know the internal workings of the code.

##### Black Box testing

The technique of testing without having any knowledge of the interior workings of the application is called black-box testing. The tester is oblivious to the system architecture and does not have access to the source code. Typically, while performing a black-box test, a tester will interact with the system's user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon.

Third Party App provides ease and freedom in the field of app development. It brings efficiency and also helps in fast delivery of the output. Third Party App allows you to divide your work in parts and helps you to focus on the core part of the app or any system. This strategy helps in the development of good and quality software. We can pass on the Features of the Third Party App to the system.

1) At first, we are capturing real time images from the rear camera[4] of the mobile handset of blind people and a connection is established between mobile phone and system in laptop and then those images are sent from the mobile phone to laptop.

2)This connection is done by a Third party app which is installed in the mobile phone. All the real time images which get captured by the rear camera of the mobile phone are first transferred to the Third party app in the mobile phone and then those images are sent to the laptop where they are processed for some further conclusions.

1. The system in the laptop will test it using its APIs and SSD ALGORITHM and it detects the confidence accuracy of the image which it is testing. We reached 98% accuracy for certain classes like books, cups, and remote.
2. After testing the images we are generating an output on the laptop based system and its prediction is being translated into voice with voice modules and sent to the blind person with the help of wireless audio support tools

##### CONCLUSION

AI based intelligent blind stick design is proposed in this work. It helps blind people in choosing the right direction of movement and helps in navigation. It directs the blind person to choose a path that has no obstacle in their way. Three sensors installed in front, right and left side of the stick detects any near obstacles and selects the path that has no hindrance somewhere near comparatively. This information is given to the blind person through a speech feature which successfully informs the blind person to follow an obstacle free path.This AI based blind stick uses the features of ultrasonic sensors and makes it artificially intelligent by adding the decision-making feature in it to select the most suitable path intelligently. The speech feature is clear enough to give an accurate name of the image. There was no delay or inaccuracy in voice recommendations. Hence it did path selection and decision making for the blind friends successfully and directed them at each point of their walk to move either right, left or go straight; based on results of distance from obstacles in all three directions. The most interesting observation was that the stick proves to be affordable for them as well.

##### REFERENCES

1. Srinivas K.S., Sahithya K., Tejaswi G.L., Gopal K.H. and Karthik B.P., A new method forrecognition and obstacle detection for visually challenged using smart glasses powered with Raspberry Pi., International Journal of Engineering Applied Sciences and Technology, Vol. 5, Issue 1, ISSN No. 2455-2143, Pages 408-412, May 2020.
2. Hegde P., Devathraj N., Sushma S.K. and Aishwarya P., Smart Glasses for VisuallyDisabled Person, International Journal of Research in Engineering and Science (IJRES), Volume 9 Issue 7, ISSN: 2320-9356, PP. 62-68, July 2021.
3. Aditya Raj, "Model for Object Detection using Computer Vision and Machine Learning forDecision Making," International Journal of Computer Applications, 2019.

**APPENDIX**

##### CODING

**CAMERA:**

import tensorflow as tf import tensorflow\_hub as hub

# For downloading the image. import matplotlib.pyplot as plt import tempfile from six.moves.urllib.request import urlopen from six import BytesIO

# For drawing onto the image.

import numpy as np from PIL import Image from PIL import ImageColor from PIL import ImageDraw from PIL import ImageFont from PIL import ImageOps

# For measuring the inference time. import time

# Print Tensorflow version print(tf.\_\_version\_\_)

# Check available GPU devices. print("The following GPU devices are available: %s" % tf.test.gpu\_device\_name()) def display\_image(image):

fig = plt.figure(figsize=(20, 15)) plt.grid(False) plt.imshow(image)

def download\_and\_resize\_image(url, new\_width=256, new\_height=256, display=False):

\_, filename = tempfile.mkstemp(suffix=".jpg") response = urlopen(url) image\_data = response.read() image\_data = BytesIO(image\_data) pil\_image = Image.open(image\_data) pil\_image = ImageOps.fit(pil\_image, (new\_width, new\_height), Image.ANTIALIAS) pil\_image\_rgb = pil\_image.convert("RGB") pil\_image\_rgb.save(filename, format="JPEG", quality=90) print("Image downloaded to %s." % filename) if display:

display\_image(pil\_image)

return filename

def draw\_bounding\_box\_on\_image(image,

ymin, xmin, ymax, xmax,

color, font, thickness=4, display\_str\_list=()): """Adds a bounding box to an image.""" draw = ImageDraw.Draw(image) im\_width, im\_height = image.size

