ASSISTIVE TOOL FOR VISUALLY IMPAIRED USING OBJECT DETECTION, LANE DETECTION, DEPTH ESTIMATION AND NAVIGATION TOOLS

Report Submitted in partial fulfillment of requirement for
the award of B.Tech in Computer Science Engineering

Submitted by:-

Harmeet Singh **(2019UCO1515)**
Rupesh Kumar **(2019UCO1547)**
Sumit Yadav **(2019UCO1548)**
Aditya Kumar **(2019UCO1557)**

Under the guidance of
Dr. Savita Yadav
(Assistant Professor)

Department of Computer Science and Engineering
NETAJI SUBHAS UNIVERSITY OF TECHNOLOGY
NEW DELHI-110078

May 2023

Department of Computer science & engineering
Netaji Subhas University of Technology
Dwarka, Delhi-110078, India

CERTIFICATE OF ORIGINALITY

We, **Hameet Singh** (2019UCO1515), **Rupesh Kumar** (2019UCO1547), **Sumit Yadav** (2019UCO1548), **Aditya Kumar** (2019UCO1557) of B.Tech Department of Computer science & engineering, hereby declare that the project titled "**Assistive tool for visually impaired using object detection, lane detection, depth estimation and navigation tools**" which is submitted by me/us to the **Department of Computer science & engineering, Netaji Subhas university of technology** (NSUT) Dwarka, New Delhi in partial fulfillment of the requirement for the award of the degree of bachelor of technology is original and not copied from the source without proper citation. The manuscript has been subjected to plagiarism checks by Turnitin software. This work has not previously formed the basis for the award of any degree.

Place: New Delhi, India

Date:

Hameet Singh	Rupesh Kumar	Sumit Yadav	Aditya Kumar
2019UCO1515	2019UCO1547	2019UCO1548	2019UCO1557

Department of Computer Science & Engineering
Netaji Subhas University of Technology
Dwarka, Delhi-110078, India

CERTIFICATE OF DECLARATION

This is to certify that the work embodied in project thesis titled, "**Assistive tool for visually impaired using object detection, lane detection, depth estimation and navigation tools**" by **Hameet Singh** (2019UCO1515), **Rupesh Kumar** (2019UCO1547), **Sumit Yadav** (2019UCO1548), **Aditya Kumar** (2019UCO1557) is the bonafide work of the group submitted to **Netaji Subhas University Of Technology** for consideration in 8th Semester B.Tech Project Evaluation.

The original research work was carried out by the team under my guidance and supervision in the academic year 2022-2023. This work has not been submitted for any other diploma or degree of any university. On the basis of declaration made by the group, we recommend the project report for evaluation

Place: New Delhi, India

Date:

Dr. Savita Yadav

(Assistant Professor)

Department of Computer Science and Engineering

Netaji Subhas University of Technology

ACKNOWLEDGEMENT

We would like to express our gratitude and appreciation to all those who make it possible to complete this project. Special thanks to our project supervisor **Dr Savita Yadav** whose help, stimulating suggestions and encouragement helped us in writing this report. We also sincerely thank our colleagues for the time spent proofreading and correcting our mistakes.

We wish to express heartfelt gratitude to our college, Netaji Subhas University Of Technology for giving us this opportunity for research and development.

Hameet Singh	Rupesh Kumar	Sumit Yadav	Aditya Kumar
2019UCO1515	2019UCO1547	2019UCO1548	2019UCO1557

ABSTRACT

This study focuses on the development and evaluation of a mobile application aimed at assisting visually impaired individuals in road navigation. The application is built using Flutter and utilizes advanced computer vision techniques, including object detection, lane detection, and depth estimation, to detect obstacles and suggest lane changes. Additionally, the integration of NLP and speech to text allows users to interact with the app verbally, improving its accessibility. The Google Maps API is also integrated to facilitate navigation. User testing was conducted to evaluate the app's usability, functionality, and overall effectiveness in aiding navigation. Results showed a positive response from users, with the app's ability to detect obstacles and provide useful navigation information being highlighted. However, areas for improvement were also identified, including the need for better voice recognition and more detailed navigation instructions. Our study represents a significant advancement in the development of technology to assist visually impaired individuals, with the potential to greatly enhance their quality of life.

Keywords:

Computer vision, NLP, Google Maps API, Visual Impairment, Mobile application, User testing.

INDEX

LIST OF FIGURES

CHAPTER 3

CHAPTER 4

LIST OF TABLES

CHAPTER 4

List Of Algorithm

CHAPTER 3

CHAPTER 1: INTRODUCTION

1.1. Motivation

The prevalence of visual impairment is increasing globally, with the World Health Organization estimating that around 2.2 billion people have some form of visual impairment. Road navigation can be a particularly challenging aspect of daily life for visually impaired individuals, with obstacles such as curbs, traffic lights, and other pedestrians posing potential hazards.

In recent years, there have been significant advancements in the development of technology to assist visually impaired individuals with road navigation. However, many of these technologies are expensive, complex, and require specialized training to use. Therefore, there is a need for cost-effective and accessible technology to aid visually impaired individuals in road navigation.

Our project aims to address this need by developing a mobile application that utilizes advanced computer vision techniques, NLP, and the Google Maps API to aid visually impaired individuals in road navigation. The application is built using Flutter, a cross-platform development framework, making it accessible to a wide range of users. Additionally, it utilizes object detection, lane detection, and depth estimation to detect obstacles and suggest lane changes, allowing for safer navigation. The integration of NLP and speech to text improves app accessibility, and the Google Maps API facilitates navigation.

Advantages of our project are numerous:

- First, it is cost-effective and accessible, with the use of Flutter making it compatible with a wide range of devices.

- Second, it is user-friendly, with the integration of NLP and speech to text making it easy for visually impaired individuals to interact with the app verbally.

- Third, it utilizes advanced computer vision techniques to detect obstacles and suggest lane changes, improving the safety of road navigation for visually impaired individuals.

- Fourth, the integration of the Google Maps API facilitates navigation, providing detailed and accurate directions.

Our project represents a significant advancement in the development of technology to assist visually impaired individuals with road navigation. By utilizing cost-effective and accessible technology, we hope to make road navigation safer and more independent for visually impaired individuals worldwide. Additionally, our project has the potential to be adapted for other applications, such as indoor navigation or obstacle detection in other contexts.

1.2. Key Challenges

The development of a mobile application aimed at assisting visually impaired individuals in road navigation using advanced computer vision techniques, NLP, and the Google Maps API presents several challenges. In this section, we will discuss some of the key challenges faced while making the project.

Firstly, one of the most significant challenges was the decision between Cloud computing and Edge computing. Both patterns presented their own pros and cons, we had to analyze both of them to come to a decision. The analysis included comparing the accuracy of models, response time, latency and ease of implementation.

Secondly, the subtask of depth estimation also presented some complexities. The task of depth estimation can be solved by applying basic trigonometry and using 2 cameras with a fixed distance between them. This approach cannot be used in our use case because the height of the camera also matters. The users using our application will be of different heights. Fortunately depth estimation can also be done using Monocular depth estimation which requires only a single image to produce the depth map. The depth map produced by this approach seemed to be relativistic rather than being absolute. We found a workaround to this problem as well, covered in Chapter 3.

Thirdly, ensuring that the app was accessible to visually impaired individuals was a challenge. Our application uses computer vision, the input and output of computer visions tasks are images. We had to find a way to convert the input and output to some other sensory mode. We decided to go with Auditory sense. The input can be given via the user through voice commands and the application will give users sound cues.

Fourthly, integrating the Google Maps API presented several challenges. The API provides a wealth of information that can be used to facilitate navigation. However, this information needed to be presented in a clear and concise manner, while also taking into account the needs of visually impaired individuals. Additionally, the app needed to be

able to provide accurate and up-to-date directions, even in areas with poor network connectivity.

Finally, ensuring that the app was stable, usable and efficient and was a challenge. This was mainly pertaining to implementation of the application and making key decisions which would affect the user experience. We explored various options like - Abstract classes or Interfaces or Singleton classes approaches to implement individual services. Another key decision was to decide where the Orchestration layer should reside - on Flutter app or on Backend API. We have explained the decision making process in a detailed manner in Chapter 3.

In conclusion, the development of a mobile application aimed at assisting visually impaired individuals in road navigation using advanced computer vision techniques, NLP, and the Google Maps API presents several challenges. By addressing the challenges, we were able to develop a cost-effective and accessible mobile application. It can greatly improve the quality of life for visually challenged individuals worldwide.

1.3. Problem Addressed in the thesis

The problem addressed by this thesis is the lack of effective solutions for visually impaired individuals to navigate roads and streets. Visually impaired individuals face significant challenges when it comes to independent travel, and this has a significant impact on their daily lives.

Traditional aids such as canes and guide dogs can only provide limited assistance, and they rely heavily on the user's ability to perceive their surroundings. In addition, these aids do not provide any information about potential hazards such as incoming objects, road obstacles, or other pedestrians.

Multiple machine learning models solve individual problems. If we can make them work together, then the problem can be solved more efficiently.

1.4. Approach to the problem and organization of thesis

Our solution to the problem of road navigation for visually impaired individuals is a mobile application that uses advanced computer vision techniques, NLP, and the Google Maps API. The application provides real-time feedback to the user on the surrounding environment, helping them to navigate roads and sidewalks safely and independently.

The application uses object detection to identify obstacles in the user's path, lane detection to suggest lane changes, and depth estimation to determine the distance of objects. The NLP and speech to text software allows users to interact with the application using their voice, making it more accessible for visually impaired individuals.

The Google Maps API provides navigation functionality, allowing the user to enter a destination and receive turn-by-turn directions. The application also integrates with the user's calendar, allowing them to schedule appointments and receive reminders for upcoming events.

The organization of the thesis begins with an introduction to the problem of road navigation for visually impaired individuals, followed by a literature review of previous research in this area. We then discuss the methodology used in the development of the mobile application, including the computer vision techniques, NLP, and the Google Maps API.

Finally, we present the results of the prototype and ML models. We also discuss the limitations of the application and areas for future development.

Overall, the thesis presents a comprehensive solution to the problem of road navigation for visually impaired individuals, using advanced computer vision techniques, NLP, and the Google Maps API.

CHAPTER 2: LITERATURE REVIEW

Before beginning our project, we reviewed some past material. The following literature was reviewed and examined to help us determine where we should be heading.

