

Pathway to Perception: A Smart Navigation Approach for Visually Impaired Individuals Leveraging YOLO, Faster R-CNN, and LLaMA

by

Tahsin Ashrafee Susmit

20301088

Maliha Mehejabin

20301264

Isratul Hasan

20301072

Azmain Ibn Kausar

20301144

Suraiya Binte Akbar

20301007

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

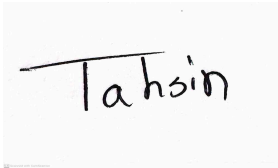
Department of Computer Science and Engineering
Brac University
October 2024

© 2024. Brac University
All rights reserved.

Declaration

It is hereby declared that

1. The thesis submitted is our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.



Tahsin Ashrafee Susmit

20301088



Isratul Hasan

20301072



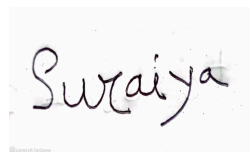
Maliha Mehejabin

20301264



Azmain Ibn Kausar

20301144



Suraiya Binte Akbar

20301007

Approval

The thesis titled “Pathway to Perception: A Smart Navigation Approach for Visually Impaired Individuals Leveraging YOLO, Faster R-CNN, and LLaMA” submitted by

1. Tahsin Ashrafee Susmit (20301088)
2. Maliha Mehejabin (20301264)
3. Isratul Hasan (20301072)
4. Azmain Ibn Kausar (20301144)
5. Suraiya Binte Akbar (20301007)

Of Summer, 2024 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on October 9, 2024.

Examining Committee:

Supervisor:
(Member)



Dr. Golam Rabiul Alam

Professor

Department of Computer Science and Engineering

School of Data and Sciences

Brac University

Program Coordinator:
(Member)



Dr. Golam Rabiul Alam

Professor

Department of Computer Science and Engineering

School of Data and Sciences

Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, Ph.D

Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Ethics Statement

For the avoidance of doubt, the aforementioned study work has not infringed upon any human rights, welfare, or dignity provisions. In addition to cited references from a variety of credible sources, the study incorporates the truthfulness and integrity of the research participants. Neither the data collecting process nor the outcomes have been influenced by any bias or prejudice. We think our effort contributes to the long-term well-being of mankind and demonstrates our respect for all intellectual property.

Abstract

The purpose of our study is to create new technology that will provide a revolutionary navigation system with significant improvement of mobility and independence for visually impaired people. We utilize YOLOv11 and Faster R-CNN to detect an object which is used in combination with Llama 3.2–3B Instruct for context-aware navigation by providing helpful guidance of our current essential location. Our paper tackles the failure points in today’s technologies with lack of flexibility for dynamic and unfamiliar environments, unreliable performance under changes in lighting conditions and inefficient obstacle detection. By training these models together and selecting the one with the highest confidence score, we enhance spatial awareness, identifying obstacles in key areas like the left, right, or center. This approach, complemented by personalized navigation instructions, ensures improved decision-making and safety in real-world scenarios. Using advanced locational technologies available today and imagining those of tomorrow, we aspire to render current navigation methods obsolete by fostering more efficient, real-time and autonomous tools for visually impaired people as they become part of the familiar or unfamiliar environments. After fine-tuning the Llama 3.2-3B-Instruct model, BLEU-4 increased from 0.0442 to 0.1175, and ROUGE-L improved from 0.2102 to 0.3204, indicating enhanced text generation fluency and coherence.

Keywords: YOLOv11, Faster R-CNN, Llama 3.2-3B Instruct, Object Detection, Navigation System, Visually Impaired, Location Detection.

Acknowledgement

We are appreciative to Allah, the Almighty, for giving us the resources and the motivation to finish our studies on time. We would like to express our gratitude to our supervisor, Dr. Golam Rabiul Alam and co-supervisor, MD Saiful Islam, for their crucial guidance throughout the study. Last but not least, we are grateful to our beloved parents who have never left our side and always pray for us.

Table of Contents

Declaration	i
Approval	ii
Ethics Statement	iv
Abstract	v
Acknowledgements	vi
Table of Contents	vii
List of Figures	ix
List of Tables	x
Nomenclature	xi
1 Introduction	1
1.1 Motivation	1
1.2 Problem Statement	2
1.3 Limitation of Existing Literature	3
1.4 Research Contributions	3
1.5 Thesis Organization	5
2 Literature Review	6
2.1 Related works	6
3 Background Study	14
3.1 Faster R-CNN	14
3.1.1 Convolutional Neural Network(CNN) Layer	15
3.1.2 Region Proposal Network (RPN)	15
3.1.3 Region of Interest(ROI) Pooling	16
3.1.4 Anchor points	16
3.1.5 Classification and Bounding Box Regression	17
3.2 You Only Look Once(YOLO)V11	17
3.2.1 Backbone	18
3.2.2 Feature Aggregation:Neck	19
3.2.3 Object Detection and Output Generation:Head	19
3.3 Large Language Model Meta AI (LLaMa 3.2 3B Instruct	20

3.3.1	Pretraining	21
3.3.2	Fine tuning	21
3.3.3	Inference	22
3.4	Low-Rank Adaptation (LoRA)	22
3.5	Google Text to Speech (TTS)	23
4	Dataset	24
4.1	Image Dataset Source and Composition	24
4.1.1	MSCOCO Dataset	24
4.1.2	Primary Dataset	27
4.1.3	Data Annotation Process	28
4.2	Fine-Tuning Dataset Generation for Navigation Sentence	28
4.2.1	Navigating Sentence Generation Dataset	29
4.2.2	Description Generation Dataset	30
4.2.3	Dataset Generation Process	30
5	Methodology	32
5.1	Proposed Model	32
5.1.1	YOLOv11 for Object Detection	34
5.1.2	Faster R-CNN for Object Detection	34
5.1.3	Ensembling YOLOv11 and Faster R-CNN	34
5.1.4	Post-Processing and Combining Detections	35
6	Experimental Evaluation	37
6.1	Experimental Setup	37
6.2	Experimental Configuration	38
6.2.1	YOLOV11	38
6.2.2	Faster R-CNN	38
6.2.3	LLM Configuration and Training Setup	38
6.3	Experimental Findings	39
6.3.1	Evaluation of Faster R-CNN and YOLOV11	39
6.3.2	Evaluation of Fine Tuned Llama 3.2-3B-Instruct	41
6.4	Implications and Results	44
6.5	Discussion	51
7	Limitations and Future Work	54
7.1	Conclusion	54
7.2	Limitations	54
7.3	Future Work	56
	Bibliography	60

List of Figures

3.1	Faster R-CNN Model Architecture	14
3.2	CNN Layer Architecture	15
3.3	Anchor point Architecture	16
3.4	YOLOV11 architecture	18
3.5	LLaMa 3.2 3B Instruct Diagram	21
3.6	LoRA Diagram	22
4.1	Distribution of training and Validation images	26
4.2	Localized Images	26
4.3	Distribution of training and Validation images	27
4.4	Localized Images	28
5.1	Top Level Overview of The Proposed Model Diagram	33
6.1	Loss vs Epoch Curve of Faster R-CNN	40
6.2	Training Results of YOLOV11 on COCO Dataset	41
6.3	Training Results of YOLOV11 on Primary Dataset	41
6.4	Steps vs Loss Curve for LLaMa3.2-3B-Instruct	43
6.5	Output of YOLO and Faster R-CNN Model for Zebracrossing	44
6.6	Output of YOLO and Faster R-CNN Model for Outdoor Stairs	45
6.7	Output of YOLO and Faster R-CNN Model for Indoor Stairs	46
6.8	Output of YOLO and Faster R-CNN Model for Red Traffic Light	47
6.9	Output of YOLO and Faster R-CNN Model for Pole	48
6.10	Output of YOLO and Faster R-CNN Model for Pedestrian Red	49
6.11	Output of YOLO and Faster R-CNN Model for Manhole without Person	50
6.12	Output of YOLO and Faster R-CNN Model for Manhole with person	51
6.13	Comparison with GPT	52
6.14	Comparison with VQA	53

List of Tables

4.1	Training and Validation size of object Class	25
4.2	Training and Validation size of object Class	27
6.1	Model Performance Metrics of YOLOV11 and Faster-RCNN	40
6.2	Llama3.2-3B-Instruct BLEU Score	42
6.3	Llama3.2-3B-Instruct ROUGE Score	42
6.4	Generated Sentence for Zebracrossing from Llama	44
6.5	Generated Sentence for Outdoor Stairs from Llama	45
6.6	Generated Sentence for Indoor Stairs from Llama	46
6.7	Generated Sentence for Red Traffic Light from Llama	47
6.8	Generated Sentence for Pole from Llama	48
6.9	Generated Sentence for Pedestrian Red from Llama	49
6.10	Generated Sentence for Manhole without Person from Llama	50
6.11	Generated Sentence for Manhole with Person from Llama	51

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

BLEU Bilingual Evaluation Understudy

CNN Convolutional Neural Network

FPS Frames Per Second

GTTS Google Text-to-Speech

IoU Intersection over Union

LLaMA Large Language Model Meta AI

LLM Large Language Model

LoRA Low-Rank Adaptation

mAP mean Average Precision

MEMS Micro-Electro-Mechanical Systems

ML Machine Learning

MSCOCO Microsoft Common Objects in Context

NMS Non-Maximum Suppression

PCA Principal Component Analysis

RCNN Region-based Convolutional Neural Network

ROI Region of Interest

ROUGE Recall-Oriented Understudy for Gisting Evaluation

RPN Region Proposal Network

YOLO You Only Look Once

Chapter 1

Introduction

Advancement in technology and our focus on inclusivity have made assistive devices invaluable for improving the daily quality of life for people with disabilities. Using the most recent studies, the World Health Organization (WHO) estimates that the number of people with visual impairment is 285 million. Of these, 246 million have low vision and 39 million are estimated to be fully blind[3]. Although assistive technology and medical treatments have improved, many visually impaired individuals continue to encounter difficulties in navigation without some support from others, especially in more challenging and unfamiliar situations. Their dependence affects their independence and mobility. Few existing navigation systems offer sufficient support, often proving to be too adaptive and not functional in real-time. Current systems face challenges in dealing with varying landscapes and uneven areas. One of the main limitations in current navigation technologies is that they combine object recognition and customized suggestions in ways that are not always effective, making everyday use difficult[17]. We are now using YOLOv11 with Faster R-CNN to assist the visually impaired individuals in a better way. All models are trained together, and the one with better confidence score is selected. Which allows us to know if some obstacles are on the left, right or center. This object detection and spatial awareness combination enables better navigation decisions. We have added Llama 3.2-3B Instruct, an improved version of the Large Language Model (LLM) to increase flexibility in dynamic environments with look-up tables for codebook bindings. This potentially has an effect on the system decision-making which in turn increases its ability to deploy these types of smart interconnected systems.

1.1 Motivation

Blind and visually impaired people are able to live more independent lives even in their predicament with the help of technology which promotes a safer environment as well as easier access to the outside world. That said, there is still a barrier facing for blind people which impedes them in any meaningful way to travel like others instantly and independently causing to their communications with people around the world as well as their quality of life. To address this, we aim to create a smarter and real-time base navigation system compared with any similar navigation systems.

Our research focus has always been toward useful real-world applications to help people. Here are a few motivations for carrying out this research:

- **Empowering visually impaired people to gain independence:** Our system helps blind individuals safely move independently through any given space with 180-degree navigation. It enables people to move independently and navigate difficult, even hazardous places without the help of a human by means of object identification (obstacle avoidance) and real-time guiding.
- **Objective — Detection, Identification & Location:** By employing advanced YOLOv11 with Faster R-CNN the system makes it possible for detecting and identifying location . It also helps users by representing whether the blocks are on the left, center or right side because it guides them properly that impediments are coming from which location so they change their direction immediately.
- **Better Decisions:** This system unlike other systems uses Llama 3.2-3B Instruct for Personalized and Context-Aware Navigation Instructions. Good decisions facilitate the ability to get by with changing circumstances, which in turn provide more refined and reliable guidance.

