**Visual Assistant For Visually Impaired**

A Project-II Report

Submitted in partial fulfillment of requirement of the

Degree of

**BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE & ENGINEERING**

BY

**CHIRAYU YADAV - EN18CS301072**

**HANY JAGWANI - EN18CS301093**

**ISHIKA RAJAWAT - EN18CS301110**

Under the Guidance of

**Dr. Ratnesh Litoriya**



**Department of Computer Science & Engineering**

**Faculty of Engineering**

**MEDI-CAPS UNIVERSITY, INDORE- 453331**

**JAN-MAY 2022**

**Report Approval**

The project work **“Visual Assistant For Visually Impaired”** is hereby approved as a creditable study of an engineering subject carried out and presented in a manner satisfactory to warrant its acceptance as prerequisite for the Degree for which it has been submitted.

It is to be understood that by this approval the undersigned do not endorse or approved any statement made, opinion expressed, or conclusion drawn there in; but approve the “Project Report” only for the purpose for which it has been submitted.

**Internal Examiner**

Name: Dr. Ratnesh Litoriya

Designation: Assistant Professor

Affiliation: Medi-Caps University, Indore

**External Examiner**

Name:

Designation:

Affiliation:

**Declaration**

I hereby declare that the project entitled **“Visual Assistant For Visually Impaired”** submittedin partial fulfillment for the award of the degree of Bachelor of Technology in ‘Computer Science & Engineering’ completed under the supervision of **Dr. Ratnesh Litoriya, Assistant Professor Computer Science & Engineering,** Faculty of Engineering, Medi-Caps University Indore is an authentic work.

Further, I declare that the content of this Project work, in full or in parts, have neither been taken from any other source nor have been submitted to any other Institute or University for the award of any degree or diploma.

**Ishika Rajawat**

**10th Nov, 2021**

**Certificate**

I, **Ratnesh Litoriya** certify that the project entitled **“Visual Assistant For Visually Impaired”** submittedin partial fulfillment for the award of the degree of Bachelor of Technology by **Ishika Rajawat** istherecordcarried out by her under my guidance and that the work has not formed the basis of award of any other degree elsewhere.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Dr. Ratnesh Litoriya

Computer Science & Engineering

Medi-Caps University, Indore

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

<Name of External Guide (If any)>

<Name of the Department>

Name of the Organization

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Dr. Pramod S. Nair

Head of the Department

Computer Science & Engineering

Medi-Caps University, Indore

**Acknowledgements**

I would like to express my deepest gratitude to Honorable Chancellor, **Shri R C Mittal,** who has provided me with every facility to successfully carry out this project, and my profound indebtedness to **Prof. (Dr.) D. K. Patnaik,** Vice Chancellor, Medi-Caps University, whose unfailing support and enthusiasm has always boosted up my morale. I also thank **Prof. (Dr.) Suresh Jain,** Dean, Faculty of Engineering, Medi-Caps University, for giving me a chance to work on this project. I would also like to thank my Head of the Department **Dr. Pramod S. Nair** for his continuous encouragement for betterment of the project.

I express my heartfelt gratitude to my Internal Guide Dr. Ratnesh Litoriya without whose continuous help and support, this project would ever have reached to the completion.

**Ishika Rajawat**

B.Tech. IV Year

Department of Computer Science & Engineering

Faculty of Engineering

Medi-Caps University, Indore

**Executive Summary**

Artificial Intelligence has been touted as the next big thing that is capable of altering the current landscape of the technological Domain. Through the use of Artificial Intelligence and Machine Learning, pioneering works have been undertaken in the area of Visual and Object Detection. In this paper, we undertake the analysis of a Visual Assistant Application for guiding visually-impaired individuals. With recent breakthroughs in computer vision and supervised learning models, the problem at hand has been reduced significantly to the point where new models are easier to build and implement over the already existing models. Now, different Object Detection models exist that provide object tracking and detection with great accuracy. These techniques have been well used in automating the detection task in different areas. Some of the newfound detection approaches such as YOLO(You Only Look Once), SSD(Single Shot Detector) , and R-CNNs have proved to be consistent and pretty accurate in Real-Time Object Detection. We showed how visual assistance for visually-impaired individuals that uses an IoT-based Cane fitted with a camera is achievable. We have demonstrated our idea using a mobile application and have successfully achieved a great response time within 4 seconds. Applications are developed to provide these individuals with visual aid. We aim to develop a visual assistant for visually-impaired individuals that uses an IoT-based Cane fitted with a camera. The important aspect is to train the model on a large number of samples so that we can obtain a fine-tuned and accurate Object-detection model. Many Object Detection algorithms have been proposed with wide-scale applicability. Choosing one such model that pertinently solves the problem at hand is a major determiner in obtaining good accuracy. We have provided reviews of various object detection techniques that work with the scenic view of generic images. A wide-scale comparison among the various object detectors has encouraged us to use **YOLOv3**, an incrementally modified form of **YOLO**, as the object detector. The accuracy and mAP score of YOLO is above par with most of the contemporary detectors, and for another reason YOLO is simpler in implementation, allowing simple and robust construction of an object detector. We have discussed the Project architecture, i.e., a client-server model along with the various necessary components. The modular approach has enabled us to achieve a great average response time of 5s. We are going to have a brief look at these techniques in order to find a good base model for implementing our ‘Visual Assistant’.