(left, right, top, bottom) = (xmin \* im\_width, xmax \* im\_width, ymin \* im\_height, ymax \* im\_height)

draw.line([(left, top), (left, bottom), (right, bottom), (right, top),

(left, top)],

width=thickness, fill=color)

# If the total height of the display strings added to the top of the bounding # box exceeds the top of the image, stack the strings below the bounding box # instead of above. display\_str\_heights = [font.getsize(ds)[1] for ds in display\_str\_list] # Each display\_str has a top and bottom margin of 0.05x. total\_display\_str\_height = (1 + 2 \* 0.05) \* sum(display\_str\_heights)

if top > total\_display\_str\_height:

text\_bottom = top

else:

text\_bottom = top + total\_display\_str\_height

# Reverse list and print from bottom to top. for display\_str in display\_str\_list[::-1]: text\_width, text\_height = font.getsize(display\_str) margin = np.ceil(0.05 \* text\_height) draw.rectangle([(left, text\_bottom - text\_height - 2 \* margin),

(left + text\_width, text\_bottom)], fill=color) draw.text((left + margin, text\_bottom - text\_height - margin), display\_str, fill="black", font=font)

text\_bottom -= text\_height - 2 \* margin

def draw\_boxes(image, boxes, class\_names, scores, max\_boxes=10, min\_score=0.1): """Overlay labeled boxes on an image with formatted scores and label names.""" colors = list(ImageColor.colormap.values())

try:

font =

ImageFont.truetype("/usr/share/fonts/truetype/liberation/LiberationSansNarrow-Regular.ttf",

25)

except IOError:

print("Font not found, using default font.") font = ImageFont.load\_default()

for i in range(min(boxes.shape[0], max\_boxes)): if scores[i] >= min\_score:

ymin, xmin, ymax, xmax = tuple(boxes[i]) display\_str = "{}: {}%".format(class\_names[i].decode("ascii"), int(100 \* scores[i]))

color = colors[hash(class\_names[i]) % len(colors)] image\_pil = Image.fromarray(np.uint8(image)).convert("RGB") draw\_bounding\_box\_on\_image( image\_pil, ymin, xmin, ymax, xmax,

color, font, display\_str\_list=[display\_str])

np.copyto(image, np.array(image\_pil)) return image

image\_url = "https://upload.wikimedia.org/wikipedia/commons/6/60/Naxos\_Taverna.jpg"

#@param

downloaded\_image\_path = download\_and\_resize\_image(image\_url, 1280, 856, True) module\_handle =

"https://tfhub.dev/google/faster\_rcnn/openimages\_v4/inception\_resnet\_v2/1"

detector = hub.load(module\_handle).signatures['default'] def load\_img(path):

img = tf.io.read\_file(path) img = tf.image.decode\_jpeg(img, channels=3) return img

detector = hub.load(module\_handle).signatures['default'] def load\_img(path):

img = tf.io.read\_file(path) img = tf.image.decode\_jpeg(img, channels=3) return img

def run\_detector(detector, path): img = load\_img(path)

converted\_img = tf.image.convert\_image\_dtype(img, tf.float32)[tf.newaxis, ...] start\_time = time.time() result = detector(converted\_img) end\_time = time.time() result = {key:value.numpy() for key,value in result.items()}

print("Found %d objects." % len(result["detection\_scores"])) print("Inference time: ", end\_time-start\_time)

image\_with\_boxes = draw\_boxes( img.numpy(), result["detection\_boxes"], result["detection\_class\_entities"], result["detection\_scores"])

display\_image(image\_with\_boxes)

run\_detector(detector, downloaded\_image\_path)

image\_urls = [

"data:image/jpeg;base64,/A//2Q==",

"", [https://upload.wikimedia.org/wikipedia/commons/thumb/0/0d/Biblioteca\_Maim%C3%B3nid es%2C\_Campus\_Universitario\_de\_Rabanales\_007.jpg/1024px-Biblioteca\_Maim%C3%B3ni des%2C\_Campus\_Universitario\_de\_Rabanales\_007.jpg](https://upload.wikimedia.org/wikipedia/commons/thumb/0/0d/Biblioteca_Maim%C3%B3nides%2C_Campus_Universitario_de_Rabanales_007.jpg/1024px-Biblioteca_Maim%C3%B3nides%2C_Campus_Universitario_de_Rabanales_007.jpg)

]

def detect\_img(image\_url): start\_time = time.time() image\_path = download\_and\_resize\_image(image\_url, 640, 480) run\_detector(detector, image\_path) end\_time = time.time() print("Inference time:",end\_time-start\_time)

detect\_img(image\_urls[1])