1.2 Problem Statement

Our research mentions the challenges of ensuring safety and safer navigation for blind and visually impaired people. Despite these improvements in navigational technologies, many blind people and visually impaired people still struggle to move independently when they are in unfamiliar or chaotic environments. One of the major issues have always been the current systems that are poor In object detection, and navigation support. A number of current technologies are unable to react in different lighting conditions or everyday dynamic obstacles that can often obstruct the user’s goal of safely completing a walk.

Additionally, only a small number of object detection models are widely used; however promising results can be achieved with more advanced ones like YOLOv11 with Faster R-CNN which are able to enhance the precision and running speed of navigation systems. Nevertheless, object detection and personalized navigation guidance still pose a great challenge in integrating current technologies that results in disorientation when it comes to human interaction with the environment.

Additionally, many current solutions lack deeper integration with Llama 3.2-3B Instruct and optimization for contextual understanding even if they are using it in some form or another. This type of fine-tuning can drastically increase the system’s capability to give navigation instructions adapted to the user’s concrete environment.

In addition to the high degree of human intervention required, these systems generally have poor performance because they lack a consistent and extensive set of training datasets for more comprehensive environments commonly encountered by visually impaired individuals. These trials highlight a pressing necessity for an integrated, tech-based and real-time assistance typology that works seamlessly to assist people who are blind or visually impaired trust their mobility independently.

1.3 Limitation of Existing Literature

Our research depends on solving many challenges for visually impaired persons. There are several factors that might prevent the study from being conducted. Some of them are:

- **Inadequate Object Classification:** Existing systems fail to classify objects in dynamic environments. This type of limitation can lead to potential risks for visually impaired individuals. Since the developing assistive technologies for the visually impaired focus primarily on classification rather than detection, the challenges lie in accurately identifying objects to ensure user safety and navigation efficiency.
- **Lack of Datasets:** The majority of navigational systems rely on narrow, tiny datasets that fall short of accurately capturing the range of settings that blind people encounter. The issue limits the systems generalizability and applicability in practice.
- **Poor object detection and navigation integration:** It is clear from most of the present systems that there exists a weak logical connection between recognition of objects coupled with navigational guidance. It would slow down the responses and this way end up with an incomplete capability of serving its purpose. Current systems often miss objects in dynamic environments. However, this would likely result in false negatives that could prove harmful to blind or visually impaired end users.
- **Dependency on High-resolution Input photographs:** In order for many systems to perform at their best now we need clear and high-quality photographs while these are hard to obtain in real world severe scenarios like bad weather. That is why the absence of these visuals can lead to an inaccurate navigational aid and a decrease in identification rates.
- **Generic Navigation Directions:** Many of the systems today provide navigation commands that are generic in nature and fail to consider individual user requirements or contexts. This makes navigational systems less effective because different users may require tailored instruction for their particular situations and difficulties. This decreases any use of navigational systems because different users might require customized guidelines as per their situation and problems.
- **Underutilized Advanced Models:** Many of the newest models that yield significantly better object detection and real-time processing (e.g. YOLOv11 or Faster R-CNN) are notably absent from current navigation systems They also not able to take advantage of the next generation technology which could increase their performance.

1.4 Research Contributions

The objective of our study is to design an advanced navigation system that will allow visually impaired people to move to their preferred locations safely and independently. Our work points out the most important problems visually impaired

people have, specially in new environments and proposes how to solve these problems with better technologies. For that purpose, we aim to implement some of the most widely used object detection models such as YOLOv11 and Faster R-CNN in order to improve on the accuracy rate which will result in a better navigation system able to detect an obstacle. It is complemented by Llama 3.2-3B, optimized for context-aware navigation to provide individual instructions with regard to the respective environment. This will enhance decision-making in changing circumstances and provide more appropriate advice. One of the main goals is to seamlessly connect objects detected and personalized navigation instructions, making this process more quick and improving experience. Our mission is to revolutionize independence and movement by challenging the limits of current technological advancements. We want to do this more efficiently, without compromising safety or functionality for visually impaired people.

Some of the most essential contributions that this research has made are as follows:

- We used the MSCoCo2017 dataset. Then we also introduced a custom dataset, as part of our research dedicated to navigation for the visually impaired community, which we have put together from different sources like including the Telstra Pole Detection dataset, Pothole Detection dataset, Stairs Detection dataset, and Traffic Lights and Signs dataset from Roboflow. Initially, this dataset contained 80 classes. Then we removed 3 classes and expanded it by adding 10 new classes to enhance its functionality and applicability. The new classes include: 1) Pole, 2) Zebra Crossing, 3) Pedestrian Green Light, 4) Pedestrian Red Light, 5) Red Traffic Light, 6) Green Traffic Light, 7) Yellow Traffic Light, 8) Manhole, 9) Stairs and 10) Bus Stop. By doing so, this dataset can help to bridge the gap of resources available for training and evaluating models that support users navigating unfamiliar environments.
- Our study shows an innovative approach for object detections using models like Faster R-CNN with YOLOv11. Using these models we can improve the performance in terms of accuracy in which obstacles are detected, ensuring that a visually impaired individual is provided accurate navigational guidance on time based on what he/she sees around.
- We introduce Llama 3.2-3B Instruct, optimized for context-aware navigation as well in order to provide individualized and user-specific guidance. It makes the system more resilient to changes and improves overall user experience.
- Experimental results confirm that the proposed model performs better compared to existing systems targeted towards this kind of navigation assistance in a complex environment. The presented study is expected to result in significant enhancements of safety, mobility and independence of visually impaired individuals during daily life routines by demonstrating the superior performance obtained using our novel dataset combined with advanced object detection techniques when implementing Llama 3.2-3B Instruct efficiently.

1.5 Thesis Organization

Our thesis is organized as follows: To address the aim of this paper, chapter 2 presents several literature reviews dating back up to earlier years detected but also some current ones found on object detection. A detailed history of the evolution of our work paradigm is provided in Chapter 3. Chapter 4 contains all the information associated with datasets that we actually used to conduct our research. Chapter 5 provides more details of our Learning System. In this part we have displayed how the models are constructed, what changes made to get the right parameters of a model, How the resultant is generated and the whole workflow(criteria) as shown below. The primary segments of technology required to facilitate our research have been described in the beginning section of chapter 6. This is followed by qualitative and quantitative comparisons of our model with existing models to show that it can achieve better performance. Chapter 7 addresses Limitations and Suggestions for Further Research. Chapter 8 shows results, implication and conclusion of our Study. Chapter 9 is about the summarization of our thesis.

Chapter 2

Literature Review

Smart navigation systems for visually impaired people have made attention over the years. Technology has made major progress, with advancement in object detection models and (LLMs)-Large Language Models enabling new approaches to enhance navigation assistance. These systems not only increase the individual independence and safety of visually impaired individuals, but also provide them with guidance that takes into account different environments. These are YOLO, Faster R-CNN etc., and LLMs for use in building efficient navigation systems through some literature reviews.

2.1 Related works

In the paper, You Only Look Once: Unified, Real-Time Object Detection, the author doesn't follow traditional methods which often rely on classifiers for detection tasks. When implemented for object detection framework YOLO attempts to frame the problem of detecting objects in images as a single regression process that predicts bounding boxes and class probabilities directly from full image in one evaluation. By this new approach, a single neural network runs at full speed on the images in 45 frames per second for the entire base model and up to 155 frames per second with Fast YOLO. Although the architecture of YOLO enables end-to-end optimization that is directly associated with detection performance, it produces more localization errors than methods like R-CNNs. Additionally, YOLO presents a lower false positive rate on non-object regions than the conventional object detection algorithms which makes it more practical. Although this model has better accuracy than prior models, the localization errors it produces are not insignificant and may be problematic in situations where high precision is required which is an important area for future research. Though YOLO models have made a large improvement in real-time object detection, there are some constraints which need to be surpassed to improve accuracy [5].

The paper, "Embedded Implementation of an Obstacle Detection System for Blind and Visually Impaired Persons' Assistance Navigation", proposes a system using an improved version of the YOLO v5 neural network to solve navigation problems encountered by individuals with visual impairment. By integrating DenseNet in YOLO structure and improving the full-network both speed wise as well detection

accuracy, this system is able to run at 43 frames per second (FPS) and achieve an impressive accuracy of 83.42%. The pruning and quantization of the channels were used to make it possible as embedded implementation in a ZCU 102 board's system. The authors point out limitations such as the difficulty of detecting a set of diverse obstacles within dynamic-cluttered scenes which might affect the system's general robustness and adaptability. In the future, further research and development are required to improve system robustness under real-world conditions for helping BVI users in a reliable way [18].

In the paper "Enhancing Surveillance Systems with YOLO Algorithm for Real-Time Object Detection and Tracking" the author attempts to develop an object detection system for assisting Visually Impaired Persons (VIP), who cannot identify or detect objects in the environment. Based on web technology, a study was done with the use of YOLO (you only look once) for accurate object prediction with a deep learning approach. The classifier that is trained on 500 images achieves an accuracy rate up to 94%, meaning it can perform significantly better than traditional machine learning or image processing techniques. Nevertheless, to develop a system that can be even better in other real-world settings or more robust against unseen changes adaptation will require future research [9].

In the paper Real Time Object Detection using YOLO Algorithm, authors made use of the You Only Look Once (YOLO) method for object detection and checked its efficiency compared to the leading models in speed and performance. Whereas traditional algorithms might scan regions of an image through multiple forward and backward propagations, YOLO architecture performs a single evaluation that first predicts bounding boxes and associated class probabilities from features maps using logistic regression itself. This has the benefit of speeding up the detection process drastically which can be applied to real-time applications. Thus the research correctly showcases benefits of YOLO which is quick and accurate, thereby proving it to be a good alternative solution in scenarios like autonomous navigation etc. However, it limits its effectiveness and needs more work to enhance the robustness of YOLO under a general case like localization error in complex environments [16].

The study YOLO-Based Models for Smoke and Wildfire Detection in Ground and Aerial Images focuses on improving UAV-based object detection, specially for small objects, by optimizing the YOLOv8 model for aerial photography scenarios. Intended for application in UAV detection systems, the newly proposed model named UAV-YOLOv8 aims to solve a recurring challenge of low accuracy due primarily to resource constraints. Using Wise-IoU (WIoU) v3 for accurate localization, an attention module called BiFormer to increase the focus on salient information and a feature processing unit known as Focal FasterNet block (FFNB), which helps unify features by leveraging multiscale representations. Experimental results show that UAV-YOLOv8 achieves mean detection accuracy up to 7.68% higher than the baseline, meanwhile better in detecting small objects. A few seconds later, the study indicates that detection may be more difficult for small, featureless objects (like a

bike-like vehicle) and requires further optimization [22].

In the paper Real Time Object Detection using YOLO Algorithm, the paper provides a comprehensive overview of the benefits that can be derived from the YOLO approach when it comes to object detection, and how these results are compared with existing algorithms, especially Faster R-CNN. While Faster R-CNN achieves great results on both detection accuracy and speed it still needs to well-handle a massive amount of negative samples so it may be not that robust in real application. The authors suggest two approaches to address this problem: hard negative sample mining and alternating training. In other words, hard negative sample mining is to find those tough to predict negative samples and use them for model updating in order to improve the classification. Meanwhile, alternating training allows the Region Proposal Network (RPN) and Fast R-CNN components to share convolutional layers, promoting a more cohesive learning process rather than treating them as independent networks. Simulation results show that this integrated method considerably enhances the detection precision and spotlight its utility in real-time applications such as surveillance, autonomous systems, etc. Moreover, the paper also suggests that further tuning may eventually be required to overcome remaining algorithm complexity and computational needs limitations [10].