2.1. IoT based route assistance for visually challenged:

1. This research paper proposes a system that leverages the IoT technology to assist visually challenged individuals in navigating their environment. The proposed system consists of a network of IoT devices, including cameras, sensors, and GPS modules, which are used to detect obstacles, track the user's location, and provide audio feedback.
2. The system has been created to offer support in diverse settings, including indoor and outdoor environments. It uses various types of IoT devices that are specifically tailored for each scenario. For indoor environments, the system uses cameras and sensors to detect obstacles and provide audio feedback to the user. For outdoor environments, the system utilizes GPS modules to track the user's location and provide audio feedback on the user's surroundings.
3. The authors conducted several experiments to evaluate the system's performance, including testing its accuracy in obstacle detection and its ability to provide reliable audio feedback. The results of the experiments showed that the proposed system achieved high accuracy in obstacle detection and provided reliable audio feedback to users.
4. One of the key advantages of the proposed system is its scalability, as it can be easily deployed in various environments and can be customized to meet the specific needs of individual users. The system also has the potential to be integrated with other assistive technologies to provide a comprehensive solution for visually challenged individuals.
5. Overall, the research paper presents a novel IoT-based route assistance system for visually challenged individuals. The proposed system leverages the power of IoT devices and machine learning algorithms to provide reliable and cost-effective assistance to visually challenged individuals, enabling them to navigate their

environment more independently and safely. The system's performance evaluation results demonstrate its potential effectiveness and usefulness in improving the quality of life for visually challenged individuals.

2.2. Mobile Based IoT Solution for Helping Visual Impairment Users:

1. The research paper titled "Mobile Based IoT Solution for Helping Visual Impairment Users" was published in the Advances in IoT journal in 2021. The paper helps visually impaired users navigate their surroundings by providing a mobile-based IoT solution.
2. The system under consideration comprises a network of interconnected IoT devices, including cameras, sensors, and beacons, along with a mobile application. It identifies the user's location and delivers live audio feedback to help them navigate their surroundings seamlessly. Additionally, the application can be personalized to offer more functionalities, such as voice commands and object recognition.
3. The experiments involved testing the accuracy of the system's object recognition capabilities, as well as its ability to detect obstacles and provide real-time audio feedback. The results of the experiments showed that the proposed system achieved high accuracy in object recognition and obstacle detection and provided reliable audio feedback to users.
4. One of the key advantages of the proposed system is its cost-effectiveness, as it uses low-cost IoT devices that are widely available in the market. The system's modular design also makes it highly scalable and easy to deploy in various environments, making it a potential solution for improving the mobility and independence of visually impaired individuals.
5. Overall, the research paper presents a novel mobile-based IoT solution for helping visually impaired users navigate their surroundings. The proposed system leverages the power of IoT devices and machine learning algorithms to provide reliable and cost-effective assistance to visually impaired individuals. The evaluation outcomes of the system display its prospective efficacy and utility in enhancing the living standards of individuals with visual impairments.

2.3. You Only Look Once paper for object detection:

1. The main objective of YOLO is to detect objects and locate them in a single pass of the neural network, enabling detecting objects in real-time.
2. The YOLO algorithm partitions the input image into a grid, and for each grid cell, makes predictions of bounding boxes and class probabilities. The predicted bounding boxes include the coordinates of the center of the box, along with its width and height, and a confidence score that indicates the algorithm's level of confidence that an object is present within the box. The class probabilities reflect the probability of the object belonging to a specific class.
3. During the training process, YOLO optimizes the sum-squared error between the predicted bounding boxes and the actual ground truth bounding boxes. To minimize false positives, the algorithm applies a method known as non-maximal suppression. This technique removes redundant detections and only retains the bounding boxes with the highest confidence scores.
4. Compared to other object detection algorithms, YOLO has several advantages, including faster detection speed and the ability to detect small objects. However, it may struggle with detecting objects that are closely grouped together or partially occluded.
5. The YOLO algorithm has undergone multiple enhancements since its initial release. The most recent version, YOLOv4, has achieved exceptional performance on various benchmarks. Due to its ability to achieve real-time object detection, the algorithm has been utilized in various fields, such as automated driving, security systems, and robotics.

2.4. Single-Image Depth Perception in the Wild:

1. In the paper "Single-Image Depth Perception in the Wild," the authors present a novel method using a single RGB picture captured in natural scenes to estimate depth. Unlike previous methods that require additional sensors or prior knowledge, this approach leverages deep CNNs to emulate human perception of depth from a single image using cues such as perspective, texture gradients, and occlusions.
2. The authors trained the CNN using a large-scale dataset called NYU Depth V2, which have the RGB-D snapshots captured by a Kinect camera in indoor scenes with diverse objects, layouts, and lighting conditions. They used the RGB-D images to generate ground-truth depth maps, which were then used to supervise the CNN training. The architecture of the CNN includes an encoder-decoder structure, where the encoder is responsible for extracting features from the image and the decoder is responsible for mapping these features to the corresponding depth map.
3. The method suggested by the authors underwent evaluation on various benchmark datasets, including Make3D, NYU Depth V2, and KITTI, and was discovered to surpass earlier methods concerning accuracy and generalization to different scenes. The paper also presents an analysis of the network's internal representations and its ability to handle occlusions, texture variations, and scale changes.
4. The authors introduced a novel loss function called Scale-Invariant Depth Loss (SID Loss) to address the scale ambiguity problem in depth estimation. The SID Loss penalizes errors in the ratio of predicted depth values between adjacent pixels, makes the depth map less sensitive to the global scale and improves the accuracy of depth estimation.
5. This method has potential applications in various fields, including robotics, autonomous driving, virtual reality, and others that require accurate depth estimation from single images. However, the authors acknowledge some limitations and suggest future directions, such as extending the method to outdoor

scenes, integrating other modalities such as LiDAR or radar, and improving the efficiency of the network for real-time applications.

2.5. Towards Robust Monocular Depth Estimation Mixing Datasets Depth Estimation: Mixing Datasets For Zero-shot Cross-dataset Transfer:

1. This study proposes a new approach to enhance the robustness of monocular depth estimation by training models on a mix of datasets. The authors contend that models trained on a single dataset might not generalize well to novel scenes and environments, and combining datasets during training can enhance the model's ability to transfer across datasets.
2. To train their model, the authors used two distinct datasets, KITTI and Cityscapes, which contain urban scenes captured by a car-mounted and pedestrian-mounted camera, respectively. These datasets have varying camera viewpoints, lighting conditions, and scene characteristics, making them ideal for testing the transferability of the model.
3. The authors evaluated their approach on both the KITTI and Cityscapes datasets and compared its performance against several modern monocular depth estimation methods. The proposed approach outperformed all other methods on both datasets, indicating its effectiveness.
4. The authors' contribution in this study is the model's zero-shot cross-dataset transfer capability. The authors demonstrated that their model could generalize to new datasets that were not part of the training dataset without any fine-tuning. This capability eliminates the need for costly data annotation and model retraining for each new dataset.
5. Since the publication of this paper, several follow-up studies have been conducted, building upon the authors' work. For example, in "Improving Monocular Depth Estimation with Cross-Dataset Transfer" by Xuelian Cheng et al. (2021), a similar approach was proposed to enhance the robustness of monocular depth estimation by combining datasets during training. However, they

utilized a domain adaptation technique to align the various datasets, improving the model's transferability.

6. In summary, the study makes a significant contribution to the field of monocular depth estimation by showing how the combination of datasets can enhance the model's robustness and transferability. The paper has also inspired several follow-up studies that have built upon the approach.

2.6. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks:

1. In this research paper, the authors suggest a novel approach to generating high-quality sentence embeddings. The proposed method involves using Siamese BERT-networks to overcome the limitations of existing methods that fail to capture semantic similarity between sentences.
2. The proposed method uses a Siamese BERT-network architecture to generate embeddings that capture semantic similarity between sentences. The authors train the model on a large corpus of text and the STS benchmark and the SentEval benchmark are used for evaluating performance. The findings demonstrate that the suggested approach outperforms previous methods both in terms of precision and computational effectiveness.
3. The impact of various pre-processing techniques and hyperparameters on the performance of the proposed method is also investigated by the authors. They find that techniques such as lemmatization and stemming can enhance the quality of the embeddings, and small amounts of domain-specific data can lead to further improvements in performance by fine-tuning the BERT Model.
4. One of the key contributions of the paper is the creation of a new benchmark dataset for evaluating sentence embeddings, called SBERT-STS. The authors argue that this dataset is more challenging than existing datasets and better reflects the real-world scenarios where sentence embeddings are used.
5. Overall, the research paper presents a novel method for generating high-quality sentence embeddings using Siamese BERT-networks. The proposed method outperforms existing methods on several benchmark datasets and demonstrates

the importance of capturing semantic similarity between sentences in natural language processing tasks. The creation of the SBERT-STS benchmark dataset is also a significant contribution to the field, providing a more challenging evaluation of sentence embedding methods.

2.7. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition:

1. The research paper is an influential contribution to the fields of NLP and speech recognition, providing an all-encompassing introduction to the fundamental concepts and techniques of these fields.
2. The paper is organized into four parts, starting with an overview of NLP and speech recognition, followed by a discussion of the basic techniques for processing language, including morphology, syntax, and semantics. The third part of the paper focuses on statistical and machine learning approaches to NLP and speech recognition, including language modeling and machine translation. Finally, the authors conclude the paper with a discussion of current research challenges and future directions in the field.
3. One of the strengths of this paper is its comprehensive coverage of a wide range of topics in NLP and speech recognition, making it an ideal resource for students and researchers in the field. The paper provides a thorough treatment of the fundamental concepts and techniques, including both rule-based and statistical approaches, as well as the latest developments in deep learning and neural networks.
4. Another important contribution of this paper is its focus on the practical applications of NLP and speech recognition, including text-to-speech conversion, speech-to-text conversion, machine translation, and information retrieval. The authors provide real-world examples of how these techniques can be used to solve practical problems, making the paper a valuable resource for practitioners as well.

5. Overall, the research paper "Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition" is a seminal work in the field of NLP and speech recognition. The paper provides a comprehensive introduction to the fundamental concepts and techniques of the field, including both rule-based and statistical approaches, as well as the latest developments in deep learning and neural networks. The paper's concentration on practical applications makes it a valuable resource for both students and practitioners in the field.

CHAPTER 3: METHODOLOGY

3.1. Object Localization

Object localization is a standard computer vision task. It involves finding the bounding boxes objects present in a picture. In our project, object localization is used to identify any object in the user's point of view and provide real-time response to the user.

One popular algorithm which is used widely for Object localization is YOLO.The YOLO algorithm employs a solitary neural network to forecast the class probabilities and bounding boxes. It is a single shot algorithm, that is, it only requires a single image unlike other algorithms like FAST RCNN and CNN. This makes YOLO highly suitable for real-time applications.

Another popular algorithm for object localization is the Single Shot MultiBox Detector (SSD), which is also a deep learning architecture for detecting objects. SSD algorithm partitions the image into a grid of cells and estimates class probabilities for every cell. This approach is highly accurate and works well for objects of various sizes.

Both YOLO and SSD are highly effective for object localization and are commonly used in a variety of applications. Both of them could be used in our application, as their characteristics align with our applications requirements.

The focus of our project is object localization, which is used to detect and locate obstacles along the user's path. The system provides real-time feedback to the user to help them navigate around these obstacles. By using advanced computer vision techniques like YOLO and SSD, we are able to accurately and efficiently detect obstacles and provide the user with the information they need to navigate safely and independently.

3.1.1. YOLO

You Only Look Once (YOLO) is a cutting-edge algorithm for detecting objects in real-time. YOLO is an end-to-end deep learning-based algorithm that can detect objects in an image and provide their bounding boxes and class probabilities.