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **TITLE** | **Page No.** |
|  |  |  |
|  | Report Approval | ii |
|  | Declaration | iii |
|  | Certificate | iv |
|  | Acknowledgment | v |
|  | Executive Summary | vi |
|  | Table of Contents | vii |
|  | List of figures | ix |
|  | List of tables | xii |
|  | Abbreviations | xiii |
| Chapter 1 | Introduction | 1-4 |
|  | 1.1 Introduction | 2 |
|  | 1.2 Literature Review | 2 |
|  | 1.3 Objectives | 3 |
|  | 1.4 Scope | 4 |
|  | 1.5 Chapter Scheme | 4 |
| Chapter 2 | System Requirement Specification | 5-9 |
|  | 2.1 Existing System | 6 |
|  | 2.2 Proposed System | 6 |
|  | 2.2 Procedures Adopted | 7 |
|  | 2.3 System Feasibility | 8 |
|  | 2.4 Hardware Specification | 8 |
|  | 2.5 Software Specification | 8 |
|  | 2.6 Functional Requirement | 8 |
|  | 2.7 Non-Functional Requirement | 9 |
| Chapter 3 | System Analysis & Design | 10-18 |
|  | 3.1 ER Diagram | - |
|  | 3.2 Use case Diagram | 13 |
|  | 3.3 Activity Diagram | 15 |
|  | 3.4 Sequence Diagram | 19 |
| Chapter 4 | Implementation | 21-41 |
|  | 4.1 Procedural Description | 22 |
|  | 4.2 Module Description | 23 |
|  | 4.3 Database Implementation (with screenshots) | 24 |
|  | 4.4 User Interface Implementation (with screenshots) | 28 |
|  | 4.5 Technical Discussions | 36 |
| Chapter 5 | Testing | 42-54 |
| Chapter 7 | Results and Discussions | 55-57 |
| Chapter 8 | Summary and Conclusions | 58-59 |
| Chapter 9 | Future scope | 60-61 |
|  | References | 62 |

**List of Figures**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **TITLE** | **Page No.** |
|  | Use Case Diagram | 11 |
|  | Activity Diagram | 12 |
|  | Sequence Diagram | 13 |
|  | Landing Page of the App | 15 |
|  | Clicking the picture | 17 |
|  | Terminal Window | 18 |
|  | After entering the url | 23 |
|  | Clicking the picture | 23 |
|  | Terminal Window | 24 |

**Abbreviations**

|  |  |
| --- | --- |
| YOLO | You Only Look Once |
| COCO | Common Objects In Context |
| SDLC | Software Development Life Cycle |
| STLC | Software Testing Life Cycle |

**Chapter-1**

**Introduction**

* 1. Introduction
* One of the primary goals of the image-based learning is to understand and differentiate among various scenic description of common objects of interest. This task can be subdivided into a number of subtasks - bounding box creation, object localization, attribute determination and relationship establishment. The images of various objects can be broadly classified into Iconic and Scenic Views. The Iconic approach assumes the presence of a single object with clear boundaries and separation edges .But, the iconic view is too simple to accommodate real-life situations ,where in images are seldom iconic ,but involves a large number of intertwined objects in a small space. In order to detect objects of interest, image segmentation and context mining should be applied to filter out points of interest. Most of the existing systems perform well under these iconic views ,but achieve lower accuracy in scenic instances. Objects in scenic environments are cluttered, overlapping and without good contrast. Various techniques of segmentation are applied to extract useful information from these scenic views. When building new models, it is of paramount importance to select a learning domain most suitable to the need and implementation. In order to train these models, the dataset employed plays a crucial part in establishing good results. One of the major challenges is to find pertinent training images and samples to accommodate a more modular and robust learning. Various pioneering works have been done in collecting these image samples under one roof into a dataset. Some of these datasets contain millions of samples and training instances, spanning thousands of objects. Currently, some of the more popular Datasets include - Google’s ImageNet, Microsoft COCO Dataset, PASCAL VOC, SUN, etc.
* Globally, the number of people of all ages visually impaired is estimated to be 285 million, of whom 39 million are blind. Although technology has grown leaps and bounds, the accessibility, especially that of the internet for differently-abled people is still far-fetched. These individuals have trouble running rudimentary daily errands due to low visual capacity. We aim to utilize the hearing capacity of these individuals to provide them with Real-Time scenic descriptions using object detection and scene labeling.
  1. **Literature Review**

1. You Only Look Once : Unified, Real-Time Object Detection (Joseph Redmon, Santosh Divalla, Ross Girshick,) [1]

This paper presents YOLO as a new approach to object detection. They have framed object detection as a regression problem to spatially separated bounding boxes and associated class probabilities. Fast YOLO, processes an astounding 155 frames per second while still achieving double the mAP of other real-time detectors. Compared to state-of-the-art detection systems, YOLO makes more localization errors but is less likely to predict false positives on background. Finally, YOLO learns very general representations of objects. It outperforms other detection methods, including DPM and R-CNN, when generalizing from natural images to other domains like artwork.

1. Incremental Few-Shot Object Detection (Juan-Manuel, Perez-Rua, Xiatian Zhu, Timothy Hospedales, Tao Xiang) [2]

Despite the success of deep convolutional neural networks (CNNs) in object detection, for almost all the current models a lengthy process of numerous iterations in a batch is used to train them .Here in the current setting, all the target classes with a large number of training samples interpreted with training data are familiar and for training purposes all the training images are used.The potential for these methods to accommodate new classes online and grow is restricted severely by the interpretation cost and training complexity.

1. Microsoft COCO : Common Objects in Context(Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnik, Piotr Dollar [3]

This paper has been written as a guide to the novel COCO Dataset created for Object detection and classification. It mainly focuses on the non-iconic or scenic views of images, pointing out the difficulties encountered when detecting scenic views of images, and pointing out the difficulties encountered when detecting scenic views. It outlines image segmentation, bounding-box generation, heatmap, and per-pixel color location. The focus is on 2D and 3D image localization and per-pixel semantic segmentation. The paper outlines the need for a large and rich annotated images with a large number of instances per sample of objects

1. YOLOv3: An Incremental Improvement(Joseph Redmon, Ali Farhadi[4])

This paper presents some design changes to YOLO, which makes it a little bigger but more accurate and fast. YOLOv3 is approximately as accurate as an SSD but three times faster. YOLOv3 clusters the dimensions of ground truth labels to generate anchor boxes for predicting the bounding boxes, where each bounding box has 4 coordinates, tx, ty, tw, th. Each box predicts the classes which may be present using multilabel classification. During training, it uses the binary cross-entropy loss for making class predictions.