In the paper "An Improved Faster R-CNN for Small Object Detection", authors introduce methods to overcome difficulties in detecting small objects under complex scenes by using convolution neural networks(CNN). This paper proposes a refined Faster R-CNN based algorithm for small object detection. In this method, a two-stage detection strategy is used. Improved loss function based on Intersection over Union (IoU) is introduced for bounding box regression during the positioning stage. Moreover, bilinear interpolation is applied to further improve the region of interest (RoI) pooling operation for precise localization. In the recognition stage, multi-scale convolutional feature fusion helps to supplement the feature map with more information and a modified Non-Maximum Suppression (NMS) algorithm is utilized in order to reduce overlap object losses. The results show that in the (0, 32], proposed method has a recall rate as high as 90% and an accuracy rate of up to 87%, which is far better than the original Faster R-CNN. The effectiveness of these enhancements motivates the further study on object detection frameworks and provides us a valuable way to consider handling small objects for them. However, further studies remain to be done on the adaptability of this algorithm to different environment and object types [11].

In the paper "Accurate Object Classification and Detection by Faster-RCNN", the authors show how to combine Region Proposal Networks (RPN) with Faster R-CNN for high quality object detection faster. They concentrate on utilizing a joint network that can let the RPN predict region proposals, which are later sent to Faster R-CNN for classification instead of incorporating externally RIO extraction and reducing your computational overhead. They trained their VGGNet architecture on two popular datasets (PASCAL VOC 2012 and MS COCO) only using a low cost

GPU, and they explored the performance change as the region proposals increased. Their findings reveal an increase in the Mean Average Precision (mAP), with improvements peaking at 1.2% for 1,800 proposals before reaching saturation around 2,000 proposals. In this paper, it highlights the efficacy and precision of algorithms proposed in comparison with other methods. Yet, scalability of the approach to larger datasets and more diverse object classes is an area that should be further investigated [8].

In the paper "A Closer Look at Faster R-CNN for Vehicle Detection", the authors investigate the application of the Faster R-CNN algorithm to vehicle detection, noting its initial unimpressive performance when directly applied to large vehicle datasets. After much trial and error, the authors detail their study on model architecture as well as tweaks to parameters and algorithms. Due to their modification the model performance gets greatly enhanced; the most competitive result is achieved on the KITTI vehicle dataset. This work provides a good reference point for others to use in optimizing Faster R-CNNs further and shows the benefits of different kinds of designs used on object detection tasks. However, using parameter tuning to reach for this additional performance may also indicate difficulties in achieving robust performance across diverse datasets without the level of customization needed [4].

Based on Faster R-CNN, In this paper, "Face Detection With Different Scales", the authors dealt with detection of small faces because most existing detectors suffer from severe feature map shrinkage in deep convolutional networks. They proposed Different Scales Face Detector (DSFD) that achieves better accuracy on face detection in real time manner, preserving the efficiency of their model by using multitask RPN for region proposal and modifying anchor generation from single-size to multi scales at different facial landmarks. The DSFD consists of three parallel Fast R-CNN networks customized to different proposal scales, utilizing various strategies such as multitask learning and feature pyramids to improve detection accuracy. Experiments on multiple benchmarks such as FDDB, WIDER FACE demonstrate that the proposed method outperforms techniques like UnitBox and HyperFace. However because its architecture is quite complex it would be difficult to implement in real-time application [14].

The paper, "An Automated Navigation System for Blind People", addresses the challenges faced by visually impaired persons. The proposed system incorporates ultrasonic sensors for obstacle detection and stereo camera to capture an image of the surroundings, based on deep learning algorithms that will achieve better environmental perception. The software also includes face recognition, to determine if it knows people in the images, as well as speech recognition for user input. Furthermore, the system is a cloud-based storage to store information and comes with web and mobile applications that can trace users in real time where parents or guardians or others may respond during emergencies. Experimental results demonstrate that this system as a complete accessible solution supplies detailed environmental intelligence and friendly interaction manners, which can greatly benefit the navigations of

visually impaired people. Nonetheless, comprehensive technology dependence may be harmful with respect to accessibility and generality in different application areas [17].

The study "Smart Eye: A Navigation and Obstacle Detection for Visually Impaired People through Smart App", aims to solve the largest hurdle which is faced by blind people in navigating an unfamiliar environment as reported by WHO, who states that globally over 283 million people are visually impaired. Their intended product model seeks to improve the everyday living of visually impaired persons (VIPs) by providing low-cost, lightweight assistive devices utilizing artificial intelligence and sensor technology. Ultrasonic sensors will detect obstacles that occur in the way of the user and a smart app takes photos to categorize them. It gives voice commands to run alerts, bringing the interface and allowing users to identify imminent threats. The proposed system provides qualitative and quantitative improvements in performance, providing a means of independently navigating through groups of VIPs. The study does, however, note that current navigation aids are not always used efficiently because of their expense and complexity, which the researchers say supports a push for less prohibitive devices [19].

The Paper, "A Review of Navigation Systems (Integration and Algorithms)" analyses the latest developments in navigation technology with particular reference to GPS/INS integration. This integration has greatly increased the accuracy of navigation, performance and overall cost-effectiveness. The paper review focuses on the performance evaluation of these systems and reported improvements for optimizing their operational efficiency. Yet, the majority of them only regard high-precision Inertial Measurement Units (IMUs) that are dedicated to inertial sensing without direct navigation outputs. The research paper suggests further investigation in this field to improve the overall navigation solutions with respect to cost and efficiency [28].

Over the years, autonomous navigation systems have evolved from traditional methods involving dead reckoning to based on an amalgamation of GPS technology, Inertial Navigation Systems (INS), LiDAR and Radar; which drastically raises precision levels in relation to navigating different environments. The real-time data on direction, speed and position are critical for navigation carried out by integrated sensor systems. Yet limitations remain, partly due to expensive advanced sensor technologies, vulnerability of sensors in changing climatic conditions that can reduce performance accordingly and complication from the need for feature level fusion which increases computation requirements potentially compromising system reliability. As a result, while improvements in autonomous navigation are encouraging evidence to support the feasibility of this vision, they also highlight the many challenges that are yet unsolved and need solving for greater scale-up until they can be practically used everywhere [1].

Inertial navigation sensors specially Ring Laser Gyros (RLGs), Fiber-Optic Gyros (FOGs) and Micro-electro-mechanical Systems (MEMS)-based gyros and accelerometers has revolutionized navigational applications by providing smaller, rugged solutions while retaining high-performance capabilities. That includes the ability to bake navigation tech into new areas previously considered impractical like personal navigation devices and artillery systems. RLGs and FOGs have matured technologically, but there are continuing developments which may make MEMS practical for commercial or tactical three dimensional gyroscopes. As good as MEMS sensors are, they can't meet the performance of RLGs and FOGs and high-performance MEMS require more exotic materials which isn't conducive to cost-effective mass production. The future of it almost certainly involves smaller sizes and better stability (cold atom interferometry is an example tech that can be expected to deal with a few limitations) [2].

Image captioning is an interdisciplinary research field that bridges the gap between computer vision and natural language processing by learning models that integrate top-down semantic attention with bottom-up information in image representation. The selective focus on semantic concepts in this innovative algorithm enhances the output quality from recurrent neural networks. Quantitative evaluations on Microsoft COCO and Flickr30K datasets show that this approach significantly outperforms several technologies. On the downside though, it does not do object detection, and is not a good way to provide accurate precise locations of objects in complex scenes. Additionally, the prerequisite of large annotated data sets, that are not always available, emphasizes the further need for research.[6].

The study proposes Object Relation Transformer, for generating image captions that builds on top of traditional encoder-decoder architectures by adding geometric attention to model spatial relationships between detected objects. The related experiments on the MSCOCO dataset demonstrate that, by leveraging object detector-based feature vectors as external guidance signals in a simple parallel module, this solution achieves substantial performance gains over image captioning benchmarks according to widely recognized figures of merit. However, the geometric attention model has limitations in terms of complex scenes with many objects congested together. The drawback of this constraint is that it reduces its ability to capture all relevant items appropriately and give accurate spatial information, hence the model needs future research for strengthening performance in such situation. [12].

The paper introduces the Attention on Attention (AoA) module, which improves existing image captioning architectures by mitigating their well-known shortcomings of traditional attention mechanisms. The main difference is that AoA helps to get the over information needed for decoding better connected with query. This module will be integrated in the proposed AoA Network (AoANet), on both encoder and decoder, which outperforms previous methods and achieves superior performance when evaluated with MS COCO dataset. While it is true that these make this model very hard to use across datasets generally, they are also its biggest challenge. The

ability of a neural network system to generalize beyond training set size. Besides, AoANet does not provide object location information. It only gives a salience score of the attended region, so generated captions are very high level but lack specific context. The speed of AoANet for real-time applications should be improved so that it is relevant when considering more rapidly evolving environments.[13].

In their paper "Intelligent LiDAR Navigation: Leveraging External Information and Semantic Maps with LLM as Copilot", Xie Zhang Schwertfeger introduce a new take on robot navigation which combines Large Language Models (LLMs) with traditional occupancy grid maps and laser-based sensing approaches. The research aims to improve robotic navigation systems with some latent contextual understanding similar to human cognition by exploiting osmAG and a fabulous topometric hierarchical map representation. With this integration, robots can leverage external information and experiential knowledge from requests to other robotic services like elevator maintenance updates for better navigation efficiency. But as useful and receptive as these maps are, a big hindrance is that the need to use it for the robot to get anywhere which brings up potential obstacles when in real-world situations where a rather dated map can be practically worthless. Furthermore, when the events reach LLMs to examine their paths and if the traffic is examined a lot, real-world areas that are not directly affected nearby or outside these zones (the distance after them) leads to some problems. Authors argue that addressing these problems by using LLMs in path planning can prevent the system from being too careful, and ensure all available passages are recognized to improve navigation results[27].

In the paper ATLAS, an autonomous navigation system that transcends conventional immobile navigational agents solely based on static maps and predefined landmarks. ATLAS expands a navigable landmark set through an object detection module and a path planning unit that is combined with Transformers for Natural Language Understanding (this task-based Transformer, or TB-Transformers). This creative technique enables the model to reason abstractly at a high level and elevates agent performance in challenging navigational tasks across different simulator domains. But the technique has had struggles with navigational accuracy when it tries to recognize and localize objects in real-time, which constricts path planning. Moreover, it may have a sensation of difficulty in performing navigation as result never been exposed to environment changes since system mostly based on dynamic landmark acquisition[24].

This article presents a new approach for household service robots and similar multi-agent groups to communicate using massive language models (LLMs) as part of the unit design. In this paper, a communication-triggered dynamic leadership structure is incorporated, which can balance team consensus and navigation effectiveness under lower cost of inter-robot communications. This novel approach highlights the ability of LLMs to cooperatively traverse intricate terrains. The study thus has scalability limitations, as team size grows it becomes harder to communicate effectively. At the same time coordinating multiple people in busy environments can be

susceptible to lead frequent breaks and chances of impact on overall navigation efficiency, task completion or may even cause clashes between each other due to lack or confusion if appropriate measures are not taken into account for coordination[26].

Chapter 3

Background Study

Our research work is divided into two parts which are object category identification and object localization. We combined LLaMA 3 with YOLOv11 and Faster R-CNN in the work. To improve image recognition and localization speed, we modified these models. For an example, We have implemented Principal Component Analysis (PCA). It can make the extract features smaller and efficient at the same time. Their modifications led to higher results in accuracy as well as in peaked effectiveness.