One of the major benefits of YOLO is its high processing speed, it can process images in real-time on normal GPUs. This makes it ideal for real-time applications, like ours.

YOLO performs object detection using a single CNN, which makes it simpler and more efficient than other object detection algorithms that use multiple networks. The input image is divided into a grid and applies object detection to each cell of the grid, predicting each object’s bounding box and object class. Non-max suppression further helps it to reduce overlapping same class boxes and makes it robust for real world use.

YOLO's accuracy and efficiency have been demonstrated in various benchmarks and real-world applications, and it has been continuously improved in the latest versions, such as YOLOv3 and YOLOv4.

In our project, we utilized YOLOv3 for real-time object detection to detect anything in the user's direction and give immediate response to the visually impaired individuals. We integrated YOLO with our mobile application built on Flutter, enabling us to detect and alert users of incoming obstacles and potential hazards.

3.1.2. Our Methodology

For our use case we had 2 option

- Run YOLO algorithm on Edge
- Run YOLO algorithm on Cloud

Both the approaches had their pros and cons as follows -

Edge Computing

Mertis

1. Reduced dependencies on Flutter app
2. No connectivity required as all the computations will happen on the edge device itself.

Demerits

1. Due to size and processing power constraints, the time taken by Yolo was very large. It was more than the sum of CPU execution and HTTP latency over the same WiFi.
2. Reduced accuracy.

Cloud Computing

Merits

1. Good accuracy.
2. Faster computation time due to powerful processor and GPU support.

Demerits

1. Requires uninterrupted internet connectivity
2. Variable latency due to slow HTTP protocol.

We decided to proceed with Cloud computing as it met our requirements - ‘Low latency’ due to real time constraints and ‘Accuracy’.

The Yolo algorithm comes with a long list of labels and class ids. For our use case we decided to choose only relevant classes - ‘Person’, ‘Bicycle’, ‘Car’, ‘Motorbike’, ‘Bus’, ‘Truck’, ‘Bench’, ‘Cat’, ‘Dog’, ‘Parking’, ‘Bench’. These objects are most likely obstacles in the path of a visually impaired person. This truncated list of labels means that our model will be faster than straightforward YOLO.

The YOLO algorithm is wired up with a speech assistant. Whenever the user gives a command which needs to localize the position of an object, the flask API will call the YOLO algorithm to get the bounding boxes.

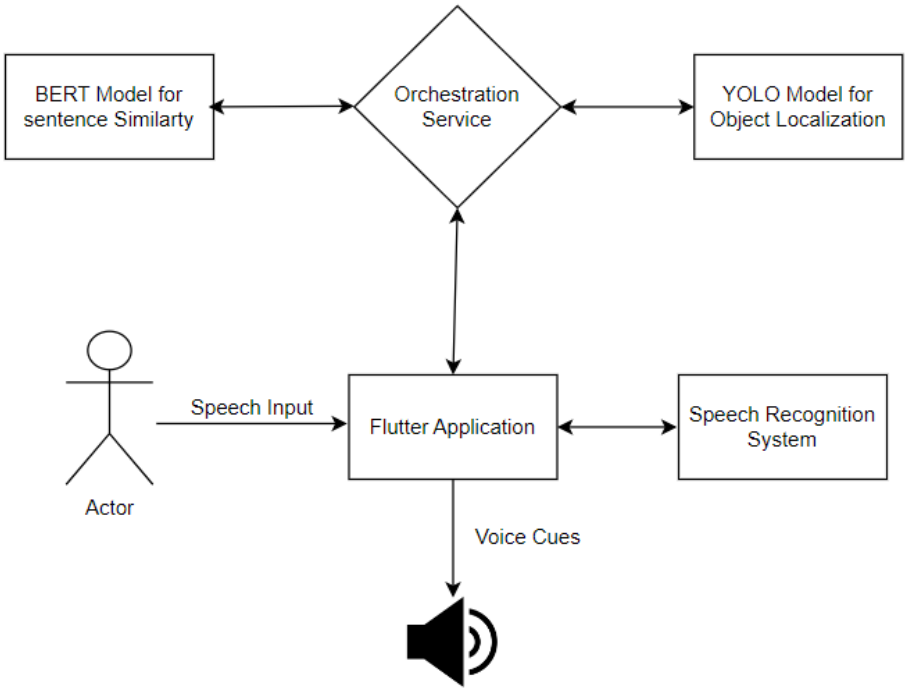


Figure 1 : How Yolo algorithm is triggered

3.2. Lane Detection

The idea behind incorporating lane detection was to enable visually impaired people to identify if they are walking in lane, on the left lane or on the right lane. Alone this functionality doesn't help that much but when combined together with object localization and depth estimation, which we will explore in the later sections, Lane detection can be very useful.

Formally, Lane detection is the task of identifying edges of lanes or roads from an image. It is a standard computer vision task and can be done by applying multiple operations in a specified order.

3.2.1. Methodology

We are perform lane detection by performing the following operations in the given order-

1. Convert image to grayscale
    a. Processing a three channel image is computation expensive. Analyzing gradients in a grayscale image will have the same results as calculating gradients in 3 channels.
    b. This is done using OpenCV inbuilt functions.

2. Apply Gaussian blur
    a. It is performed to eliminate any noise present in the image.
    b. It is done by kernel convolution of a Gaussian kernel over the image.
    c. The Gaussian filter /kernel is a normally distributed filter. By convolving it, each pixel intensity is replaced by a weighted average of nearby pixel intensities.

3. Apply Canny Edge Detector

    a. The algorithm is designed to detect edges in an image through multiple stages.
    b. Gradient Calculation - A sobel kernel is used to calculate gradients along X and Y direction.

$$K_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad K_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Figure 2 : Sobel kernels for X and Y axes

c. Calculation of magnitude and slope of gradient at each pixel.

$$|G| = \sqrt{I_x^2 + I_y^2}$$

$$\theta(x, y) = \arctan\left(\frac{I_x}{I_y}\right)$$

Figure 3 : Gradient calculation - slope and magnitude

d. Non-max suppression to thin the edges and preserve only the dominant edges.

4. Image Segmentation

a. By having a fixed position of a camera, we can reduce the features(edges) in the image by only taking the necessary regions into consideration.
b. In our case we use a trapezoidal segment of an image for lane detection.
c. The image is segmented by creating a trapezoid mask and taking bitwise AND with the output of a canny edge detector.

5. Hough Transform

a. We go into Hough space to analyze and find which line segments are present in the image.
b. A line in hough space is a point and the lines passing through a point form a sinusoidal curve.

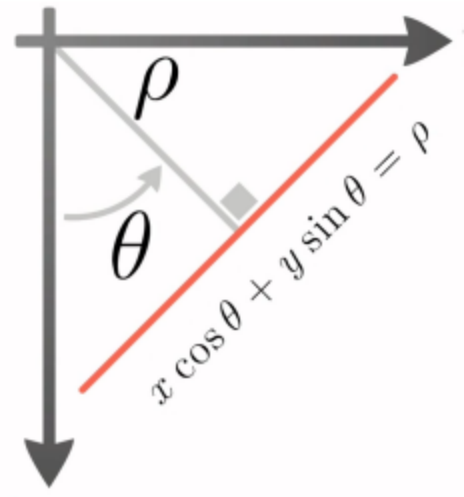


Figure 4 : Parametric representation of line using ρ and Θ

c. For each pixel with high gradient after applying a canny edge detector, we add 1 vote for all points of the sinusoidal curve.

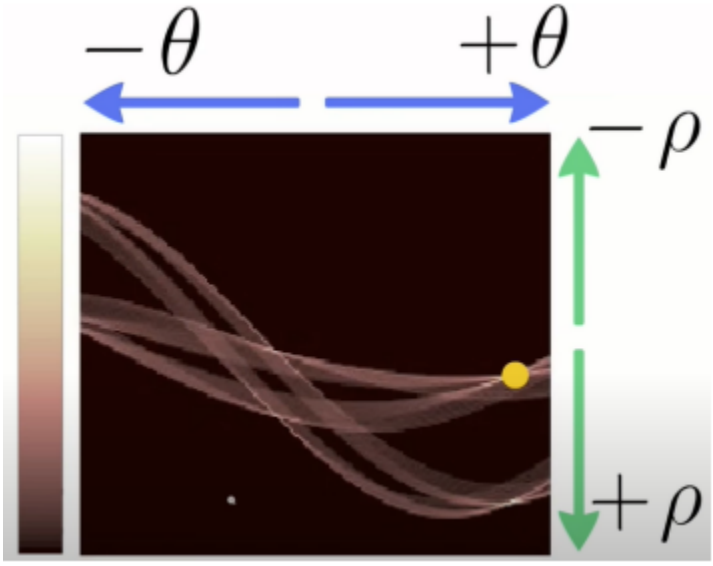


Figure 5 : Voting of lines in Hough Space

d. The lines with votes greater than a threshold value are chosen as hough lines.

There exist two parametric equations for lines -

1. $y = mx + c$
2. $\rho = y \sin\theta + x \cos\theta$

We use the parametric equation with ρ and Θ. This is so because we cannot represent lines with slope as infinity using ‘m’ and ‘c’. The second parametric equation can be used to represent all possible lines in a 2D plane.

3.3. Depth Estimation

Depth estimation is a critical aspect of computer vision that enables machines to perceive the depth information of the surrounding environment. In our project, we used depth estimation to determine the distance of the obstacles detected by our object detection algorithm, allowing us to provide more accurate feedback to visually impaired individuals.

To estimate depth, we used a monocular depth estimation-based algorithm trained on the KITTI dataset, which contains a large number of annotated images with corresponding depth maps. This algorithm uses deep learning techniques to examine the relationship between the image features and the respective depth values, allowing us to determine the depth of the objects in the user's path.

Depth estimation is essential to our project as it enables us to provide more precise feedback to the visually impaired individuals. By determining the distance of the obstacles, we can warn the user of potential hazards in advance and provide them with more accurate guidance for navigation. For example, if an obstacle is detected to be close, the application can alert the user to stop or change their route.

Furthermore, depth estimation can be applicable in a variety of computer vision applications too like autonomated vehicles, robotics and AR. It allows machines to perceive the work in three dimensions, providing more accurate information for decision-making.

In our project, the use of depth estimation and object detection algorithms, along with other techniques such as lane detection and speech-to-text, creates a comprehensive solution for visually impaired individuals to navigate roads safely and independently. The integration of these techniques enables the app to provide real-time feedback and guidance to the users, enhancing their overall mobility and quality of life.

3.3.1. KITTI Dataset

The KITTI dataset is a valuable resource for autonomous driving research. It provides a comprehensive set of data that includes stereo and optical flow information, 3D object detection, and road network data. The dataset is used to evaluate computer vision algorithms for depth estimation, scene understanding, and visual odometry.

One of the primary applications of the KITTI dataset is depth estimation. This task involves predicting the distance of every pixel from the camera, which is essential for accurate perception and planning in autonomous driving applications. The KITTI dataset includes high-resolution images, LiDAR point clouds, and GPS/IMU data, which makes it an excellent resource for depth estimation research.