1. Object Detection in 20 years: A survey(Zhengxia Zou, Zhenwei Shi, Member, IEEE, Yuhong Guo, and Jieping Ye [5])

This paper reviews more than 400 papers on object detection spanning from the 1990s to 2109, focussing on the technical advancements made in this area. This paper emphasizes several topics which include several early-stage detectors, datasets for detection, metrics, possible speed-up techniques whithin be used, and the race state-of-the-art detection methods. This paper also sheds light on some important applications of detection, such as text detection, face detection, pedestrian detection, etc, and make an analysis of the development made and challenges faced in recent time.

Various aspects make this paper different from all the reviews done on object detection. In-depth research on the key technologies and state-of-the-art object detection systems has been done here, while the previous reviews lacked fundamental analysis to give readers a complete understanding of complex techniques. Most of the previous reviews were focused on a short period of time or on some specific detection task without considering the development history.

1. Application for the Visually Impaired People With Voice Assistant ( Abhijeet Mohanta, Shah Yash Jitendra, Khandelwal Niketa Dinesh, Wable Saurabh Suhas and Aruna K. Gupta [6])

This paper aims to study the implementation of a voice assistant for visually impaired individuals. Various modules have been discussed in the paper which can be implemented. After capturing a photo from a smartphone, the user can easily read menu cards of restaurants, the room number of the hotel and can also find their belongings. The voice control feedback mechanism is also used in the app through which the user can perform various tasks with the help of the voice assistant. Cloud Computing, Image processing, and machine learning is used to develop the application

1. Blind Path Obstacle Detector using Smartphone Camera and Line Laser Emitter ( Rimon Saffoury, Peter Blank, Julian Sessner, Benjamin H. Groh, Christine F. Martindale, Eva Dorschky, Joerg Franke, and Bjoern M. Eskofier[7])

Visually impaired people find themselves wandering inside unusual challenging areas. Many smart systems have intended to help blind people in these difficult, often dangerous, situations. However, some of them are not free, hard to find, or simply too expensive. In this paper, a low-cost wear system for blind people was is designed to allow them to discover and discover obstacles in their place. The proposed program consists of two main components hardware components, a laser pointer ($ 12) and an android smartphone, which makes our system cheaper and more accessible. I Conflict avoidance algorithm uses image processing to measure distances to objects in the surrounding area.

1. H. Virtual Assitant for the Visually Impaired ( Vinayak Lyer, Kshitij Shah, Sahil Sheth, Kailas Devadkar)

An accessible web interface for visually impaired individuals is presented in this paper. In order to maximize ease of use and provide users with a hassle-free experience, the virtual assistant is operating system independent and doesn't rely on keyboard input from the user. Communication with and customization of the system are possible using speech-to-text and text-to-voice interfaces. This presentation provides an overview of the system design and implementation methodology for the three modules currently in use. To answer user queries quickly accurately, Wikipedia uses a BERT model built from the SQuAD dataset. It was found that 80.88% of the words exactly matched. Anyone with visual impairments can easily access any website using the virtual assistant. With this program, you don't have to memorize complex keyboard commands or use screen readers. As a tool for interacting with the websites, the assistant is not only very convenient, but also quite effective. According to the results, the software was successfully run on the three most popular sites: Google, Gmail, and Wikipedia. It was run separately on each of these sites. The software is a stepping stone towards Web 3.0 where all functions can be controlled through voice commands.

* 1. Objective

1. Develop a model that identifies general objects from an image.
2. Develop a mobile application that captures an image and relays the scenic description from the app via the earpiece.
3. Build an intermediate link between the app and model that helps in generating queries and reading out responses.
   1. Scope
4. At present, the application will detect objects and pronounce them only in the English language. It can be expanded and made available in most of the daily used languages thus people from all parts of the world can access the web without any issue.
5. A web application can also be developed that would perform the same task without having to download the mobile application.
   1. **Chapter Schema**

**Chapter 2: System Requirement Specification**

This chapter include the information about system requirements like functional and non-functional requirement and specification of the software like software and hardware specifications.

**Chapter 3: System Analysis & Design**

This chapter include all the diagrams related to the system for better understanding of the system.

**Chapter 4: Implementation**

This chapter include the implementation phase of the system include proposed methods for the system and screenshots of interfaces and database.

**Chapter 5: Testing**

This chapter include the testing phase of the system. Several test cases are included in this chapter.

**Chapter 6: Result and Discussion**

This chapter include results drawn by the system.

**Chapter 7: Summary & conclusion**

This chapter includes summary about the project.

**Chapter 8: Future Scope**

This chapter include the future scope possible in the system.

**Chapter-2**

**System Requirement Specification**

* 1. Existing System

**VisionCap**

VisionCap is based on the Microsoft's Computer Vision API. I've programmed the API with various custom images of entrances and exits (doors), walkways (Yellow dotted strips to assist blind people in navigating public spaces, mostly found in blind schools and government buildings, and basic day to day objects, some of which were already hardcoded in the API. VisionCap runs on Raspberry Pi board (SBC), along with an 8 MP camera which captures images in real time and narrates and describes the scene to the user. There is also a programmable button on the top which can be used to capture images, and also using basic smartphone features like making phone calls, narrating news, narrating headlines, reading SMS and emails, speech-calculator, etc. The device has a Bluetooth 4.0 module which connects it to the smartphone, for data connection for transmission of images to API, as well as using basic smartphone features and commands.

**VIRTUAL ASSISTANT FOR BLIND PEOPLE**

The proposed system acts like an intermediate level of virtual assistant. It performs more and more numbers of operations and do it easily for user. The main objective of this system is to reduce the user task and do it by itself for the user. In future, we can take our system to next level and make it advanced level of virtual personal assistant which can do almost all the operation which is done by the user.

**ViT Cane**

ViT Cane utilizes the recent Vision Transformer architecture to detect obstacles in a fast and accurate manner. The objects detected are then classified as obstacles and eventually relayed to the user through motors integrated at the handle of the cane. The experiment demonstrates the efficiency of the ViT Cane in real-world scenarios captured by the five courses.