3.1 Faster R-CNN

Faster Region-based Convolutional Neural Networks (Faster R-CNN) represent a significant advancement in object detection. It is combining both object localization and classification in a single architecture. Faster R-CNN developed by researchers at Microsoft and this model integrates the generation of region proposals with the object detection network itself. Unlike its predecessors which is R-CNN and Fast R-CNN, it does not rely on external methods for generating region proposals.[21]

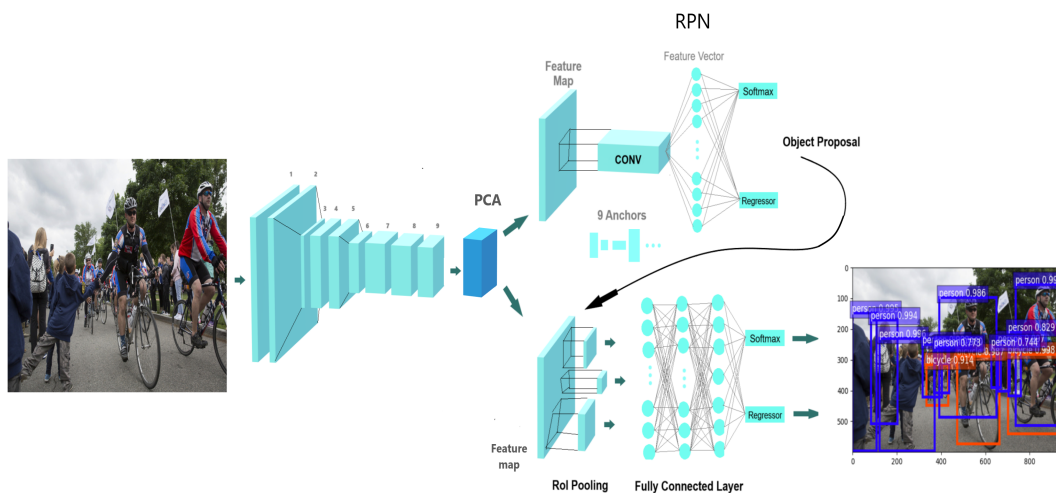


Figure 3.1: Faster R-CNN Model Architecture

The key innovation in Faster R-CNN is the use of a Region Proposal Network (RPN), which significantly speeds up the process of identifying objects by sharing computation with the convolutional layers. It also has other key components like

ROI pooling, anchor boxes, Softmax classification, and bounding box regression contribute to its success in tasks of object detection and image classification.

3.1.1 Convolutional Neural Network(CNN) Layer

The Convolutional Neural Network (CNN) is the backbone of Faster R-CNN, as it used for feature extraction from input images. The input is processed by a CNN with several layers of convolutional operations. It captures virtually all the important spatial and hierarchical features, such as edges and textures in an image or photographic pattern information. The information in these feature maps must be significant if work is to be done transformer processes down-stream to make the region proposals and learn about classes. In Faster R-CNN, well-established CNN architectures such as VGG-16 or ResNet are often used as feature extractors. These

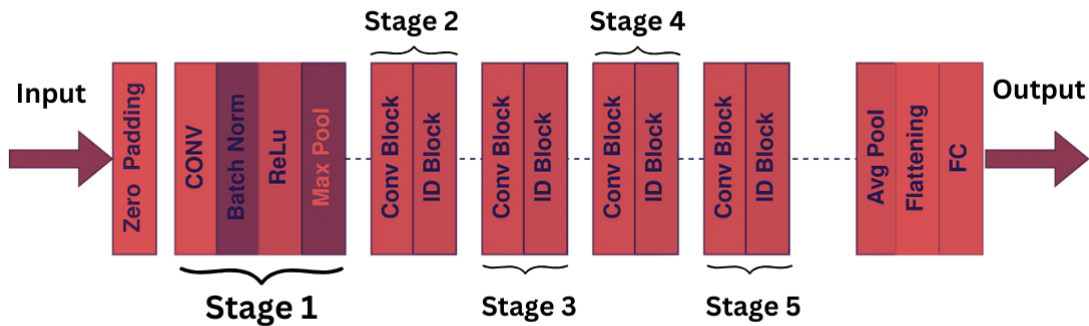


Figure 3.2: CNN Layer Architecture

feature maps created by the CNN are important in the Faster R-CNN system as a whole since they are passed over to the Region Proposal Network (RPN), which uses them to suggest promising places for objects to appear. These feature maps serve as inputs for both the Region Proposal Network (RPN) and the classification and bounding box regression components, underscoring the pivotal role of CNN in the Faster R-CNN architecture.

3.1.2 Region Proposal Network (RPN)

The key underlying innovation of Faster R-CNN is the Region Proposal Network which is used to generate region proposals that integrate directly in the object detection process. This replaces traditional methods such as Selective Search with a trainable neural network. For this component Faster R-CNN is different from its predecessors model. Due to this key innovation, Faster R-CNN have a faster speed. At each chunk of location, the Region Proposal Network (RPN) slides a window over feature maps generated by CNNs that have been trained to recognize objects. Region proposals are represented by pre-defined anchor boxes over different scales and aspect ratios to cover different object sizes.

A region proposal will have at least one anchor box associated with it. Each anchor box is also associated with an objectness score, which serves to identify if a region is an object or background. Moreover, the slide would result in the coordinates

of the anchor box not aligning properly with the actual object. Therefore, the output results of RPN are as follows: ‘(probabilities of the region being object, coordinates of the anchor box)’. Finally, the results are put through the Non-Maximum Suppression to filter out redundant and overlapping region proposals.

3.1.3 Region of Interest(ROI) Pooling

The Region of Interest Pooling layer is important when processing region proposals of varying size generated by the RPN. The ROI Pooling layer transforms the regions by organizing them into a fixed-size feature map. Meanwhile, in the ROI Pooling stage, each region proposal is divided into a set of cells (equally-sized grid). This is followed by max pooling, which applies individually to each cell and picks the highest value in that area. In the second step, features and objectness are extracted from each resized region to produce a fixed-length representation for every proposal. This transformation helps to maintain the processing uniformity after layers and it is indeed very helpful in proper object classification along with an accurate bounding box prediction[7]

These are then pass through Classification and Regression layers, which produces a fixed length representation for predictions that can handle proposals of variable shapes and sizes.

3.1.4 Anchor points

Another important concept in Faster R-CNN is anchor boxes, they are used to detect objects of various shapes and scales. RPN uses these hardcoded boxes as the anchor points, and from this point onward predicts object presence in other frames. At every position on the feature map, we generate multiple anchor boxes at different scales and aspect ratios.

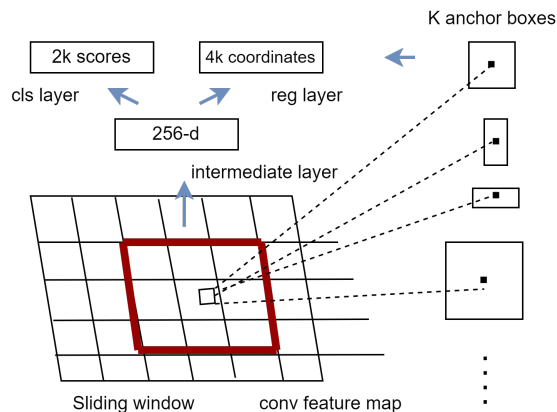


Figure 3.3: Anchor point Architecture

During anchor matching, anchors are assigned with IoU thresholds to be the positive or negative samples. An anchor is a positive example if its IoU with any ground truth box exceeds 0.7, otherwise it is negative which means no object. Anchors with low overlap, generally less than 0.3, are negative anchors or considered as backgrounds.

$$\text{Objectness}_{\text{score}}(\text{IoU}) = \begin{cases} \text{Positive} \rightarrow \text{IoU} > 0.7 \\ \text{Positive} \rightarrow 0.5 < \text{IoU} \leq 0.7 \\ \text{Negative} \rightarrow \text{IoU} < 0.3 \\ \text{Not Negative/Positive} \rightarrow 0.3 \leq \text{IoU} \leq 0.5 \end{cases} \quad (\text{i})$$

This technique helps the model learn to localize where different objects appear in an image and can provide a hint of the most object-containing area vs non-object region, which instructs the network for better predictions. This anchor-based mechanism enables RPN to predict and refine location of objects which can have different lengths rather than localization assuming standard single configuration.

3.1.5 Classification and Bounding Box Regression

After the region proposals have been passed through the ROI Pooling, the feature maps are passed to two parallel layers. One of the parallel layers is responsible for classification, and the other is responsible for bounding box regression. The two layers are as shown below: The regions are passed for softmax classification, and each class is assigned a probability distribution including the background class. The class with the maximum probability is chosen to be the prediction. This layer assigns to each class (including the background) a probability. The probabilities are used to determine the identity of the object contained in the proposed region. The class that has the maximal probability is usually used as the prediction. Bounding box regression adjusts the proposed bounding boxes for the image. The regression layer refines the anchor coordinates by predicting some offset values. This kind of classification and bounding box regression helps the faster R-CNN to classify and give the accurate location where the object is.

3.2 You Only Look Once(YOLO)V11

The introduction of the YOLO (You Only Look Once) marked an important leap forward in real-time object detection with high accuracy. The newest iteration is YOLOv11, which further enhances the detection speed and precision as well more complex object features. YOLOv11 is a neural network for multi-class object detection in an image using deep learning techniques, and it processes the images directly to identify objects without multiple passes. This kind of approach makes YOLO very fast even when accuracy requirements are too high as well. The unique backbone, innovative anchor-free mechanism and the support for multi-scale feature detection are major reasons that make this model performance with high accuracy.[23]

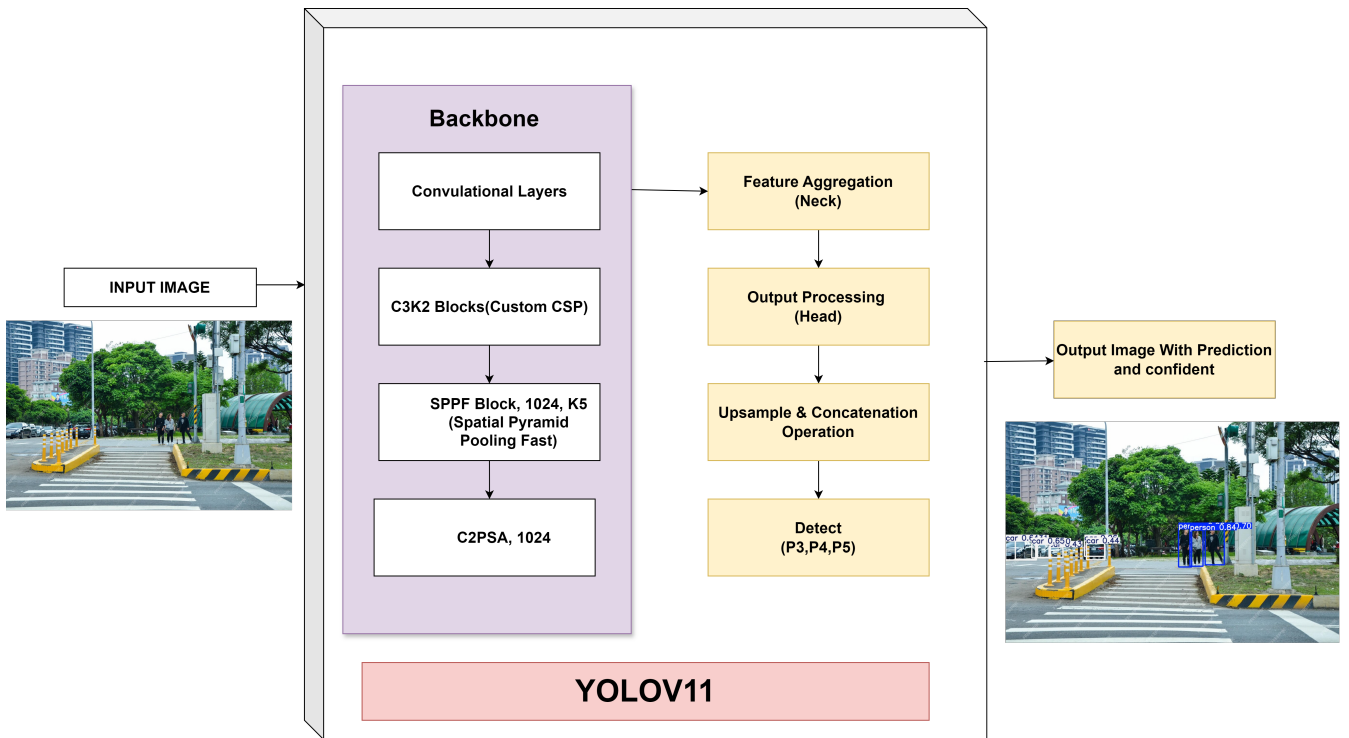


Figure 3.4: YOLOv11 architecture

YOLOv11 has a pretty straight-forward detection pipeline consisting of several key components. These are the steps including Feature extraction, Feature Aggregation, Output Processing etc. At each of those steps, YOLOv11 enhances its ability to infer objects more efficiently and accurately. This guide will go through each process.[20]

3.2.1 Backbone

YOLOv11 consists of different stages, the first one is Backbone which has many processes and which extracts fundamental features from input image. Here it has few composite and technical blocks

Convolutional Neural Networks (CNN)

These are the convolutional layers at the start of the backbone. Each layer learns simple visual patterns like edges, shapes, and textures, while subsequent layers learn more complex features as the data moves deeper through the network. This step is really important for understanding the image content.