**DISTANCE:**

import cv2 import numpy as np model = 'C:/Users/reshm/Downloads/yolov3.weights' config = 'C:/Users/reshm/Downloads/yolov3.cfg' min\_confidence=0.14 net = cv2.dnn.readNetFromDarknet(config,model)

with open("lib/coco.names", "r") as f:

classes = [line.strip() for line in f.readlines()]

layer\_names = net.getLayerNames()

output\_layers = [layer\_names[i - 1] for i in net.getUnconnectedOutLayers()] COLORS = np.random.uniform(0, 255, size=(len(layer\_names), 3)) import math def dist\_calculator(startX, startY, endX, endY, box\_width, box\_height, img\_w, img\_h):

x\_3, y\_3 = startX, endY - (box\_height / 7) # top left of the triangle

# assumption: camera is raised above the ground so considering 90% of the height of the image height x\_1, y\_1 = img\_w / 2, 0.9 \* img\_h # bottom of the triangle x\_2, y\_2 = endX, endY - (box\_height / 7) # top right of the triangle

# find the angle between bottom and right point angle\_x1\_x2 = math.degrees(math.atan2(x\_1 - x\_2, y\_1 - y\_2))

# find the angle between bottom and left point angle\_x1\_x3 = math.degrees(math.atan2(x\_1 - x\_3, y\_1 - y\_3))

angle\_right = 90 + angle\_x1\_x2 angle\_left = 90 - angle\_x1\_x3

# total angle of view for the bench from bottom center point of the image. total\_angle = angle\_right + angle\_left

# Bench length assumed to be 2 metres in millimetres. This value can be automated, based on the type of bench used. bench\_length = 2000 distance = (bench\_length \* (1 / total\_angle) \* 57) / 1000

print(total\_angle) print(distance) return total\_angle, distance

image = cv2.imread('images/objectname.jpg') height, width, ch = image.shape resize\_img = cv2.resize(image, (225, 225)) blob = cv2.dnn.blobFromImage(resize\_img, 1.0 / 255.0, (416, 416), True, crop=False)

net.setInput(blob) predictions = net.forward() probability\_index = 5 for i in range(predictions.shape[0]):

prob\_arr = predictions[i][probability\_index:] class\_index = prob\_arr.argmax(axis=0) confidence = prob\_arr[class\_index] if confidence > min\_confidence:

x\_center = predictions[i][0] \* width y\_center = predictions[i][1] \* height width\_box = predictions[i][2] \* width height\_box = predictions[i][3] \* height

x1 = int(x\_center - width\_box \* 0.5) # Start X coordinate y1 = int(y\_center - height\_box \* 0.5) # Start Y coordinate x2 = int(x\_center + width\_box \* 0.5) # End X coordinate y2 = int(y\_center + height\_box \* 0.5) # End y coordinate

cv2.rectangle(image, (x1, y1), (x2, y2), (0, 255, 120), 2)

if class\_index == 13:

# 13 is the index of Bench in the LABELS list for prediction roi\_corners = [[0.0 \* width\_box, 1.0 \* height], # left, down [x1 + (width\_box / 5), y2 - (height\_box / 7)], # left, up

[x1 + 4 \* (width\_box / 5), y2 - (height\_box / 7)], # right, up

[1.0 \* width, 1.0 \* height]] # right, down

image = cv2.circle(image, (int(width), int(height)), 3, (0, 0, 255), -1) triangle\_pts = [[width / 2, 0.9 \* height], # left, down / bottom point of the traingle

[x1, y2 - (height\_box / 7)], # left, up

[x2, y2 - (height\_box / 7)], # right, up

[width / 2, 0.9 \* height]] # right, down / / bottom point of the traingle src = np.float32(triangle\_pts) pts = np.array(src, np.int32) pts = pts.reshape((-1, 1, 2))

# distance calculation

\_, distance = dist\_calculator(x1, y1, x2, y2, width\_box, height\_box, width, height)

cv2.putText(image, "Distance: {} mm".format(round(distance, 2)), (x1, y1 + 13), cv2.FONT\_HERSHEY\_SIMPLEX, 0.5,

(0, 255, 120), 2)

cv2.imshow("detection and distance raw", image) cv2.waitKey(0) cv2.destroyAllWindows()

**VOICE:**

from gtts import gTTS import os

mytext = 'text you want to convert' language = 'en' myobj = gTTS(text=mytext, lang=language, slow=False) myobj.save("filename.mp3")

os.system("mpg321 filename.mp3")

**Output screens:**