**NELO, THE VISION ASSISTED BY AN AI**

The Nélo on-board visual recognition system combines tools, software and AI to allow the computer to analyze visual elements: obstacles, objects, texts, faces. Perception’s intelligent vision aid for the blind and visually impaired harnesses computer vision to deliver real-time audio information. A smart automated visual inspection tool, Nélo is a vigilant eye that films, records and interprets images in the street, shops, offices, homes, schools, green spaces

* 1. Proposed System

This project aims to develop a utility for visually-impaired individuals using an objection detection model and IoT. We propose to install camera on walking-cane of blind individual so that objects in front could be detected using Object detection model to detect the objects and the response will be pronounced to the individual from the mobile app via an earpiece. We aim to utilize the hearing capacity of these individuals to provide them with Real-Time scenic description using object detection and scene labelling. We aim to achieve a model that identifies general objects from an image and develop a mobile application that captures an image and relays the scenic description from the app via the earpiece. Build an intermediate link between the app and model that helps in generating queries and reading out responses.

2.3 Procedures Adopted

The Project Architecture will comprise various dependent components implemented as stand-alone modules. We will adopt a client-server architecture, wherein the Server is a remote entity running on a local machine. The Client Application will be implemented as a mobile application that is connected to the camera device through a wireless network either using Bluetooth, WiFi, or other wireless transmission protocols. The only requirement is sufficient bandwidth and low latency. The mobile application will send a request to the mirror site which, in turn, will forward it to the local server. The local server, running a YOLOv3 model, will detect objects within the input image and create a list of objects found. This list will finally get converted into a string and will be sent as a response to the mirror site, which, redirects the response to the client application. The client application using text-to-speech functionality will convert this string into audio that is fed into the earpiece of the visually-impaired individual. For simplicity, the entire image will be divided into 9 different zones, viz., Center, Top Left, Bottom Right. The model will also predict the zone of each object detected using the bounding-box location returned by the YOLO model.

2.4 System Feasibility

1. Technical Feasibility: It includes finding out technologies for the project, both hardware, and software. The mobile phone must have a working camera to capture the image and a working earpiece so that the user can listen to the instructions. A good internet connection is also a must on the mobile phone.
2. Operational Feasibility It is the ease and simplicity of operation of the proposed system. The system does not require any special skill set for users to operate it. In fact, it is designed to be used easily the visually impaired individual.

This project is technically feasible with no external hardware requirements. Also it is simple in operation and does not cost training or repairs. Overall feasibility study of the project reveals that the goals of the proposed system are achievable.

2.5 Hardware Specification

* 1. Memory(RAM) : Minimum 1GB: Recommended 4GB or more.
  2. Minimum 100MB space required.
  3. High-resolution camera, earpiece/speaker.

2.6 Software Specification

1. Python
2. Dart
3. Flutter
4. Flask
5. CV2
6. Ngrok

2.7 Functional Requirement

These are the requirements that the end-user specifically demands as basic facilities that the system should offer. All these functionalities need to be necessarily incorporated into the system as a part of the contract. These are represented or stated in the form of input to be given to the system, the operation performed and the output expected.

Req. 1:

* INPUT: Capture the image.
* OUTPUT: Objects detected by the system will be pronounced to the user via earpiece.

2.8 Non-functional requirements

These are basically the quality constraints that the system must satisfy according to the project contract. The priority or extent to which these factors are implemented varies from one project to another. They are also called non-behavioral requirements.

* **Portability and compatibility:** The software can only work on android.
* **Security:** No authorization required.
* **Performance:** The system should respond within 5 seconds.
* **Availability:** Our system can be accessed 24\*7 with a good internet connection.
* **Usability:** The system should have a user-friendly interface.

**Chapter-3**

**System Analysis & Design**

* 1. **Use-Case Diagram**

A use case diagram is used to represent the dynamic behavior of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models the tasks, services, and functions required by a system/subsystem of an application. It depicts the high-level functionality of a system and also tells how the user handles a system.

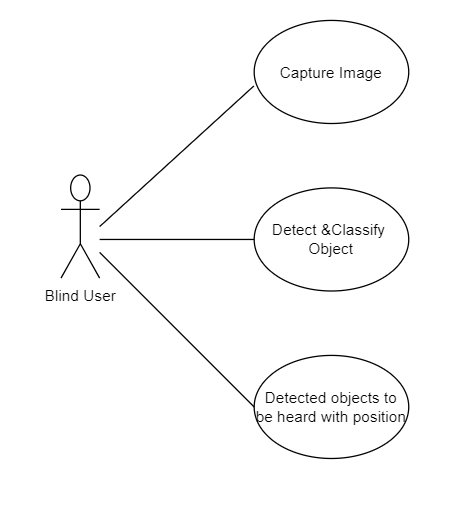
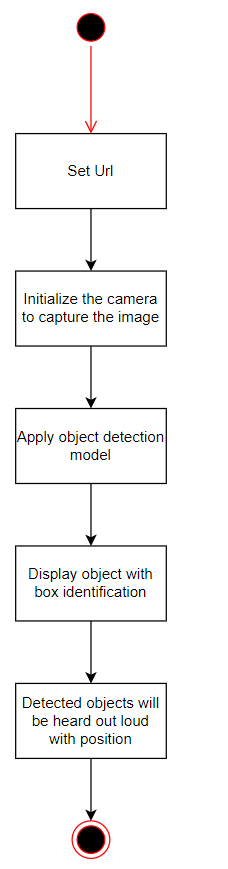


Figure 1: Use Case Diagram

* 1. **Activity Diagram**

An activity diagram is a behavioral diagram i.e., it depicts the behavior of a system. An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed. We can depict both sequential processing and concurrent processing of activities using an activity diagram.



*Figure 2: Activity Diagram*

* 1. **Sequence Diagram**

A sequence diagram is a Unified Modelling Language (UML) diagram that illustrates the sequence of messages between objects in an interaction. A sequence diagram consists of a group of objects that are represented by lifelines, and the messages that they exchange over time during the interaction.