The KITTI dataset includes a vast collection of outdoor urban scenes captured from a moving platform. The scenes are diverse and complex, and they present a range of challenges for depth estimation algorithms. Researchers can use the dataset to test and refine their algorithms.

The KITTI dataset has been widely used in academic research and industry. Its availability has contributed to the development of new techniques and algorithms for depth estimation, scene understanding, and visual odometry. As autonomous driving technology continues to evolve, the KITTI dataset will likely remain a crucial resource for researchers and engineers.

3.3.2. Monocular depth estimation

The technique of monocular depth estimation involves estimating the depth of each point in a scene using a single image, making it a valuable tool in computer vision. In our project, we utilized this technique based algorithm to determine the distance of obstacles detected by the object detection algorithm.

The monocular depth estimation algorithm utilizes a CNN to estimate the depth map of the input image. The depth map is a 2D array of values representing the distance of each

pixel in the image from the camera. The CNN is trained on a large dataset of images with corresponding depth maps, allowing it to learn the depth perception features from the input image.

In our project, we used the monocular depth estimation algorithm to determine the distance of obstacles detected by the object detection algorithm. By combining the bounding box coordinates of the detected object and depth map, we can estimate the distance of the obstacle from the user. This information is crucial in providing feedback to the visually impaired individuals about the surrounding environment.

In summary, the monocular depth estimation algorithm is a valuable tool in our project, allowing us to determine the distance of obstacles detected by the object detection algorithm and provide real-time feedback to visually impaired individuals. Its accuracy can be improved by employing advanced techniques such as multi-scale depth estimation and incorporating additional depth cues.

3.3.3. Methodology

We first used the CNN model to get a depth map of the image. Now the problem with the depth map is that it only gives a relative distance. Our workaround : On the pixel intensity of that matrix, we fix the last row’s intensity as being proportional to the distance to the floor as we intend the user to focus a bit towards the ground.

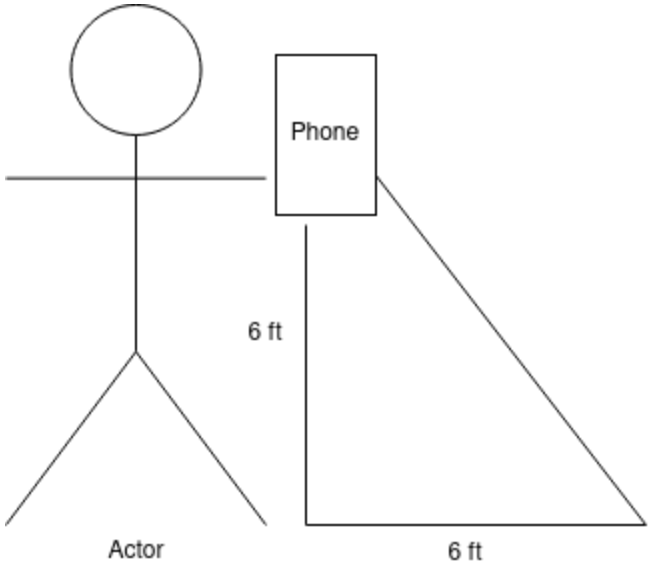


Figure 6 : Setup to get absolute distance from absolute distance

Assuming the height of the user to be 6ft and the floor in focus to be 6ft ahead of him, the distance from the floor to the camera comes out to be 6√2 ft (or 2.5863 meters).

Now assuming the distance of any other object to be I, we get:

$$\frac{Distance\ of\ point\ k\ from\ user}{Intensity\ at\ point\ k\ in\ the\ depth\ map} = \frac{Distance\ of\ floor\ from\ user}{Intensity\ at\ floor\ point\ in\ the\ depth\ map}$$

Hence,

$$Distance\ of\ point\ k\ from\ user = \left(\frac{Distance\ of\ floor\ from\ user}{Intensity\ at\ floor\ point\ in\ the\ depth\ map}\right) * 2.5863$$

meters.

Now, we get the distance of every other point of the image matrix. But the distance of the bounding boxes is required. Let the nearest point in the bounding box be our distance from the object, since some of the points in the bounding box can be of far away background, Distance of bounding boxes = Min(Distances of all points in a bounding box given by the YOLO algorithm).

3.4. Navigation

Navigation is a critical aspect of our project as it enables visually impaired individuals to navigate roads and reach their destinations independently. We integrated Google Maps API into our mobile application, allowing users to input their desired destination and receive turn-by-turn directions.

The addition of navigation to our app significantly enhances the independence and mobility of visually impaired individuals. With the ability to receive real-time guidance and feedback, users can confidently navigate roads without relying on assistance from others. Navigation also reduces the risk of getting lost or disoriented, providing a sense of security and freedom to the users.

Moreover, the integration of navigation with other techniques such as object detection and depth estimation creates a comprehensive solution for visually impaired individuals to navigate roads safely and independently. By combining these techniques, the app can provide real-time feedback and guidance to the users, alerting them of potential hazards and obstacles while providing turn-by-turn directions to their destination.

Overall, the addition of navigation to our app significantly improves the overall user experience, enhancing their independence and mobility. With real-time feedback and guidance, visually impaired individuals can navigate roads confidently and reach their destinations safely and independently.

3.4.1. Google Maps API

Google Maps API is a useful resource for integrating location-based services into mobile applications. In our project, we used the Google Maps API to provide navigation for visually impaired individuals. This feature allows users to input their destination and receive step-by-step directions to reach their desired location.

To integrate the Google Maps API into our application, we first obtained an API key from the Google Cloud Console. We then added the necessary dependencies to our Flutter project and used the API to provide directions to the users.

The users can input their destination using voice commands, which are converted to text using speech-to-text technology. The app then sends the destination information to the Google Maps API, which provides a route to the destination. The directions are presented to the user using audio feedback, providing step-by-step guidance to reach their desired location.

The integration of Google Maps API significantly enhances the navigation experience for visually impaired individuals, providing them with a reliable and efficient way to navigate roads. The use of audio feedback allows users to keep their hands free and focus on their surroundings, enhancing their overall safety and independence.

Overall, the integration of Google Maps API, along with other technologies such as object detection and depth estimation, creates a comprehensive solution for visually impaired individuals to navigate roads safely and independently. The integration of these technologies enables the app to provide real-time feedback and guidance to the users, enhancing their overall mobility and quality of life.

3.4.2. Methodology

We implemented the google maps API in flutter instead of reinventing the wheel. The flutter app works with taking the current location of the phone via user permissions in the android app, and via voice control asks for the final destination he user wants to go to.

The app then sends the required parameters to the API and gets the response as a json object containing HTML tags of the directions required to reach the final step.

The HTML tags that contain the information is due to the API being predominantly used for web applications.

We parse the HTML and get the required steps and then render them on to the screen as well as prompt them via voice output.

3.5. Flutter App

3.5.1. Flow Diagram

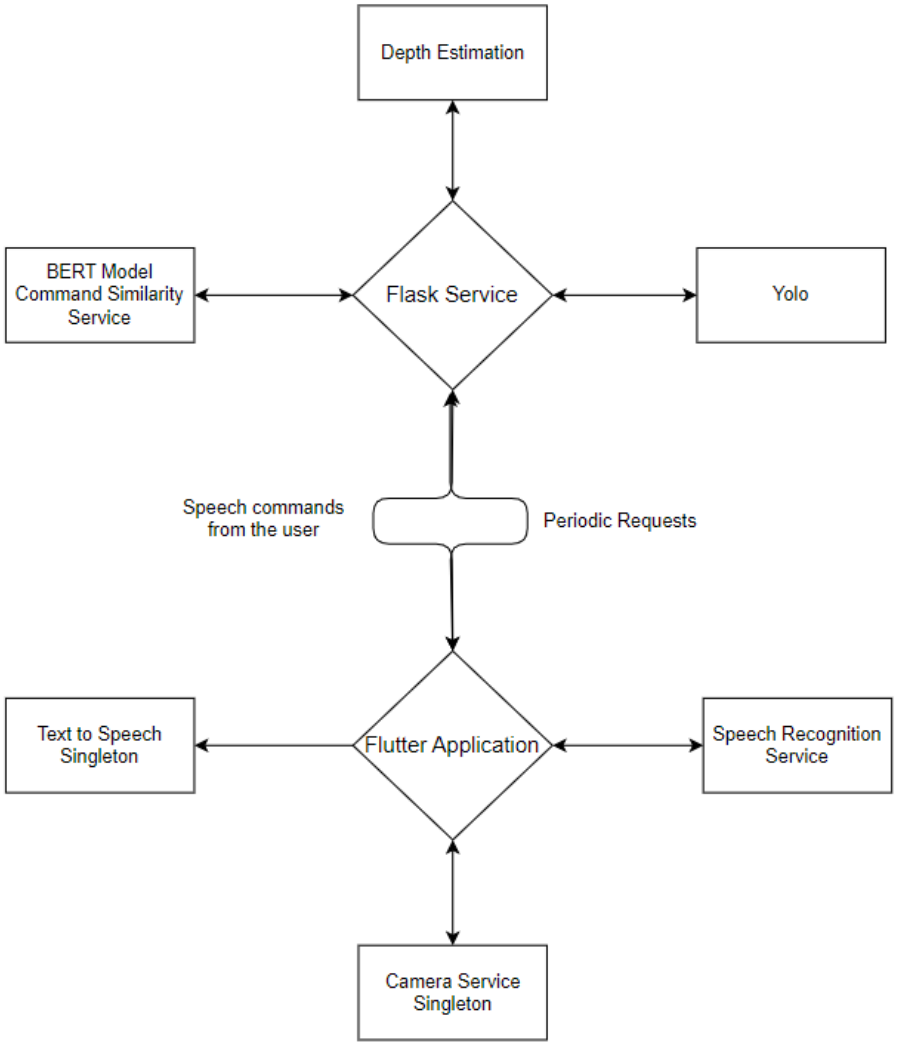


Figure 7 : Architecture of Application

The Flutter application will interact with the backend Flask API in 2 ways -

1. Periodic heartbeat messages
2. Instructions invoked by the user

The Flutter app will capture an image every 10 seconds and send the image to Flask API. The Flask API will perform Object localization and Depth estimation on the image. The results of both these processes will be overlapped to get the distance or locations of the objects present in the path of the user.

Methods can also be triggered by the user. The user can give commands via speech. The Speech Recognition system will convert the speech into text. The Flutter app will pass this text along with an image from Camera to the Flask API. The Flask API will process the request by finding the command which matches the input given by the user, this is done using the 'BERT based Sentence Similarity service'.

After identifying the command, it will run the appropriate procedure for that given task and return a textual output to the Flutter app. The Flutter app after receiving this textual response will send it to 'Text to Speech service' which will intimate the user.

3.5.2. System Design

The task of System design can be split into two categories-

1. High Level Design
2. Low Level Design

3.5.2.1. High Level Design

High level design includes working with an abstraction of the actual systems. It is used to uncover important characteristics like - communication type(half duplex, full duplex), dependencies between components etc.