Cross Stage Partial Networks (CSP)

CSP blocks are used in YOLOv11 to enhance the efficiency of feature extraction. CSPs divide the input feature map into two branches and only one of them is processed while the other one is passed through. Merging the two paths after processing This method helps reduce repetitiveness, lowers the processing burden and provides a wider range of extracted characteristics. The use of this technique also enables

the model to capture more complex relationships but without going beyond the processing capacity of the system.

Spatial Pyramid Pooling (SPP)

One difficulty in object detection is that we have to conduct image classification using the input images of various sizes, and it is also necessary to perceive the instances with different scales in a common integral image. To mitigate the above, the SPP block pools features from multiple kernel sizes which enables different receptive fields to be captured. This ensures that YOLOv11 can effectively extract multi-scale features from images enabling the network to perform good detection for smaller parts and at larger scales.

CSP with Spatial Attention Mechanism (SAM)

SAM is combined with CSP blocks to take advantage of the capability of backbone. SAM allows the model to pay more attention to only the relevant regions of an image by bringing the most distinguishable areas in focus which is essentially where one or more objects are likely to be present. Focusing on these regions helps the model to increase accuracy in detection without overloading computation resources.

3.2.2 Feature Aggregation:Neck

After the backbone extracts features, in neck, it processes the features to be ready for detection at final stage. One of the key responsibilities of the neck is to combine features at different levels from the backbone, so that we do not lose important details and context. There are a few operations that make up the neck:

Upsampling

The neck first upsamples the feature maps, in other words, it increases the resolution of some of the feature maps. It enables features from lower-resolution layers to participate in comparison and aggregation with those of higher-resolution layers. It is important for preserving fine detail which can be lost thru pooling when trying to detect small objects, so upsampling becomes more evident here.

Feature Concatenation

The neck combines features learned across multiple scales via a simple concatenation following upsampling. It is suggested that this concatenation operation works because it enables the model to use information from different layers of the neural network and allows for a more informative representation of feature maps. This feature fusion across multiple scales allows the model to better recognize objects that appear in various sizes and parts of the image.

3.2.3 Object Detection and Output Generation:Head

The head layers are the last part of the YOLOv11 architecture that we will be covering and this is where we get detection results out from our model. Employing the feature extracted multi-scale features via neck, head utilizes it right to detect

objects and generate bounding box. It is formed with a few component that are specialized in image detection, both accurately and efficiently

Detection Blocks

The V10Detect blocks, which act as a core of the head that processes feature maps to create predictions. These blocks will also execute multiple convolutions to identify objects, classify them and infer bounding box coordinates. Fine tuning of size and shape discrimination on each detection block allowing multiple objects to be detected in an image frame.

Output Processing

After generating predictions with the V10Detect blocks, the head will take the outputs and produce final detections . This means taking non-maximum suppression to remove overlapping bounding boxes, getting only the best detections. This helps to determine if something is identified as the target object in a simpler and uncomplicated manner, the result will be the bounding boxes of objects with their class labels and confidence scores surrounding them.

3.3 Large Language Model Meta AI (LLaMa 3.2 3B Instruct

LLaMA is a family of language models developed by Meta, formerly known as Facebook. The model was released in 2023, and it has shown strong performance on multiple benchmarks, the majority of them being the open-domain tasks. Despite being smaller than most large language models like Generative Pre-trained Transformer , it has achieved excellent results. Such a characteristic is beneficial if one does not have enough computational capacity and wants to work with these model[25] LLaMA is a large language model based on the Transformer architecture. Originally developed as a structure to support massive parallel processing of sequence data, Transformers have become the underlying architecture of a growing number of very big models primarily due to the self-attention mechanism it uses to effectively capture long-distance dependencies in text. Another important characteristic of LLaMA is that it is pre-trained in an unsupervised manner on the vast corpus of internet text and then fine-tuned on specific tasks. The model is available in different sizes to suit the researchers' requirements and computation resources.

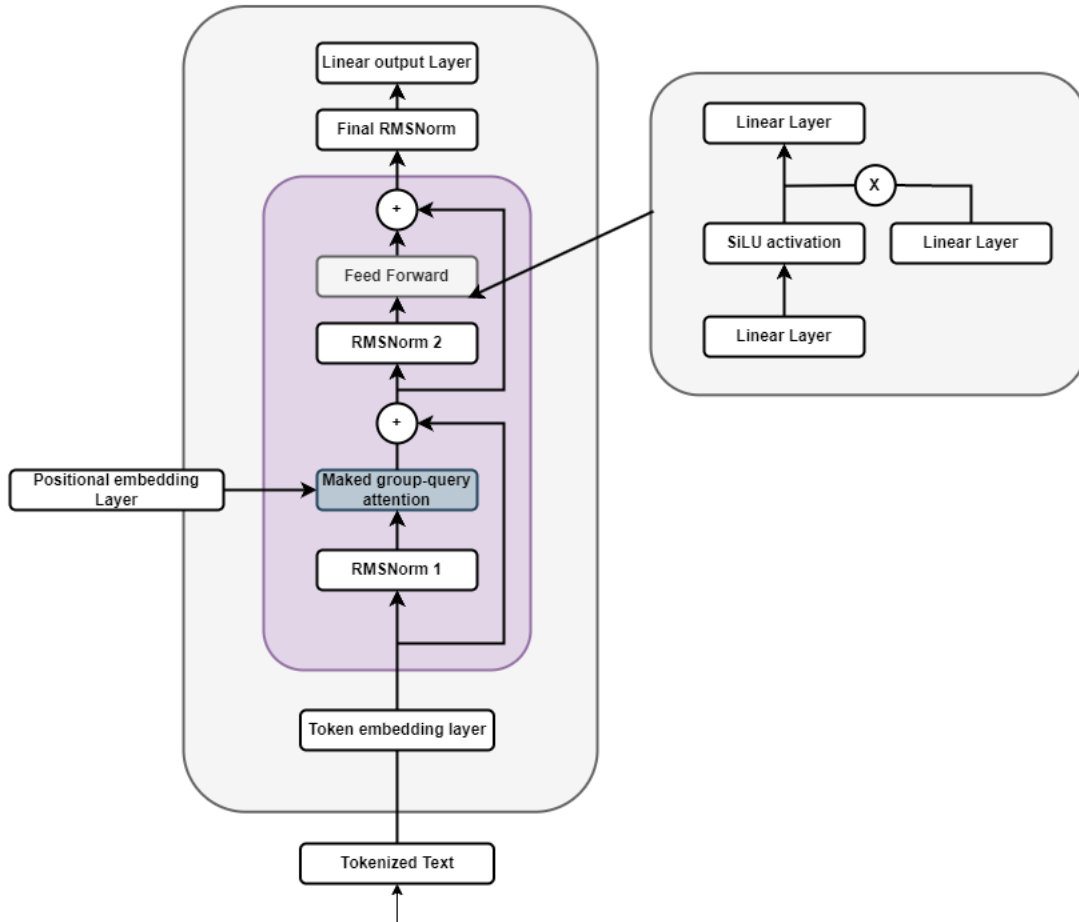


Figure 3.5: LLaMa 3.2 3B Instruct Diagram

3.3.1 Pretraining

In the first phase, LLaMA is pre-trained on a huge dataset that usually consists of text. More recently, multimodal pre-training has been introduced, including both text and images. In this phase, the model is trained to predict the next word in a sentence, learning about the fundamental properties of language, such as linguistic structures, semantics, and context. The LLM we are using is LLaMA 3.2 3B Instruct which was developed for broad language comprehension through extensive large scale pre-training by meta.

3.3.2 Fine tuning

To complete the essential training phase, LLaMA must next be fine-tuned with more data and used additional data during pre-training. Fine tuning involves retraining a model with more specific data relevant to a particular task. Examples of task training with fine-tuning are training for text classification, image captioning, or translation. This is the last stage of the LLaMA training, as in this stage LLaMA updates its parameters and trains the model more for the purpose of performing better in a particular domain.

3.3.3 Inference

This stage pertains to the process whereby the LLaMA model is trained or fine-tuned and becomes ready to generate inferences. In other words, it is the part of model use when it produces responses or predictions relative to the received prompts. For instance, given an input photo or sentence, the model can be requested to complete or generate a description, provide reasoning on a topic, or others. The previous experience on pre-training and fine-tuning is used to make sure that inferences provided by the model are within an appropriate context and precise, i.e. achieving the best performance.

3.4 Low-Rank Adaptation (LoRA)

LoRA is a technique designed for more efficient fine-tuning of large language models such as LLaMA. The main principle underlying the operation of LoRA is a reduction in the total number of parameters that require training by implementing low-rank matrices through specific model layers. It is worth noting that the given approach allows practitioners to both accelerate and reduce the overall computational and memory demands of the model. Apart from that, the technique has shown to help improve the overall performance of models while maintaining comparable accuracy results[15].

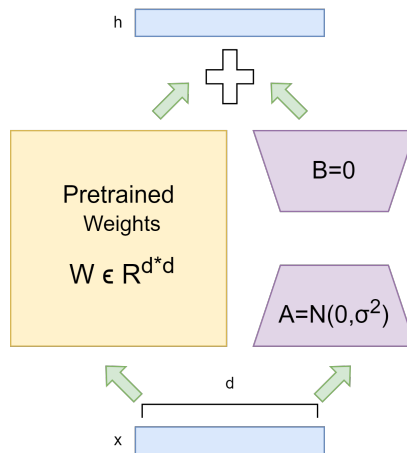


Figure 3.6: LoRA Diagram

LoRA applies low-rank decomposition matrices to the attention layers of a pre-trained model, such as LLaMA. Instead of updating the full weights, this model inserts additional matrices that are lower in rank. These matrices are responsible for catching the task or context-specific adaptations of a model and can update

during the fine-tuning process without the need for full re-training or a substantial degree of memory usage.

3.5 Google Text to Speech (TTS))

When we had the navigation sentences generated by detected objects and spatial locations, they were with Google Text to Speech (TTS) in text form converted into audio representation. Google Text-to-Speech is by far the most reliable and efficient way to synthesize text into human-like speech necessary for guiding visually impaired individuals in real time. This is applicable in the case of providing textual directions to Audible Book, with my use case for an automation around converting text-based navigation instructions into audio. This was done using GTTS library. Each sentence, for example "You can cross at the zebra crossing on your right" or "Proceed to stairs in center," has been synthesized into mp3 voice files. Easily integrated into navigation systems because they omit a spoken feedback The speech is clear and sounds human-like when produced with Google TTS, allowing easy to comprehend step-by-step directions for real time notification of the correct direction.

Chapter 4

Dataset

This section outlines the datasets used in our thesis, how they were prepared, and the methodologies employed for annotating and enhancing object detection for the visually impaired. Our dataset is made up of two major sources: the MSCOCO dataset and an additional set from Roboflow for new object classes.

4.1 Image Dataset Source and Composition

4.1.1 MSCOCO Dataset

The MSCOCO dataset is a widely recognized dataset for object detection. It comprises 80 object classes. Three of the classes were removed for this paper as they are not relevant to the task of guiding a visually impaired person. Since the MSCOCO dataset has a broad variety of images, contains rich annotation, and is strongly supported by the developmental community, it makes more sense to use it in application, as it is efficient for developing better models. The MSCOCO dataset provided around 116081 training and 4900 validation images, each with annotated bounding boxes and object locations for multiple objects. This large dataset facilitates model training for object detection and localization in different conditions, resulting in better guidance systems that can be employed to help visually impaired individuals with navigation.