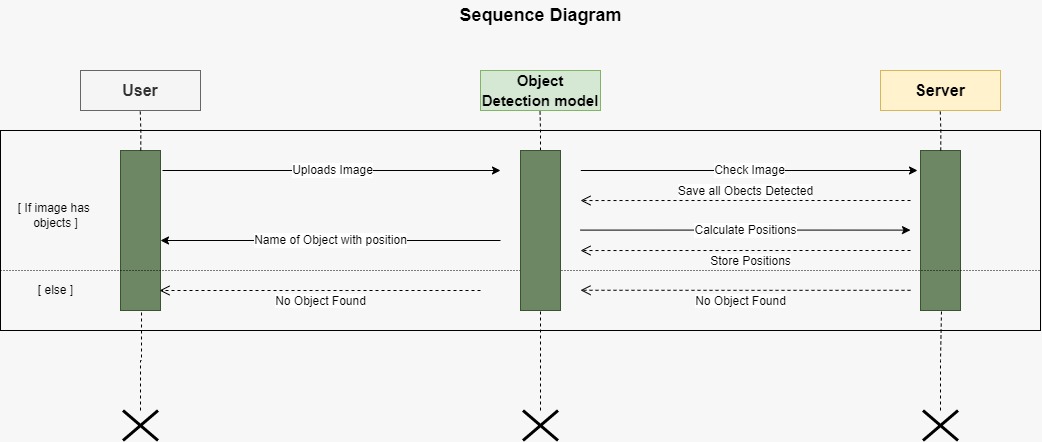


Figure 3: Sequence Diagram

**Chapter-4**

**Implementation**

1. **Procedural Description**

The suggested system is built using Python 3, OpenCV, and the flask framework of python. Using OpenCV's machine learning methods, we can educate the machine to discern between diverse user use cases and unauthorized offenders' unique undesired conduct, allowing us to take appropriate action based on the context. We are able to effectively use logic to execute the artificial intelligence idea at hand to recognize and classify the events that occur using image processing strategies and mathematical deductions. Additionally, the system is capable of acting in response to the current occurrence.

The primary goal of this system is to analyze collected image for object detection. As a result, the procedure begins by scanning captured image. This is depicted in Figure, which is also depicted in the block diagram with the entire sequence of actions.

The object detection framework is the most essential aspect of this research. This is due to the study's component that focuses on establishing an object's position from the input image. As a result, selecting the most appropriate object detection model is critical in order to prevent any issues with recognising objects.

As soon as the server is online and the user captures an image using the mobile application, its initial inclination is to scan the image for all the objects present in the frame. The recognised objects with their positions are then sent to the mobile application.

The string to data collected from the image using object detection model is sent to the Text to speech API which converts object’s names into audio and the system relay audio data through ear piece which helps the user to know the exact location and type of the object present in front of them.

1. **Module Description**

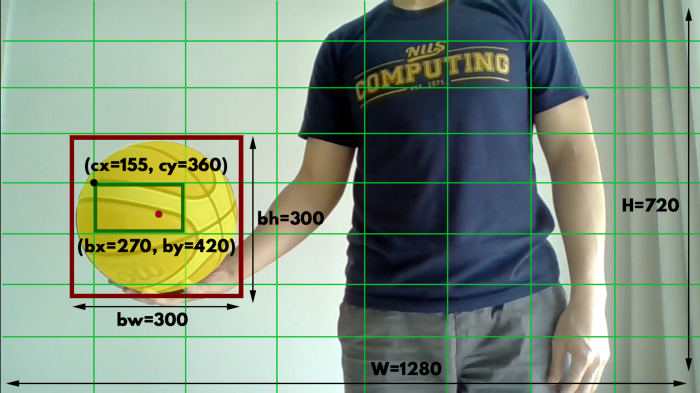
**Data:** The model is trained with the Common Objects in Context (COCO) dataset.

**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. The COCO has already been trained on YOLO v3 by others and we have already obtained the weights stored in a 200+mb file.

To understand weights, think of it as trying to find the Best Fit Line in Linear Regression. We need to find the right values of m and c in y=mx+c such that our line minimizes the error between all points. Now in our more complex prediction task, we have millions of Xs when we feed images into the complex network. These Xs will each have an m and these are the predicted weights stored in our yolov3.weights file. The **m**s have been constantly readjusted to minimize some loss function.

**Input Data:**We will be using our webcam to feed image to this trained model.

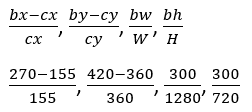
YOLO will automatically resize it to 416 x 234 and fit it into a popular standard-sized 416 x 416 network by padding the excess with 0s. YOLO divides each image into S x S cells each with a size of 32 x 32 (reduction factor=32). This creates 416/32 = 13 x 13 cells.



Using 8x8 cells for illustration. Dark-green box is the cell which contains the center of the object.

**There are 5 values in a bounding box — (bx, by, bw, bh, BC)**

If the center of an object (red dot) falls into a grid cell, only that grid cell (dark green cell) is responsible for detecting that object. Each bounding box has 5 values. The first 4 values **bx, by, bw, bh** represent the position of the box.

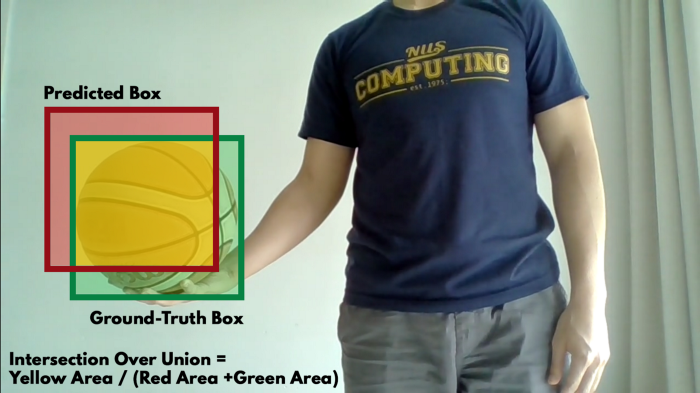


1) Normalized using the coordinates of the top-left corner of the cell which contains the object’s center. 2) using the dimensions of the entire image.