The major HLD decision we have to take in our architecture was to choose between -

1. Iterative Querying
2. Recursive Querying

The concept of Iterative and Recursive querying comes from the domain of ‘Domain Name Services’ or DNS. We will explain our architecture by drawing similarities between them.

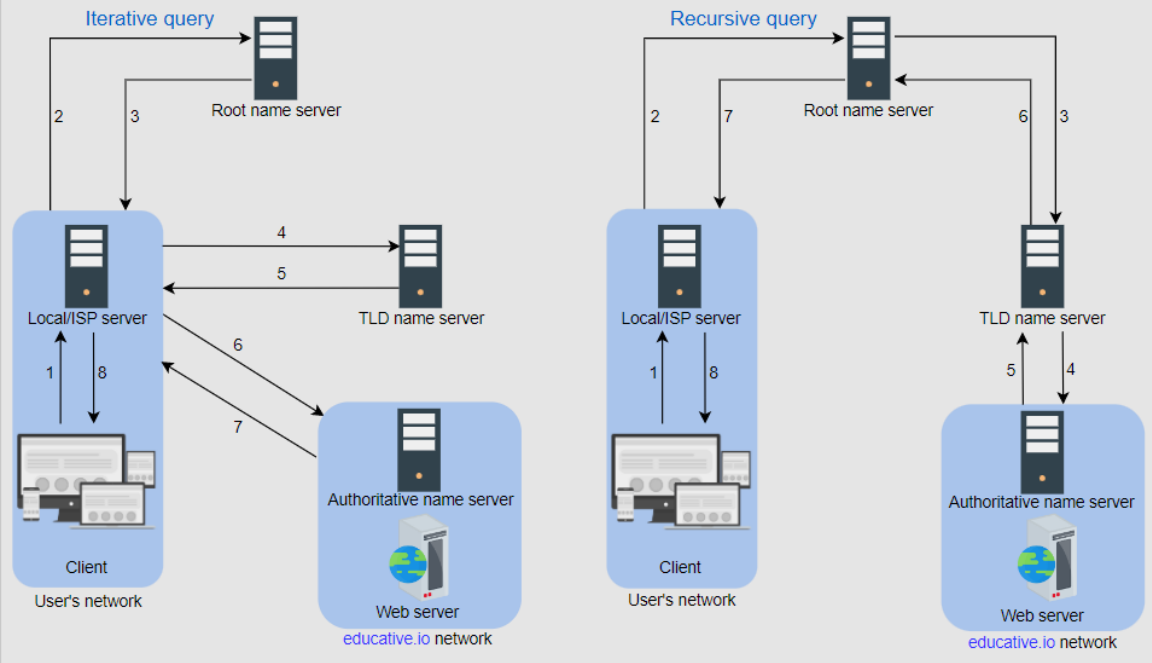


Figure 8 : DNS architectures - Iterative and Recursive

In Iterative query, the client or the user’s system has to make multiple calls to different servers to get the IP address. It first queries the Root Name server to get the address of TLD servers, then it queries TLD servers to the address of the authoritative server which will have the address of requested web server. After getting the final IP address, it can make HTTP requests. In this method, orchestration occurs, i.e, the client is responsible for orchestrating multiple servers.

In Recursive querying, Choreography occurs. The client only makes one request to Root name servers and the Root name server itself communicates with TLD servers to get the requested IP address. In this case the load on the client application is less.

DNS uses iterative querying because it has to adhere to the needs of millions of devices. Our requirements revolved around making the client application as light as possible and doing the majority of computation on the server. Therefore we decided to go with the 'Recursive Querying' approach.

The Flutter app will only make one HTTP request to Flask API. The Flask api will call multiple services and procedures based on the payload it received from Flutter.

Example workflow :

1. User gives a Voice command - 'What is in front of me?'.
2. The Speech recognition module converts the command to text.
3. The Camera module captures an image simultaneously.
4. The text command and image is then sent to the flask API.
5. The Flask API then sends the text to the BERT based Sentence Similarity module to identify the command type.
6. After identifying the command, in this case 'User wants to know what in its vision', the Flask API will first call the Yolo algorithm.
7. The Flask API then calls the Depth Estimation service.
8. A procedure is executed which maps the bounding boxes of Yolo and Depth map of Depth Estimation to identify the precise location of objects.
9. The output to be given to the user is formulated and returned back to Flutter.
10. Flutter upon receiving the text based message, calls the Text to speech service to intimate the user.

In this entire process, the Flutter App made only 1 API call and all the orchestration was handled by Flask service itself, thus reducing the complexity of the client application.

3.5.2.2 Low Level Design

Singleton Classes

Singleton classes is a OOPs concept in which a class is defined such that only one object of the class is created. This means that the class can have only 1 instance.

This approach is possible due to multiple OOPs concept -

1. Private Constructors - Yes, many languages like Dart, Java can have classes with private constructors. This is done to restrict the instantiation of class objects.
2. Static Data members - These are data members but a twist. The twist every class maintains only 1 copy of static variables for all class objects. If multiple objects are created then they will share the static variables. Even if a class is not instantiated, its static member functions and static data members can still be accessed and used.
3. Instance - It is the only object of a singleton class. It is -
    a. Public data member
    b. Static data member
    c. Calls a private constructor when referenced for the first.

This approach of using Singleton classes with instances is a very popular Low level design pattern. It is used by Google Firebase as well.

Having a singleton was very useful because now we can implement a queue based logic for Text to speech and Camera. Consider the situation where multiple objects could be created, then it is possible that 2 or more objects called the Camera service at the same time, this would result in none of them getting an image.

Queue based logic

We created queues for Text to speech service and Voice assistant. A while loop is running for both the services, if a user gives a command and another command is under processing, then the command will be pushed to the queue. The service will eventually

pick up the command and execute it. This ensures that all the user's commands are executed and there is no overlap between 2 commands.
Furthermore to save processing power, when the queues become empty, the while loop is stopped. As soon as a command is pushed into the queue, the while loop is triggered again so that new commands can be executed.

3.5.3. Single threaded vs multi-threaded

Multi-threading was considered while building the application because we have 2 two types of actions that can occur simultaneously. We can run one type of process on one thread and another type on another thread. This would make the app a lot faster.

It was easier said than done. The reason being the nature of multi-threading in Dart. Dart threads cannot share memory between them. Therefore, we will end up creating different singleton instances on both the threads. But both the singletons will use the same camera, hence blocking each other, hence causing a deadlock.

But we were able to build an optimized single threaded application whose performance will be similar to that of a multithreaded application by using the Queue based logic and Singleton classes pattern.

3.6. Voice Control

A computer vision application usually has outputs in forms of images and depth maps that can be perceived by human eyes. But the end users of our application will be people with visual impairments, therefore it only made sense to not use visual sense but auditory sense.
The user can give commands to the application via voice cues and will get prompts from the application via voice commands.

3.6.1. BERT Model

BERT stands for Bidirectional Encoder Representations from Transformers. It is a pre-trained language version advanced with the aid of Google. It is a deep neural network architecture that is trained to recognize natural language. it could perform obligations like text classification, query-answering, and named entity recognition . In contrast to traditional language fashions, BERT is bidirectional, cutting-edge; it could study the entered text in both directions. This lets BERT to seize the context and meaning of a modern phrase based on the words that come earlier than and after it.

One of the key features of present day BERT is that it could be changed for various NLP responsibilities consisting of sentiment evaluation, textual content type, and query-answering. These adjustments encompass taking the pre-trained BERT model and training it on a smaller dataset particular to the venture handy. This fine-tuning manner allows the model to obtain performance on various NLP benchmarks.

BERT has been broadly adopted and has ended up one of the most popular language models in natural language processing. Its capacity to understand the context and meaning of modern-day phrases has caused breakthroughs in several NLP tasks, including sentiment evaluation, device translation, and text summarization.

3.6.2. Command Set

Building a voice assistant from scratch would take a lot of time and research and using a prebuilt general purpose voice assistant would be redundant because we would never use all its features and our application will have its proprietary features.
It only made sense to make our own lightweight voice assistant. We defined a set of commands the user will probably ask while navigation.

1. What is in front of the user?
2. Is there a vehicle approaching the user?
3. Is there a stray animal in the user's path?
4. Is the user walking in the left lane?
5. If the user uses public transport, has the bus arrived or not?
6. Is the user's path clear and he/she can proceed forwards?
7. Is there a person in front of the user?

This list is not exhaustive, more commands can be easily added. Currently the Flask API has procedures to respond to these commands only.

3.6.3. Methodology

The Voice assistant we build is based on Google's BERT Model. We use Speech recognition to convert speech to text. This text is then passed to the BERT Model and we extract the last_hidden_state tensor.

The dimensions of this tenstor in the BERT Base model is [128 X 768], a tenstor of size [1 X 768] for every word in the sentence or token.

BERT Base model takes a 128 length sentence as input and if the length is less than 128 it adds padding to the sentence. We don't want to include these padding tokens. We remove their contribution by multiplying the tensor values of padding tokens by 0 and for the rest of the tokens we multiply the tensor values by 1.

After removing the contribution of padding tokens, we do mean pooling and create a final vector of size [1 X 768].

We precalculate this [1 X7 68] tensor for all commands. This is simple optimization, which eliminates any redundant computation.

This [1 X 768] feature vector has semantic information about the sentence. We can use any similarity index to find the semantic similarity between the sentences.We utilize Cosine Similarity to measure the similarity between the feature vectors in our case. The chosen command is the one which has the maximum cosine similarity with the input sentence.