Table 4.1: Training and Validation size of object Class

Class	Training	Validation
Person	262465	11004
Bicycle	7113	316
Car	43867	1932
Motorcycle	8725	371
Airplane	5135	143
Bus	6069	285
Train	4571	190
Truck	9973	415
Boat	10759	430
Fire Hydrant	1865	101
Bench	9838	413
Bird	10806	440
Cat	4768	202
Dog	5508	218
Horse	6587	273
Sheep	9509	361
Cow	8147	380
Elephant	5513	255
Bear	1294	71
Fork	5479	215
Knife	7770	326
Spoon	6165	253
Bowl	14358	626
Banana	9458	379
Apple	5851	239
Sandwich	4373	177
Orange	6399	287
Broccoli	7308	316
Carrot	7852	371
Hot Dog	2918	127
Pizza	5821	285
Donut	7179	338
Cake	6353	316
Chair	38491	1791
Couch	5779	261
Potted Plant	8652	343
Bed	4192	163
Dining Table	15714	697

Class	Training	Validation
Zebra	5303	268
Giraffe	5131	232
Backpack	8720	371
Umbrella	11431	413
Handbag	12354	540
Tie	6496	254
Suitcase	6192	303
Frisbee	2682	115
Skis	6646	241
Snowboard	2685	69
Sports Ball	6347	263
Kite	9076	336
Baseball Bat	3276	146
Baseball Glove	3747	148
Skateboard	5543	179
Surfboard	6126	269
Tennis Racket	4812	225
Bottle	24342	1025
Wine Glass	7913	343
Cup	20650	899
TV	5805	288
Laptop	4970	231
Mouse	2262	106
Remote	5703	283
Keyboard	2855	153
Cell Phone	6434	262
Microwave	1673	55
Oven	3334	143
Toaster	225	9
Sink	5610	225
Refrigerator	2637	126
Book	24715	1161
Clock	6334	267
Vase	6613	277
Scissors	1481	36
Teddy Bear	4793	191
Hair Drier	198	11
Toothbrush	1954	57
Toilet	4157	179

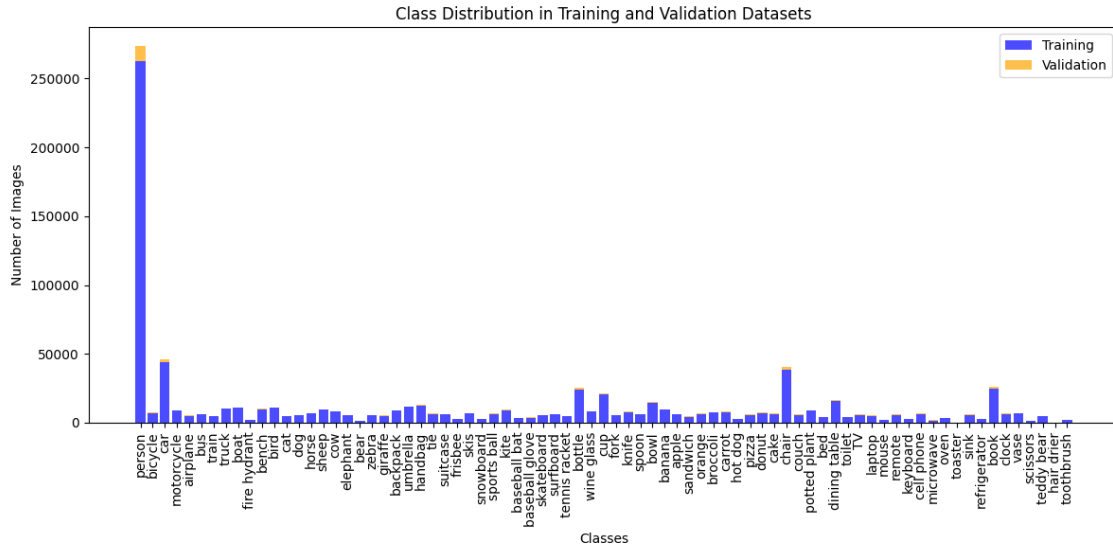


Figure 4.1: Distribution of training and Validation images

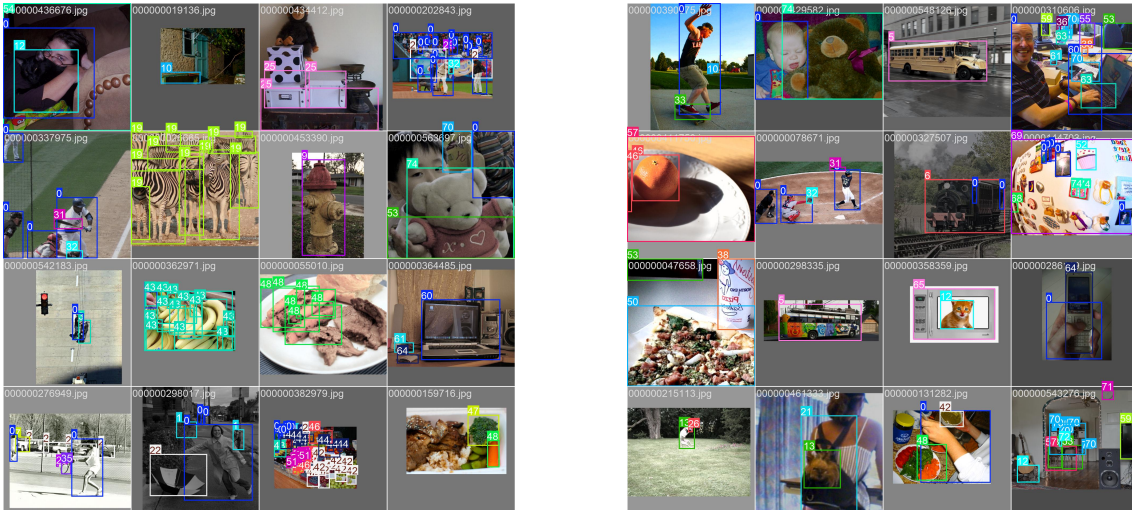


Figure 4.2: Localized Images

4.1.2 Primary Dataset

To be more precise and enhance the model’s capability we introduced 10 new classes. They are, Pole, Zebra Crossing, Pedestrian Green Light, Pedestrian Red Light, Red Traffic Lights, Yellow Traffic Lights, Manhole, Stairs and Bus stop. With Roboflow, we compiled nearly 11490 training images and 2878 validation images that include a mix of different objects. To generate annotations for these classes, Roboflow was employed which helped in accurate annotation of the object locations within an image. This integrated strategy is not only beneficial for data enrichment but also enhances the performance of training models in challenging navigation situations, facilitating better assistance to visually impaired individuals.

Table 4.2: Training and Validation size of object Class

Class	Training	Validation
Pole	247	59
Zebra Crossing	3042	744
Pedestrian Green Light	1020	254
Pedestrian Red Light	1267	314
Red Traffic Light	1739	449
Green Traffic Light	1740	443
Yellow Traffic Light	1014	255
Manhole	1512	387
Stairs	1101	277
Bus Stop	912	235

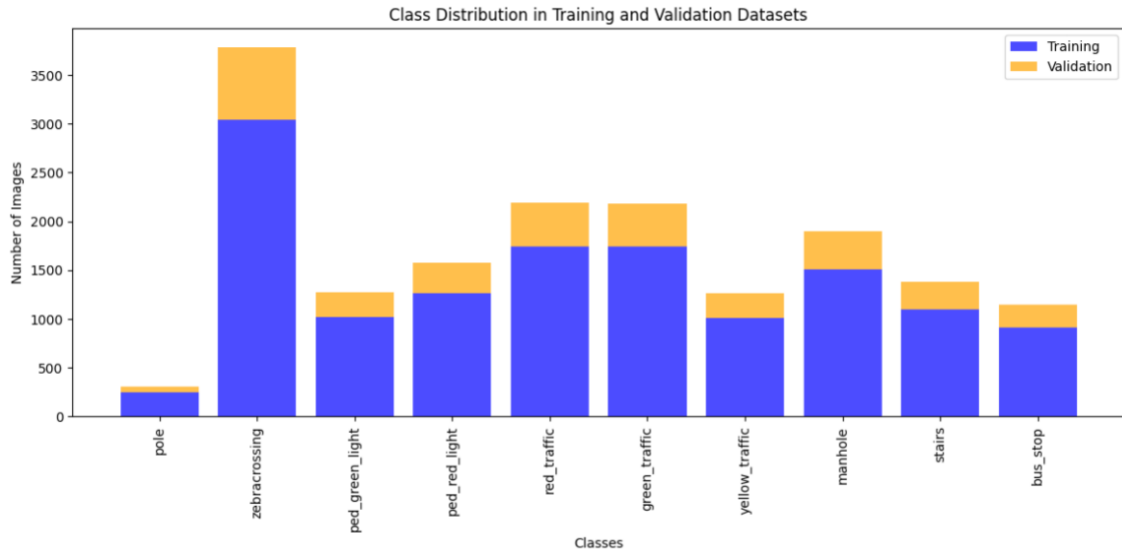


Figure 4.3: Distribution of training and Validation images

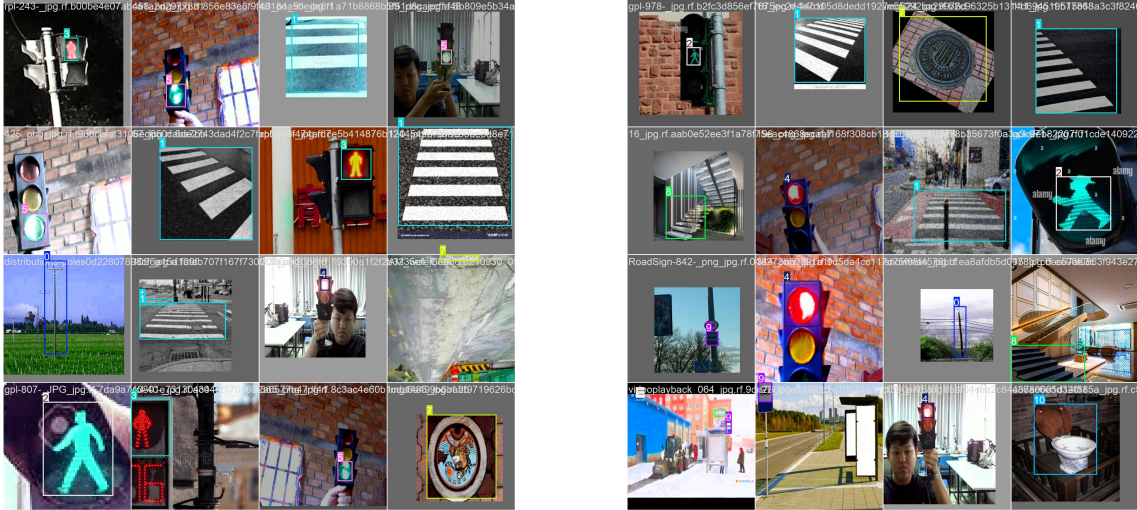


Figure 4.4: Localized Images

4.1.3 Data Annotation Process

Establishing new projects in the various settings for an annotation job, where labels are established, is the first step in using Roboflow for data annotation. Next, we bulk upload pictures to the interface from the Roboflow platform. Users can manually annotate the photos with tools such as bounding boxes for rectangular items and polygons for irregular forms, once the 10 new object classes (e.g., Pole, Zebra Crossing, Traffic Lights) are specified. This ensures that each element has the appropriate class given to it. The workflow is then subjected to quality control so that annotations may be assessed and corrected for correctness. Finally, ML methods may be utilized directly with the annotated data after it has been exported into COCO or CSV format. As a result of this streamlined procedure, the annotation accuracy is very high, allowing the system to recognize key objects in real-time for providing directions to blind people.