The 5th value is **BC**: the box confidence score.

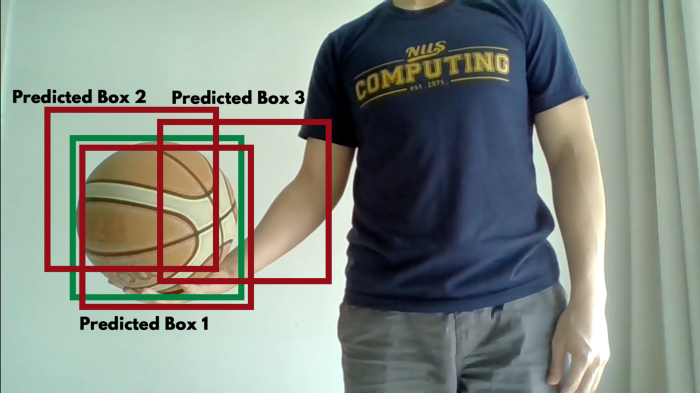
BC = Pr(Object existing in box) \* **IOU (**Intersection Over Union).

This measures how likely the box contains an object of *any class* and how accurate it is in predicting. BC=0 if no object exists in that box and we want BC=1 in predicting the ground truth.



Fairly high IOU

**There are B bounding boxes predicted in each grid cell**



YOLO v3 makes B=3 bounding boxes in each cell to predict that one object in the cell.

**There are also C conditional class probabilities in each grid cell**

There are 80 conditional class probabilities — Pr(Class i | Object) per cell when we use COCO. It is the probability that the predicted object is of Class i given that there is an object in the cell.

1 person 0.01  
2 bicycle 0.004  
.  
.  
33 sports ball 0.9  
.  
.  
80 toothbrush 0.02

In our example above, Class 33 has the highest probability and it will be used as our prediction of the object to be a sports ball.

**To sum up the above**

There are S x S cells and in each of these cells there are 2 things: 1) B bounding boxes each with 5 values (bx, by, bw, bh, BC), 2) C conditional class probabilities. The predictions are encoded as a S x S x (5 \* B + C) tensor.

**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”/“center”/“right” to the class prediction “cat”. We can then send the text description to the Google Text-to-Speech API using the **gTTS** package.

**Output:**We will also obtain the coordinates of the bounding box of every object detected in our image, overlay the boxes on the objects detected and return the string of objects detected. We will also schedule to get a voice feedback e.g. “bottom left cat” — meaning a cat was detected on the bottom-left of my camera view.

* 1. **User Interface Implementation**

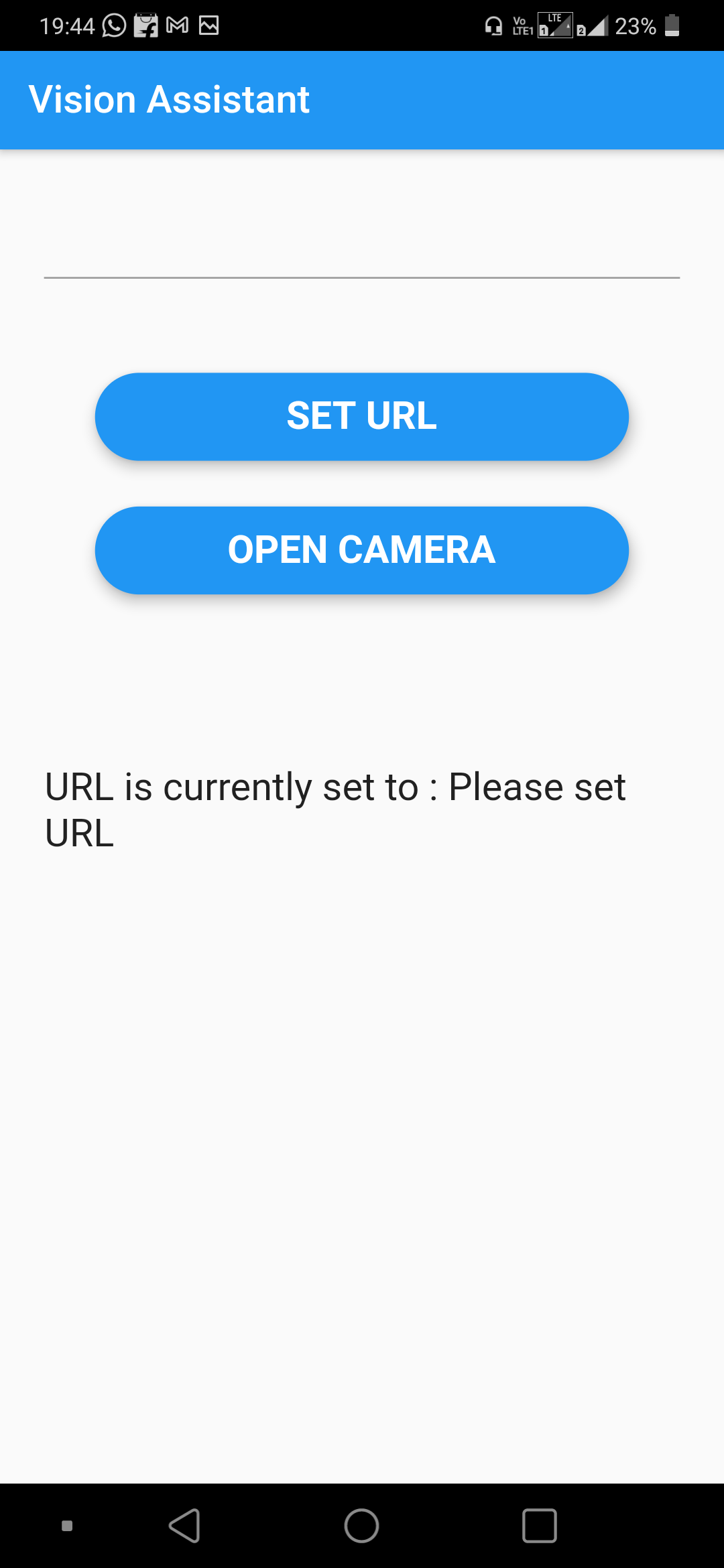
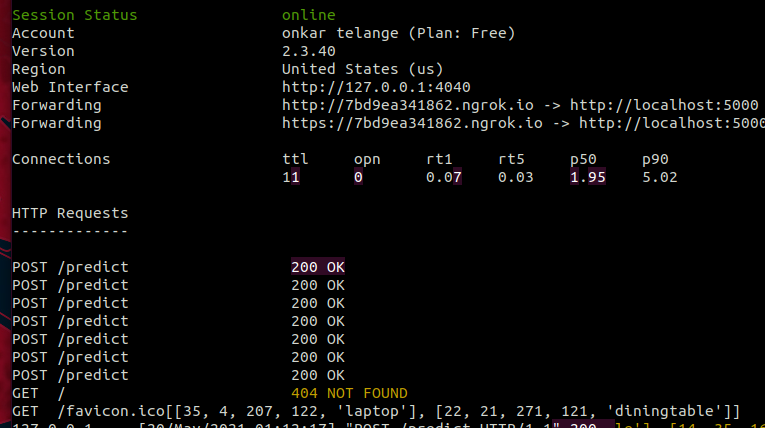
As you open the vision assistant app a landing page given below will open. After execution of project a url will be displayed on the terminal window. Enter the url and click on set url. After clicking on open camera click the picture.