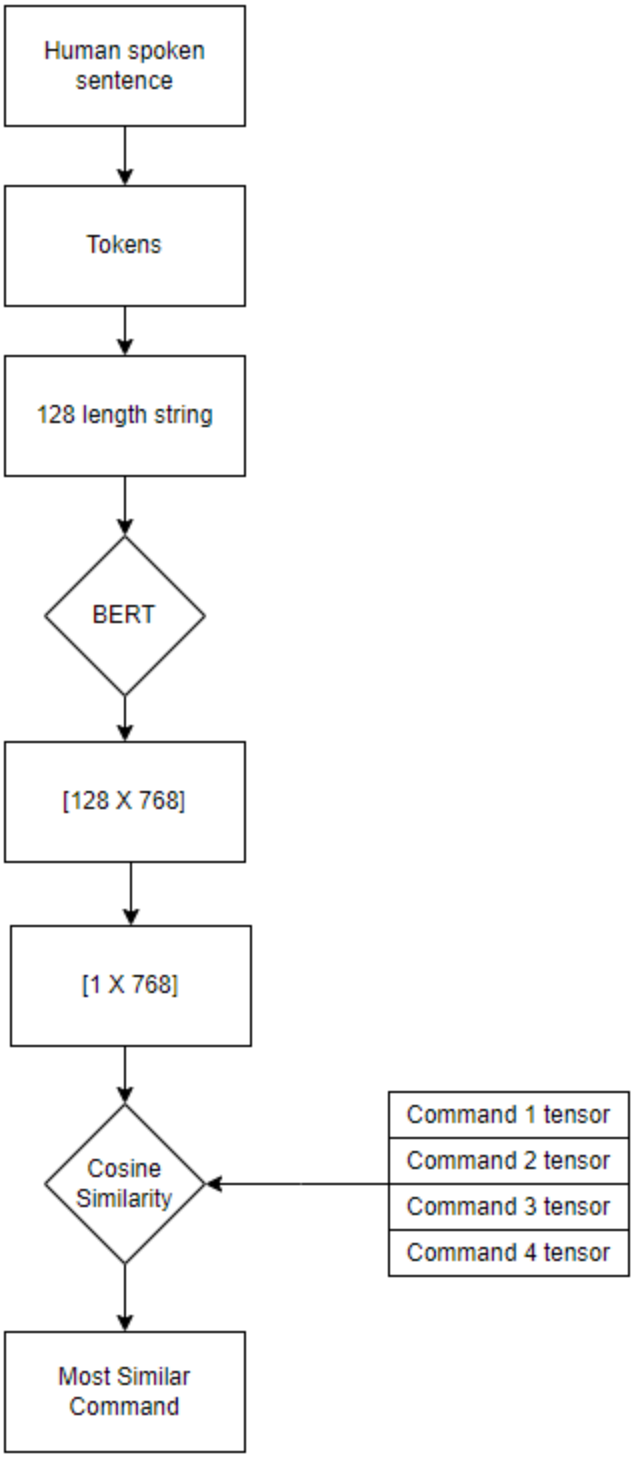
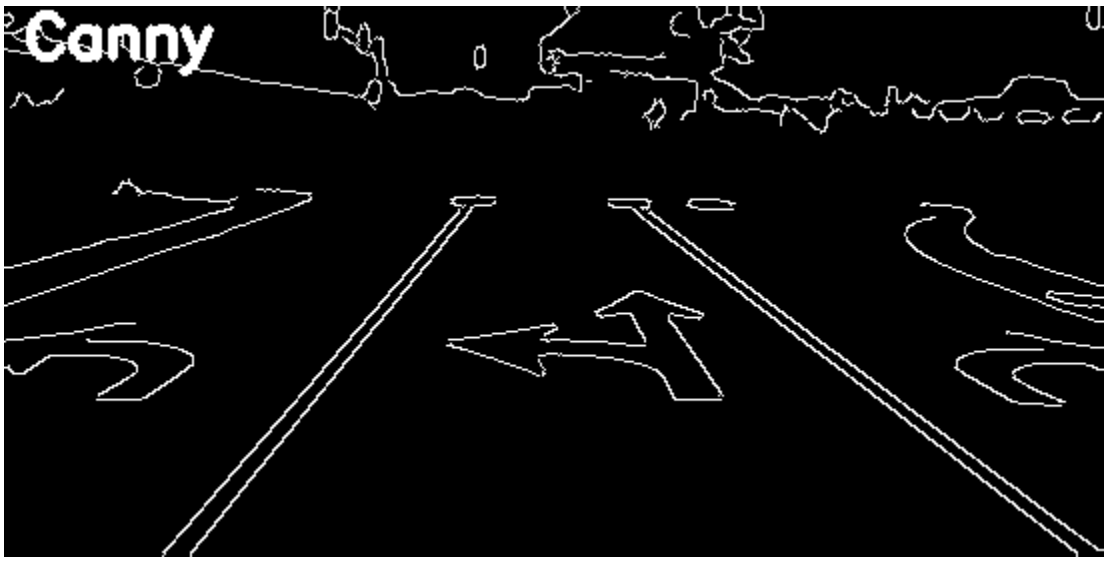