4.2 Fine-Tuning Dataset Generation for Navigation Sentence

We created two separate datasets and compiled them together for fine-tuning LLaMA 3.2-3B Instruct to generate navigational sentences. These datasets are crucial to help the model learn how to tackle navigating through real-world spaces. The datasets are:

1. Navigating Sentence Generation Dataset with 948 entries.
2. Description Generation Dataset with 2,269 entries.

Both are well-organized datasets with three important parameters (instruction, input, output). These 3 things help the model to capture what task is supposed to be done. To take input from a given environment and generate possible answers as seen on how we do a situation-based command execution.

4.2.1 Navigating Sentence Generation Dataset

The Navigating Sentence Generation Dataset is targeted for training the model to provide real-time navigational instruction. The instructions rely on object detection in the world, as well as the poses of objects relative to each other (e.g., people; zebra crossings and pedestrian signals). These data sets are particularly useful in helping visually impaired individuals who need to understand the whereabouts and routes for movement or safety. Each entry in this dataset follows a consistent structure:

Instruction: Directs the model to generate a navigational sentence.

Input: A description of the scene, including objects and their spatial positions (e.g., left, right, center).

Output: The corresponding navigation command, which directs the user on what actions to take based on the input.

Examples of Entries:

Instruction: Generate a navigating sentence.

Input: There are 2 persons on the left, 1 person in the center, 1 zebra crossing on the right, 1 ped_green_light in the center.

Output: You can cross at the zebra crossing on your right.

Instruction: Generate a navigating sentence.

Input: There are 3 persons on the right, 2 persons in the center, 1 zebra crossing on the left, 1 ped_red_light in the center.

Output: Do not cross the zebra crossing on the left.

Instruction: Generate a navigating sentence.

Input: There is 1 person on the left, 1 stairs in the center.

Output: Proceed to the stairs in the center.

Instruction: Generate a navigating sentence.

Input: There is 1 manhole on the left, 1 bus stop on the right.

Output: Manhole on the left, move right, proceed to the bus stop.

Instruction: Generate a navigating sentence.

Input: There is 1 manhole in the center, 1 footpath on the right.

Output: Manhole in the center, move right to the footpath, proceed with caution.

4.2.2 Description Generation Dataset

The Description Generation Dataset focuses on training the model to output elaborate descriptions regarding either objects or interactions in the environment. The navigating sentence dataset just gives guidance, whereas this one describes objects and what users can do with them such as positioning a baseball bat for playing or placing a bottle in order to drink. Dataset Structure: Each entry in this dataset follows the same three-parameter structure as the navigation dataset:

Instruction: Directs the model to generate a description.

Input: A detailed description of the environment, including objects and their spatial positions.

Output: The generated description of the object interactions, often focusing on how the user can interact with the object.

Examples of Entries:

instruction: "Generate a description"

input: "There are 3 persons on the right, 1 person in the center, 1 baseball bat on the left."

output: "A baseball bat is on your left, you can play with it."

instruction: "Generate a description"

input: "There are 2 persons on the right, 1 person in the center, 1 bottle on the left."

output: "A bottle is on your left, you can drink from it."

instruction: "Generate a description"

input: "There are 2 persons on the right, 1 person in the center, 1 wine glass on the left, 1 bottle on the right."

output: "The person to your left is consuming wine from a bottle using a glass."

instruction: "Generate a description"

input: "There is 1 bowl on the right, 1 apple on the right, 1 person in the center, 2 bowls on the left."

output: "You can eat the apple from the bowl on your right."

4.2.3 Dataset Generation Process

We carefully generated these datasets so that the model will be able to produce accurate results for both navigational commands and object descriptions. Here's an

outline of how it is:

Scenario Design: Real-world scenarios were created to simulate the types of environments visually impaired individuals might encounter. Scenarios were based on common objects (people, zebra crossings and pedestrian lights as well specific tasks (crossing a road, avoiding hazards).

Manual Entry Construction: We built each turn by hand to cover a wide range of diverse high-level navigation and object description entry behaviors. The input was specifically crafted to include object positions and types, while the output was adapted to deliver contextual navigation instructions.

Diverse Examples: The model can be expected to work in many potential scenarios, these datasets are training with various examples - from basic commands such as "crossing road" up until more detailed sentences listing several objects and interactions (e.g. avoiding obstacles; interacting with items).

Fine-Tuning Focus: Through this structuring of the data, we fine-tuned both our LLM models and chose the best model with a specific emphasis to learn from navigation-specific issues and provide pinpointed clear responses. By optimizing between guiding sentence generation and object description, we can provide the balance that enables models to give useful contextually rich guidance.

Chapter 5

Methodology

The goal of this research was to create a method of navigational guidance for the blind people by objectifying classes and localization using an ensemble model of YOLOv11, Faster R-CNN and then from class and location LLaMa will generate a navigational sentence and this sentence will be converted into audio through GTTS.

5.1 Proposed Model

Our proposed model aims to provide navigational guidance for the blind persons by focusing on key outdoor objects. Initially, we trained YOLOv11 on the MSCOCO dataset, but it struggled with critical classes like zebra crossings and stairs, which were absent. To address this, we created a primary dataset of a amount of images with 10 key classes and used a hybrid approach combining YOLOv11 and Faster R-CNN. Then we trained YOLOV11 with the primary dataset. After the processing steps of YOLOV11 like Convulational Neural Network layer, Cross Stage Partial network, Spatial Pyramyd Pooling, Feature Aggresstion, upsamling and concatenate etc. give the output of classified objects name with a bounding box and a confident score. For the Faster R-CNN model we also trained it with primary dataset and changed the layers to get more fine result. Here after feature extraction we used PCA(Principle Component Analysis) to resize the feature and generated smaller feature which is efficient for the model. As the Both models were trained separately, and their results were ensembled with selecting predictions by higher confidence score for each object. This improved accuracy while maintaining real-time performance.

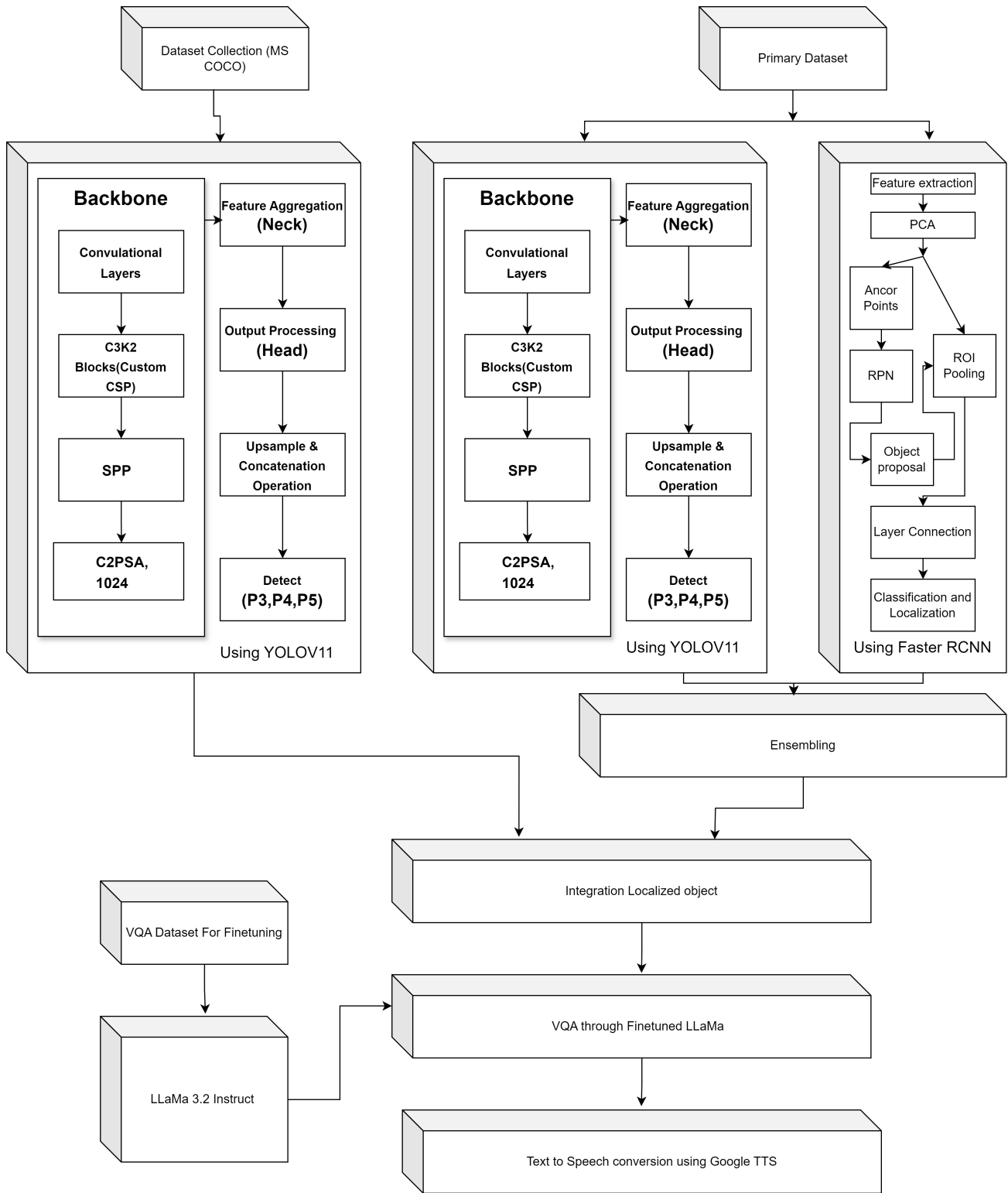


Figure 5.1: Top Level Overview of The Proposed Model Diagram

We also trained LLaMA 3.2 Instruct 3B to generate descriptive sentences from detected objects, fine-tuning it with Low-Rank Adaptation (LoRA) for efficient learning. Finally, these descriptions were converted to audio using Google Text-to-Speech (gTTS) and it is providing real-time, voice-guided navigation for users.

5.1.1 YOLOv11 for Object Detection

Firstly, we utilized YOLOv11 which is a real-time object detection system. It provides bounding boxes and class probabilities for objects that appear in images. It was first trained on COCO dataset, which includes a wide range of classes used but it was not inclusive of specific classes that would be used for outdoor navigation. Therefore, a primary dataset was developed, which includes 10 classes that are critical in outdoor navigation. The dataset includes 14,975 images. The classes to be included in the dataset were determined by the fact that they are critical in enabling the blind to navigate. They include zebra crossing, pole, stairs, bus stop.

We again trained YOLOv11 using the primary dataset to enable it to recognize navigation-related objects more effectively. We then used two models during inference:

- **Integrated Model:** It works by predicting bounding boxes and classes for navigation-related objects. It was trained on the primary dataset.
- **COCO Model:** It is used to detect general objects in the scene. It was trained on COCO classes.

During inference, the two YOLO models work independently to provide bounding boxes, labels and confidence scores.

5.1.2 Faster R-CNN for Object Detection

At first We added Faster R-CNN as an alternative which is another powerful object detection model. After implementing the model we realized it is give more accurate result than YOLOV11 though it is a bit slower than YOLOV11. The YOLO model detects objects on a single-pass basis, but Faster R-CNN is designed to detect objects based on a two-pass region of interest proposal model and then to classify these regions. This method is especially useful for detecting smaller objects or objects that are less common. Faster R-CNN was trained exclusively on the primary dataset composed of 10 navigation-related classes. In addition, PCA was implemented within the CNN layers for optimization, reducing the dimension of the extracted features while retaining essential data and reducing training complexity and time. The architecture of Faster R-CNN allowed for targeting the fine detail of our classes more efficiently, increasing the correct detection of the objects.