 Figure 4: Landing Page of app

Figure 27: Add Faculty G

*Figure 5: Clicking the photo*

After clicking the photo of a laptop “laptop is in centre” will be heard to the user. The name of the item and the position of it will be heard out loud. Moreover, the position and name of the item will be displayed on terminal window as shown in the figure below.



*Figure 6: Terminal Window*

**Chapter-5**

**Testing**

**5.1 Unit Testing:**

Unit Testing is a type of software testing where individual units or components of a software are tested. The purpose is to validate that each unit of the software code performs as expected. Unit Testing is done during the development (coding phase) of an application by the developers. Unit Tests isolate a section of code and verify its correctness. A unit may be an individual function, method, procedure, module, or object.

In SDLC, STLC, V Model, Unit testing is first level of testing done before integration testing. Unit testing is a WhiteBox testing technique that is usually performed by the developer.

As mentioned earlier, unit tests are only run on individual components. The project's focus is on a visual assistant for visually impaired individual is limited in scope, so you only need to test a few components. Our features worked fine and passed the test. Individual components such as setting up the url, clicking photo, detecting objects and speaking the detected objects out aloud were tested and it was successful in unit test phase.

**5.2 Integration Testing:**

Integration testing is the second level of the software testing process comes after unit testing. In this testing, units or individual components of the software are tested in a group. The focus of the integration testing level is to expose defects at the time of interaction between integrated components or units.Unit Testing uses modules for testing purpose, and these modules are combined and tested in integration testing. The Software is developed with a number of software modules that are coded by different coders or programmers. The goal of integration testing is to check the correctness of communication among all the modules. Once all the components or modules are working independently, then we need to check the data flow between the dependent modules is known as **integration testing**.

Various units of the project were combined to produce the desired result. Our project passed the integration testing and we obtained the desired output at the end. All features were called together at different times and worked without interruption.

**5.3 Validation Testing:**

The process of evaluating software during the development process or at the end of the development process to determine whether it satisfies specified business requirements. Validation Testing ensures that the product actually meets the client's needs. It can also be defined as to demonstrate that the product fulfills its intended use when deployed on appropriate environment.

Our project satisfied all the business requirements. It checked if the software met the user needs. We evaluated the entire product including the code. No bug was reported during testing.

**5.4 White Box Testing:**

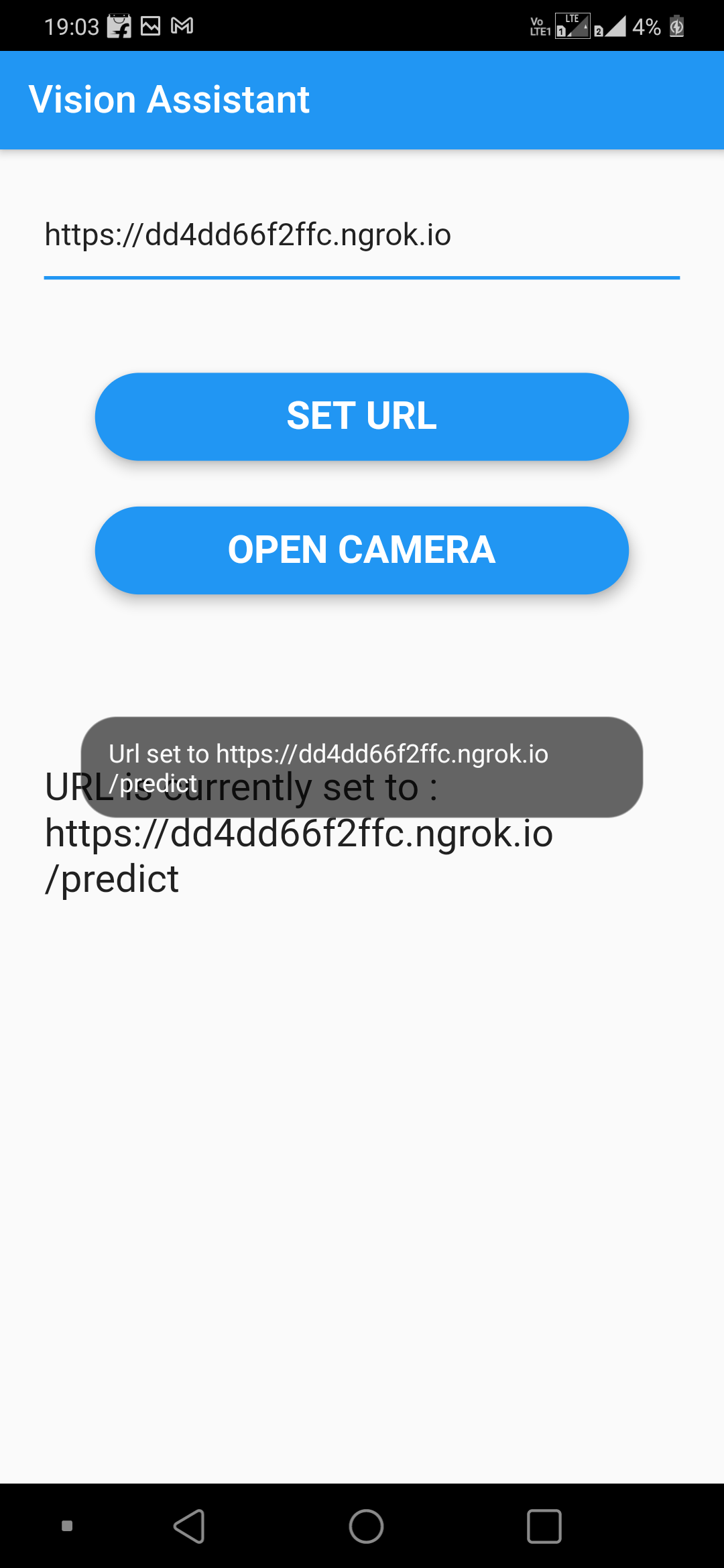
**White Box Testing** is software testing technique in which internal structure, design and coding of software are tested to verify flow of input-output and to improve design, usability and security. In white box testing, code is visible to testers so it is also called Clear box testing, Open box testing, Transparent box testing, Code-based testing and Glass box testing.