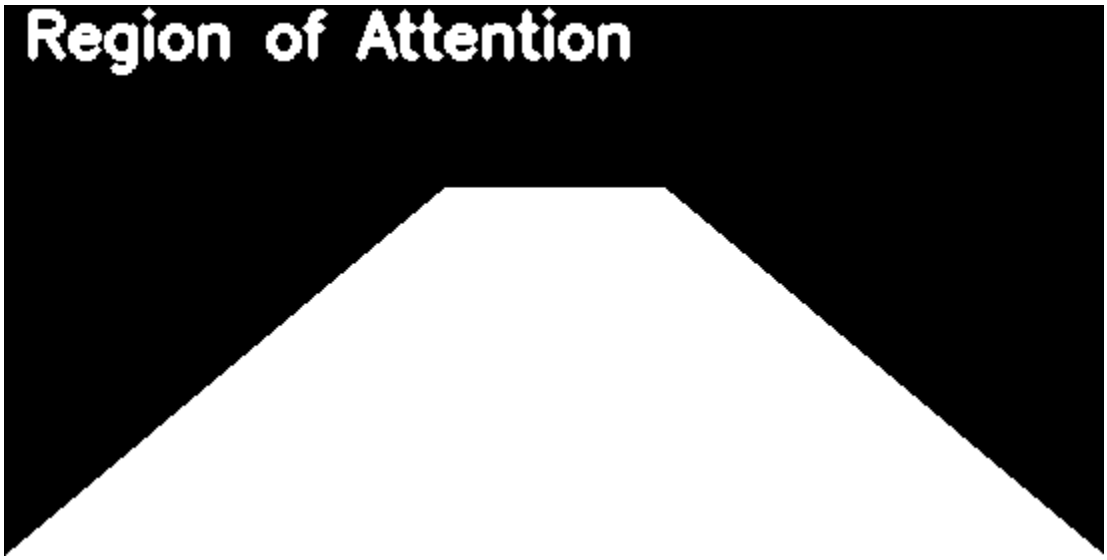
Figure 9 : Voice assistant working

CHAPTER 4: RESULTS, CONCLUSION AND FUTURE WORK

4.1 Results

4.1.1. Lane Detection : We get the following results while applying the lane detection algorithm on test images.





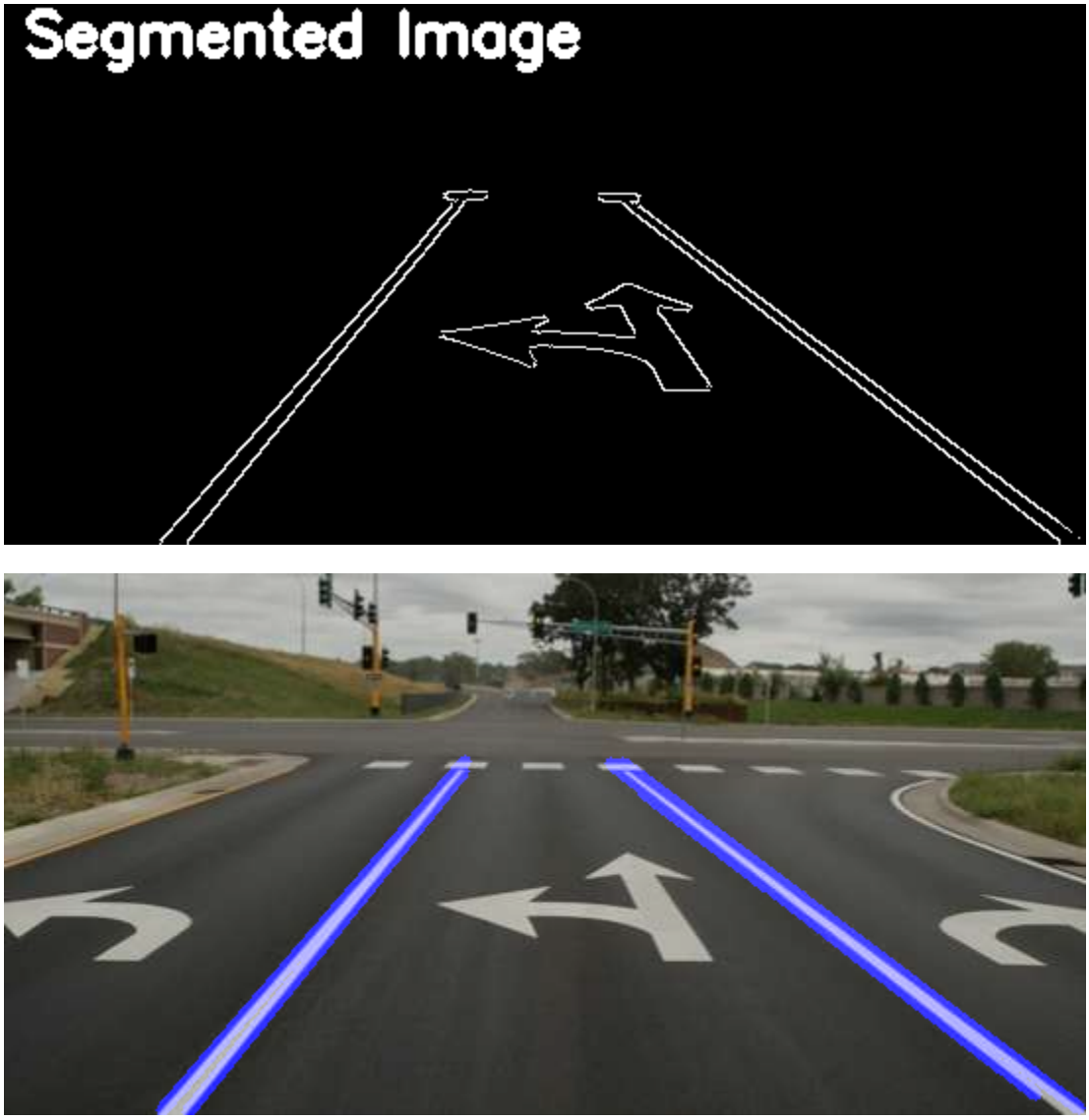


Figure 10 : Lane Detection results

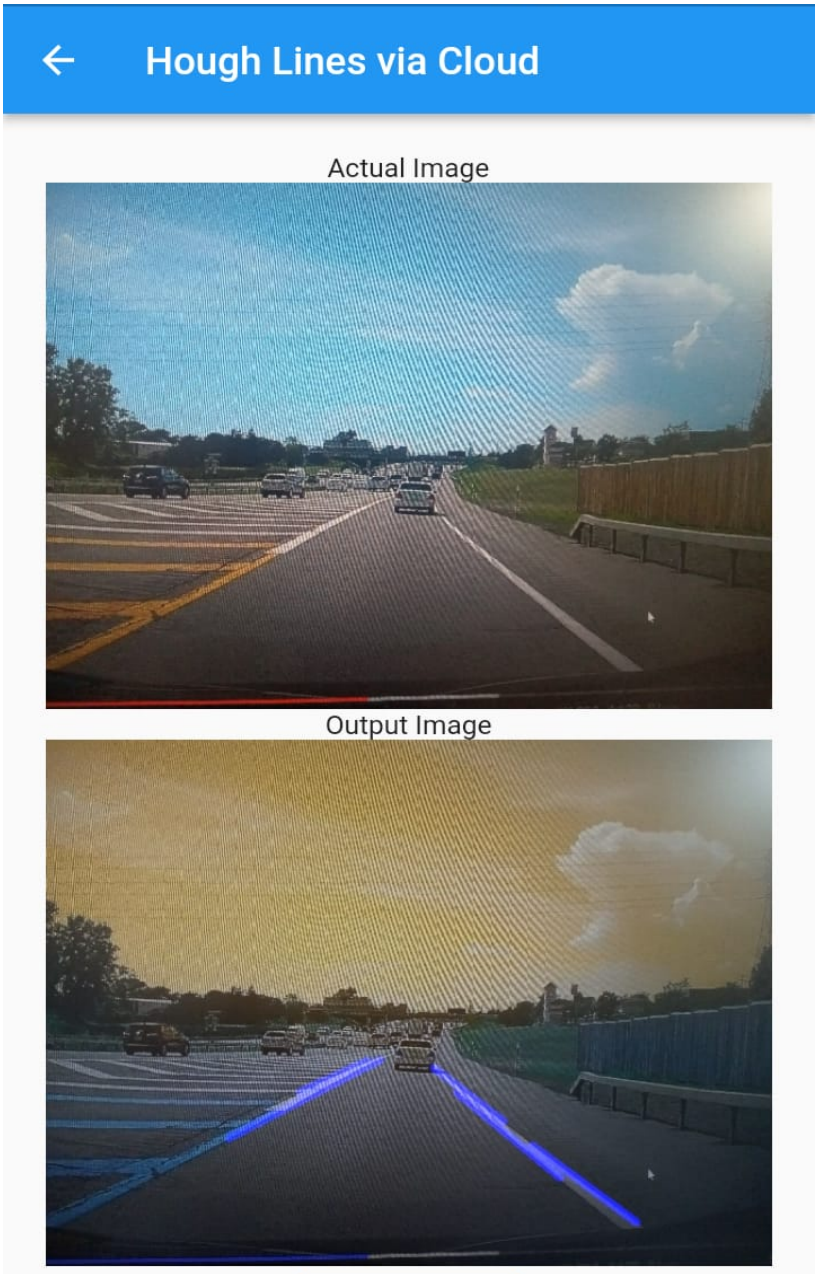


Figure 11 : Lane detection on road

4.1.2. Object Localization

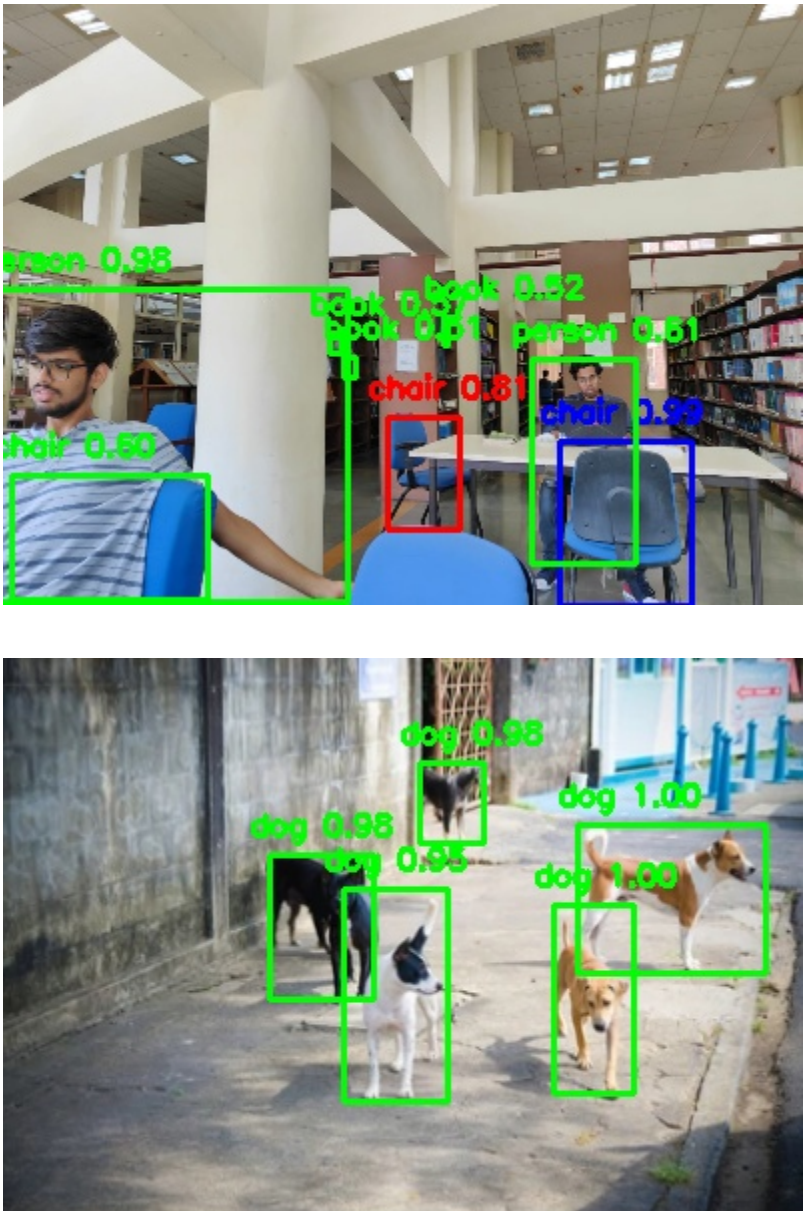


Figure 12 : Yolo algorithm results

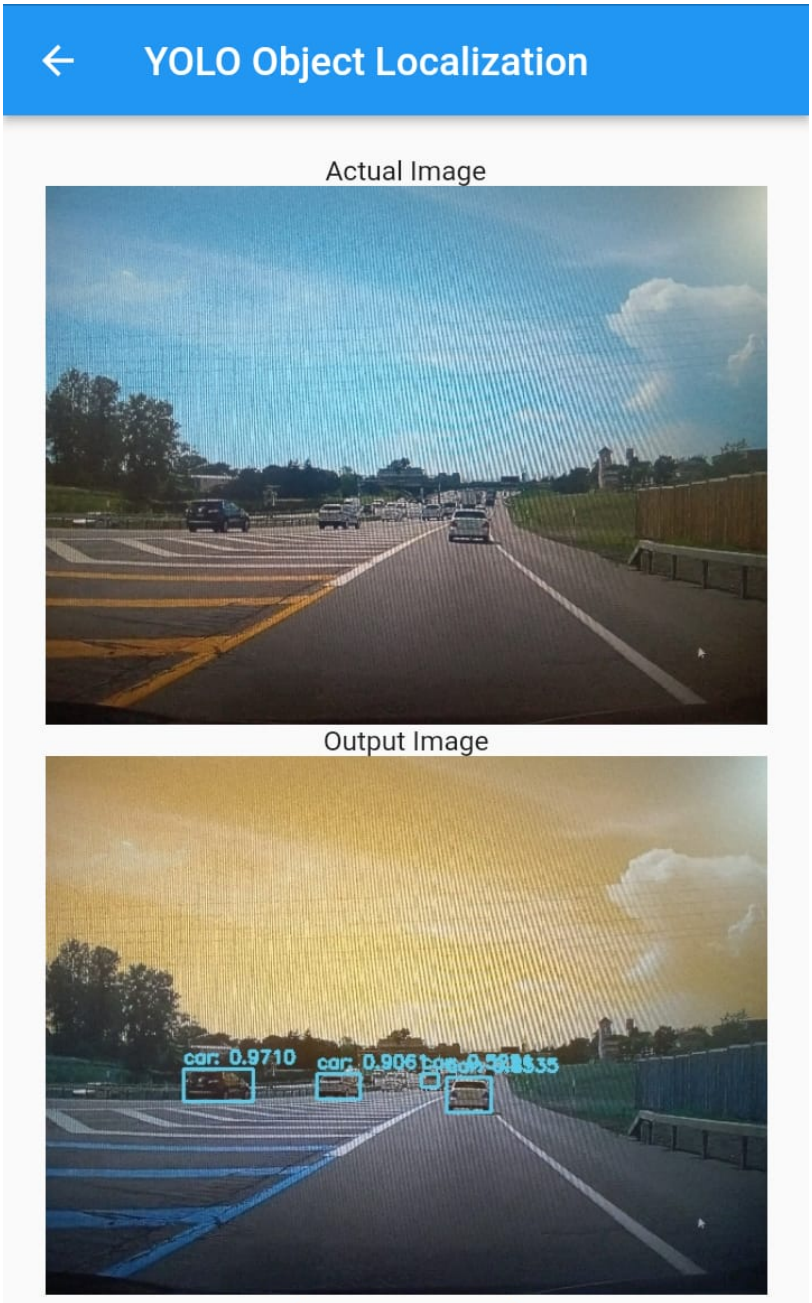


Figure 13 : Yolo result on road

4.1.3. Depth Estimation: We applied the depth estimation neural network that gives a depth map of an image given as input. Plotting the depth map and rendering it simultaneously with the original image, we get the following results shown below.

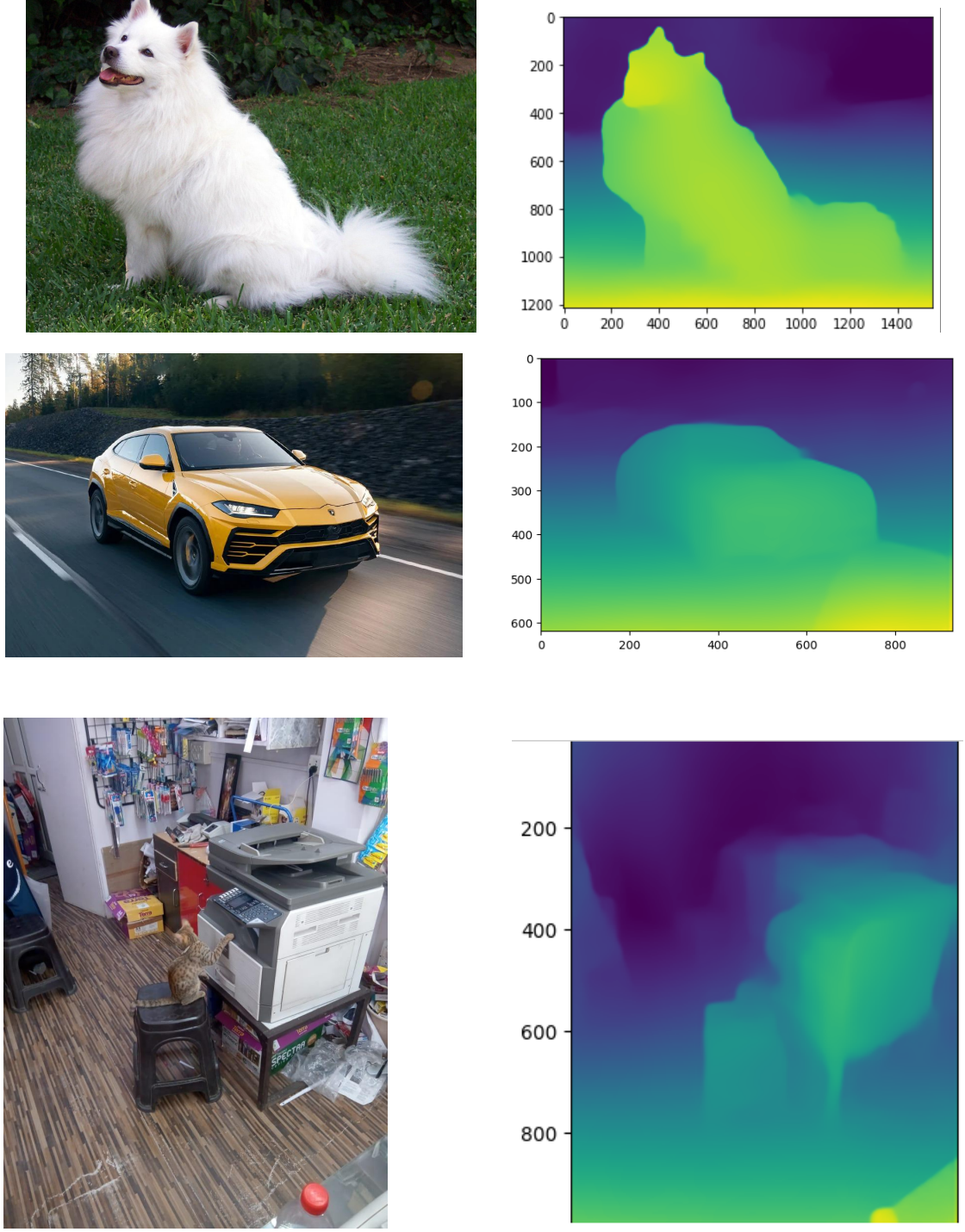


Figure 14 : Depth estimation results

4.1.4. Navigation : We got the navigation directions from the current location of the user to any other location that the user might want to go. A sample of navigation results that we implemented in the final app is shown below.