Our application performed excellently. As expected our application was able to detect the objects in the photo that was captured and furthermore, the detected objects and their position were heard out loud.

**Chapter-6**

**Result & Discussion**

* 1. **Result & Discussion**

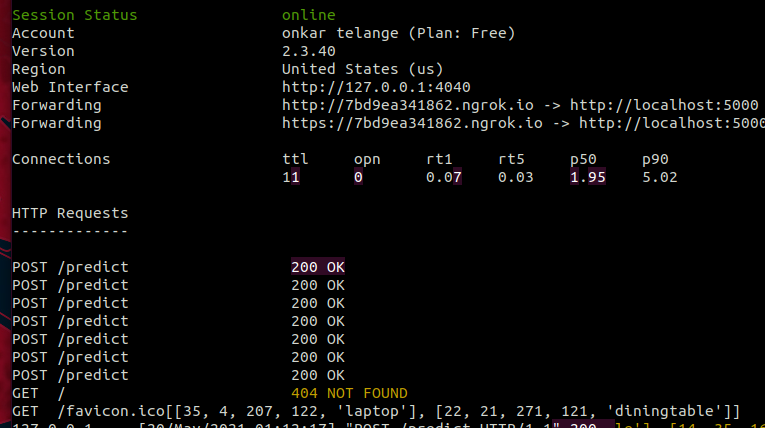


*Figure 7 : After entering url*

The url displayed on the terminal window is entered in the mobile application. After setting the url a picture is clicked. The name of the items will be heard aloud and will also be displayed on the terminal window along with its position. It can be seen in the figure below.



*Figure 8 : Clicking the Picture*



*Figure 9 : Terminal window after clicking the photo*

**Chapter-7**

**Summary & Conclusions**

**7.1 Summary & Conclusion**

Millions of visually-impaired individuals face a lot of difficulties in running daily errands that require certain visual capacity. Earlier, it was only possible to guide these individuals manually. But, now, with the advent of mobile technology, the Internet and Artificial Intelligence, Virtual Guide Applications are developed to provide these individuals with visual aid. We aim to develop a visual assistant for visually-impaired individuals that uses an IoT-based Cane fitted with a camera. The important aspect is to train the model on a large number of samples so that we can obtain a fine-tuned and accurate Object-detection model. Many Object Detection algorithms have been proposed with wide-scale applicability. Choosing one such model that pertinently solves the problem at hand is a major determiner in obtaining good accuracy. We have provided reviews of various object detection techniques that work with a scenic view of generic images. A wide-scale comparison among the various object detectors has encouraged us to use YOLOv3, an incrementally modified form of YOLO, as the object detector. The accuracy and mAP score of YOLO is above par with most of the contemporary detectors, and for another reason YOLO is simpler in implementation, allowing simple and robust construction of an object detector. We have discussed the Project architecture, i.e., a client-server model along with the various necessary components. The modular approach has enabled us to achieve a great average response time of 5s

**Chapter-8**

**Future Scope**

**8.1 Future Scope**

* At present, the application will detect objects and pronounce them only in the English language. It can be expanded and made available in most of the daily used languages thus people from all parts of the world can access the web without any issue.
* A web application can also be developed that would perform the same task without having to download the mobile application.

# REFERENCE

[1] 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.

[2] Perez-Rua, Juan-Manuel, et al. ”Incremental few-shot object detection.” Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020.

[3] Lin, Tsung-Yi, et al. ”Microsoft coco: Common objects in context.” European conference on computer vision. Springer, Cham, 2014.

[4]Redmon, Joseph, and Ali Farhadi. ”Yolov3: An incremental improvement.” arXiv preprint arXiv:1804.02767 (2018).

[5]Zou, Zhengxia, Zhenwei Shi, Yuhong Guo, and Jieping Ye. ”Object detection in 20 years: A survey.” arXiv preprint arXiv:1905.05055 (2019).

[6] “Application for the Visually Impaired People With Voice Assistant” Abhijeet Mohanta, Shah Yash Jitendra, Khandelwal Niketa Dinesh, Wable Saurabh Suhas and Aruna K. Gupta

[7] “Blind Path Obstacle Detector using Smartphone Camera and Line Laser Emitter”Rimon Saffoury, Peter Blank, Julian Sessner, Benjamin H. Groh, Christine F. Martindale, Eva Dorschky, Joerg Franke, and Bjoern M. Eskofier

[8]Visual Assistant for Visually-impaired individual using IoT

[9]Yolo Object Detection isual Assistant for visually