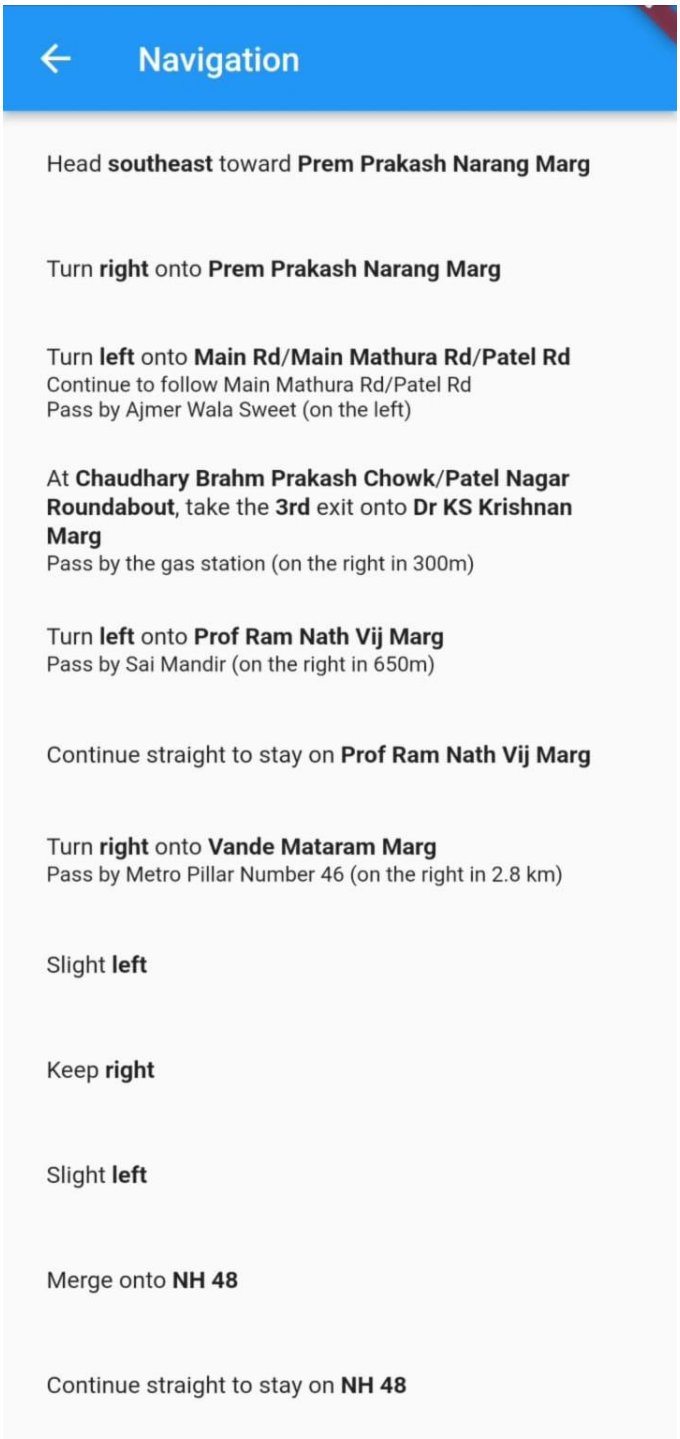


Figure 15 : Google Maps API navigation prompts

4.1.5. Response time and latency

S No	Task	Time taken (seconds)
1	Canny Edge Detection	0.0688 + c.l.
2	Image Segmentation	0.0128
3	YOLO Algo	0.09254
4	YOLO (Algo + image processing)	1.1026 + c.l.
5	Hough Lines	0.2767 + c.l.

c.l. = communication Latency

Table 1 : Cloud Computing average time(excluding communication latency)

S No	Task	Time taken (seconds)
1	Canny Edge Detection	1.705
2	Image segmentation	0.935
3	YOLO Algo	2.368
4	Hough lines	1.656

Table 2 : Cloud computing average time

S No	Task	Time Taken (seconds)
1	Canny Edge Detection	0.234
2	Hough Lines	0.306

Table 3 : Edge computing average time

4.2 Conclusion

In conclusion, we were able to make a low cost assistant for the visually impaired by using computer vision techniques and machine learning models like Lane detection, Object localisation, Depth estimation, Natural language processing and by combining it with Google Maps API the product became a lot more usable.

The Flutter app is working perfectly. It is reliable and stable mainly because of good Low level design and efficient High level design. The prompts given by the application are accurate and useful to the user. It is also able to identify the user commands and give appropriate responses.

The accuracy of machine learning models is also satisfactory. Yolo is able to identify major obstacles and depth estimation is also given accurate depth maps. Voice assistant uses Google's BERT model therefore we can be sure of its accuracy.

4.3 Scope of future work

Currently our prototype uses ML models deployed on cloud. The reason is processing constraints on a mobile phone. In the future we can get rid of this 2 tier architecture and perform all the computations on edge itself by incorporating a slightly power GPU.

Also, the end device currently needs to have an active internet connection. Internet connection is still necessary for GPS navigation but for other tasks like object localization and depth estimation, computations can be performed on edge, therefore removing the requirement of a Flask API. Although this will make the app on edge a lot more complex, it can be made efficient and implementable using multiple threads. Programming languages like Java, which has extensive support for mulit-threading are viable options.

The current app is a prototype only, if we decide to go all in with a product, we can even use 2 cameras to absolute distances rather than relativistic distances which we are getting using Monocular depth estimation. This will also reduce the processing power requirements, but concurrency between two different cameras is a challenging task to solve(both the cameras should take pictures at the same moment, exact to milliseconds).

REFERENCES

1. Ramaiah, N. S. (2019). IoT Based Route Assistance for Visually Challenged.
2. Abdel-Jaber, H., Albazar, H., Abdel-Wahab, A., El Amir, M., Alqahtani, A., & Alobaid, M. (2021). Mobile Based IoT Solution for Helping Visual Impairment Users. Advances in Internet of Things, 11(4), 141-152.
3. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).
4. Ranftl, R., Lasinger, K., Hafner, D., Schindler, K., & Koltun, V. (2020). Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. IEEE transactions on pattern analysis and machine intelligence, 44(3), 1623-1637.
5. Chen, W., Fu, Z., Yang, D., & Deng, J. (2016). Single-image depth perception in the wild. Advances in neural information processing systems, 29.
6. Reimers, N., & Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. arXiv preprint arXiv:1908.10084.
7. D. Jurafsky, J. H. Martin, P. Norvig, and S. J. Russell, Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition, Second Edition, Pearson International Edition. Upper Saddle River, NJ: Prentice Hall, Pearson Education International, 2009.

PLAGIARISM REPORT

ORIGINALITY REPORT

6%	4%	3%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Netaji Subhas Institute of Technology Student Paper	1%
2	www.coursehero.com Internet Source	1%
3	Submitted to Ahmedabad University Student Paper	<1%
4	aran.library.nuigalway.ie Internet Source	<1%
5	Submitted to University of Florida Student Paper	<1%
6	"Computer Vision – ECCV 2020", Springer Science and Business Media LLC, 2020 Publication	<1%
7	Submitted to The University of Manchester Student Paper	<1%
8	www.college-seminars.com Internet Source	<1%

9	Yunseo Hwang, Taeseon Yoon, Kyuyong Park. "GuideDogNet: A Deep Learning Model for Guiding the Blind in Walking Environments", Journal of Student Research, 2021 Publication	<1 %
10	www.wseas.us Internet Source	<1 %
11	Graham M. Seed. "An Introduction to Object-Oriented Programming in C++", Springer Science and Business Media LLC, 2001 Publication	<1 %
12	abdulkaderhelwan.medium.com Internet Source	<1 %
13	huggingface.co Internet Source	<1 %
14	www.slideshare.net Internet Source	<1 %
15	Submitted to Queen Mary and Westfield College Student Paper	<1 %
16	Submitted to Coventry University Student Paper	<1 %
17	github.com Internet Source	<1 %
18	arxiv.org Internet Source	

		<1 %
19	repository.uhamka.ac.id Internet Source	<1 %
20	gcris.iyte.edu.tr Internet Source	<1 %
21	onlinecivilengineeringbooks.blogspot.com Internet Source	<1 %
22	pdfs.semanticscholar.org Internet Source	<1 %
23	powcoder.com Internet Source	<1 %
24	www.antiessays.com Internet Source	<1 %
25	www.scirp.org Internet Source	<1 %
26	"Computer Vision – ECCV 2018", Springer Science and Business Media LLC, 2018 Publication	<1 %
27	Dr. L.C. Monticone. "REAL-TIME LANE AND VEHICLE DETECTION BASED ON A SINGLE CAMERA MODEL", International Journal of Computers and Applications, 2010 Publication	<1 %

28	Tao Huang, Shuanfeng Zhao, Longlong Geng, Qian Xu. "Unsupervised Monocular Depth Estimation Based on Residual Neural Network of Coarse–Refined Feature Extractions for Drone", Electronics, 2019 Publication	<1 %
29	Submitted to University of Surrey Student Paper	<1 %
30	Zhoutong Zhang, Forrester Cole, Richard Tucker, William T. Freeman, Tali Dekel. "Consistent depth of moving objects in video", ACM Transactions on Graphics, 2021 Publication	<1 %
31	acikerisim.iku.edu.tr Internet Source	<1 %
32	citeseerx.ist.psu.edu Internet Source	<1 %
33	hdl.handle.net Internet Source	<1 %
34	open.uct.ac.za Internet Source	<1 %
35	s3.cern.ch Internet Source	<1 %
36	vdoc.pub Internet Source	<1 %