5.1.3 Ensembling YOLOv11 and Faster R-CNN

We used an ensembling method to improve the accuracy of detections on both YOLOv11 and Faster R-CNN over our data. After having predictions from the two models, we consolidated the best from both to obtain more accurate detections. The process is as follows:

- **Separate Inference:** Both models run separately on the input images. Both outputs detections, which are lists of boxes, confidence scores, and labels for the objects present in the input image.
- **Confidence-based Selection:** For each predicted box from both models, the confidence scores are compared. The box that carries the highest confidence

label is selected. This method takes advantage of the virtue of both models, which are in the speed and precision of YOLOv11 and Faster R-CNN, respectively.

- Do not keep redundant detections: To ensure our output detections are clean and precise, the best way to obtain this is to remove duplicates. For this reason, Intersection-over-Union (IoU) calculation is done. If two detections have an IoU score that shows high overlap between the pair of detections, then one is removed. The detection with the lower confidence score is removed. After such calculations, our data has only the non-overlapping detections between both models with the higher confidence detection. We kept the threshold IoU value at 0.5 to detect overlap.

After testing we found out our approach to ensembling the output from the two models in this scheme improved the accuracy as it better detected the navigation's in the data.

IoU Calculation:

$$\text{IoU} = \frac{\text{Area of Intersection}}{\text{Area of Union}} \quad (\text{ii})$$

Intersection is the area where the two bounding boxes overlap.
Union is the combined area covered by both bounding boxes.

Area of Intersection:

$$\text{inter_area} = \max(0, \text{inter_x2} - \text{inter_x1}) \times \max(0, \text{inter_y2} - \text{inter_y1}) \quad (\text{iii})$$

Area of Box 1:

$$\text{box1_area} = (x2 - x1) \times (y2 - y1) \quad (\text{iv})$$

Area of Box 2:

$$\text{box2_area} = (x2b - x1b) \times (y2b - y1b) \quad (\text{v})$$

Final IoU Formula:

$$\text{IoU} = \frac{\text{inter_area}}{\text{box1_area} + \text{box2_area} - \text{inter_area}} \quad (\text{vi})$$

5.1.4 Post-Processing and Combining Detections

After getting the localized objects along with confidence scores for each objects from YOLOV11 trained with MSCOCO Dataset and from the ensembling of YOLOV11 and Faster R-CNN with our primary dataset we convert it into a consistent format and normalize based on the image width. . Using the normalized center, objects are classified into regions: the left region if the normalized x-center is less than or equal to 0.33, the center region if it falls between 0.33 and 0.66, and the right region if it exceeds 0.66.

Left Region:

$$\text{Region} = \text{Left}, \quad \text{if } x_{\text{center}} \leq 0.33 \quad (\text{vii})$$

Center Region:

$$\text{Region} = \text{Center}, \quad \text{if } 0.33 < x_{\text{center}} \leq 0.66 \quad (\text{viii})$$

Right Region:

$$\text{Region} = \text{Right}, \quad \text{if } x_{\text{center}} > 0.66 \quad (\text{ix})$$

As each object is classified into these regions, our system counts how many objects of each type are found in each regions of the image and creating a summary that works as a input for the fine tuned Llama that will generate the navigating or descriptive sentence.

Chapter 6

Experimental Evaluation

In this section, we will provide the specifications of the desktop computer that ran all of our models. Next, we will demonstrate the superiority of our model compared to competing models through both qualitative and quantitative measurements.

6.1 Experimental Setup

Specifications for YOLOv11 and Faster R-CNN Training

- **Processor:** AMD Ryzen 9 5950X 16-Core
- **GPU:** NVIDIA GeForce RTX 3080 Ti with 12 GB GDDR6X Memory
- **RAM:** 64 GB

Specifications for Fine-tuning Large Language Models(LLMs)

- **GPU:** NVIDIA A100 with 40 GB HBM2 Memory
- **RAM:** 84 GB System Memory

The A100 GPU is part of NVIDIA's Ampere architecture and is specifically designed for AI and machine learning workloads. It has 40 GB of high-bandwidth memory (HBM2) that is ideal for handling large models and datasets efficiently. Moreover, it is coupled with 84 GB of system RAM which configuration provides the necessary computational power and memory for fine-tuning large-scale language models, allowing for the effective handling of large parameter sets and demanding training processes.

6.2 Experimental Configuration

6.2.1 YOLOV11

MSCOCO Dataset

- Epochs: 35
- Batch Size: 16
- Learning Rate: 0.01 (cosine decay).
- Image Size: 640
- Scheduler: Cosine decay learning rate scheduler
- Optimizer: SGD (Stochastic Gradient Descent)

Primary Dataset

- Epochs: 70
- Batch Size: 16
- Learning Rate: 0.01 (cosine decay).
- Image Size: 640
- Scheduler: Cosine decay learning rate scheduler
- Optimizer: SGD (Stochastic Gradient Descent)

6.2.2 Faster R-CNN

- Epochs: 70
- Batch Size: 16
- Learning Rate: 0.001
- Optimizer: Adam optimizer

6.2.3 LLM Configuration and Training Setup

The model’s maximum sequence length was set to 512 tokens, ensuring efficient handling of input sequences without exceeding memory limits. Next, the pretrained model is loaded in 4-bit precision, which significantly reduces memory usage while enhancing computational efficiency and allowing us for faster processing with minimal resource consumption.

LoRA setup:

- r (rank) : 16
- α (alpha): 16
- Targeted modules: q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, and down_proj.
- bias = none
- Dropout: 0
- Random seed: 3407

Trainer setup:

- Batch size: Set to 2 per device.
- Gradient accumulation: 4 steps
- Warmup steps: 5 steps
- Training steps: 1,500 steps
- Learning rate: $2e-4$
- Optimizer: adamw_8bit
- Weight decay: 0.01
- Logging: Steps are logged after every iteration to monitor training progress.

6.3 Experimental Findings

6.3.1 Evaluation of Faster R-CNN and YOLOV11

1. Precision:

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}} \quad (\text{i})$$

2. Recall:

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}} \quad (\text{ii})$$

3. F1 Score:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (\text{iii})$$

Table 6.1: Model Performance Metrics of YOLOV11 and Faster-RCNN

Model	Precision	Recall	F1	mAP50	mAP50-95
F-RCNN(Primary Dataset)	0.9849	0.9862	0.9855	None	None
YoloV11(Primary Dataset)	0.9620	0.9116	0.9361	0.9422	0.7706
YoloV11(Coco Dataset)	0.6373	0.4649	0.5376	0.5115	0.3610

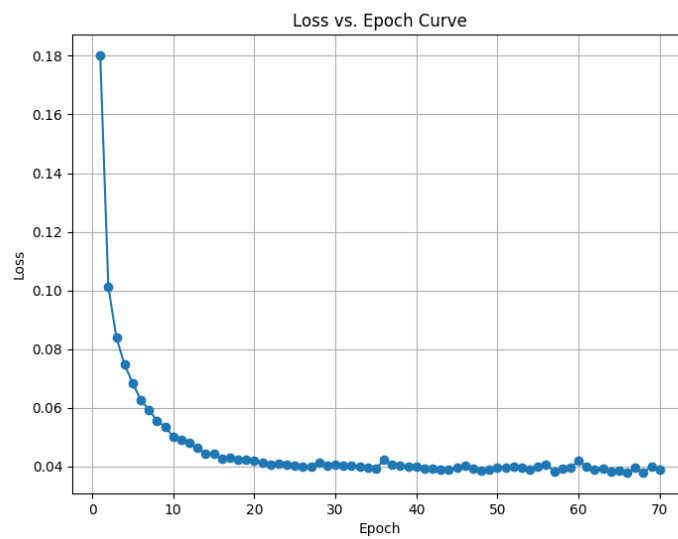


Figure 6.1: Loss vs Epoch Curve of Faster R-CNN

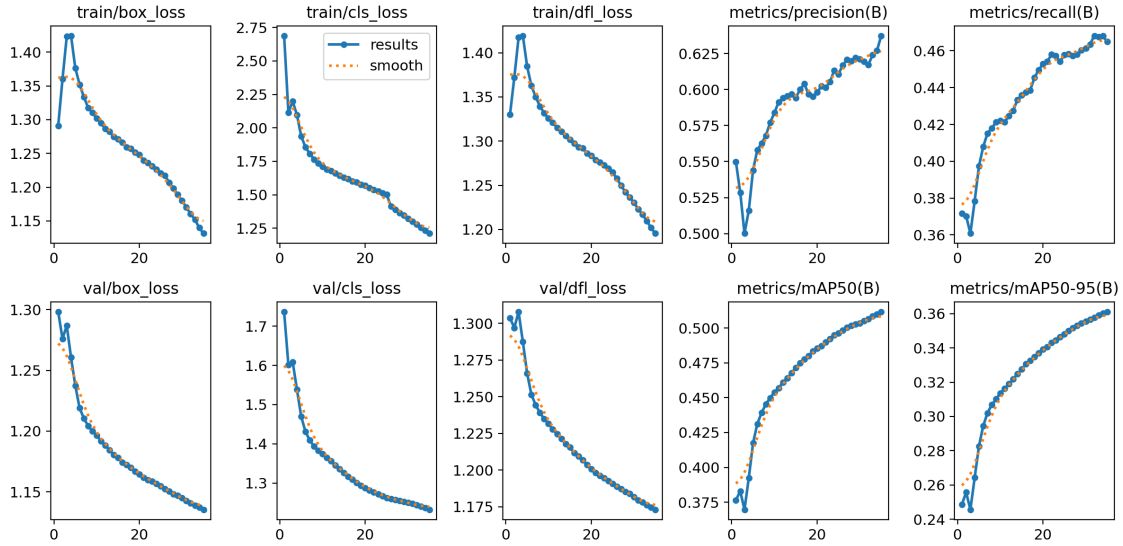


Figure 6.2: Training Results of YOLOV11 on COCO Dataset

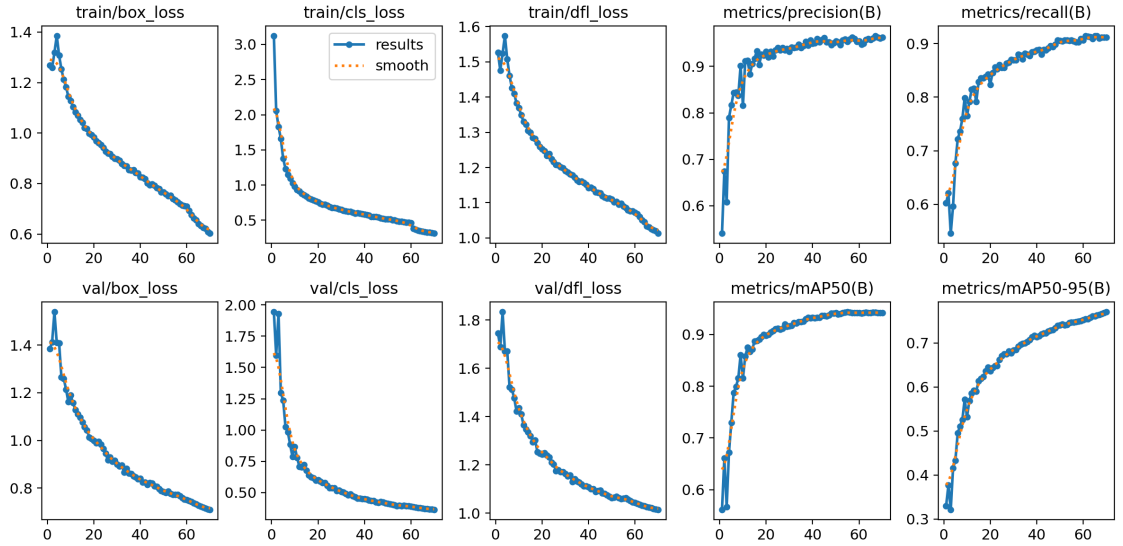


Figure 6.3: Training Results of YOLOV11 on Primary Dataset

6.3.2 Evaluation of Fine Tuned Llama 3.2-3B-Instruct

For Comparison of Llama 3.2-3B-Instruct before and after fine tuning, we have used BLEU and ROUGE metrics because they are very popular in natural language processing and are used to evaluate how well generated text is. BLEU metric is designed for recall, measuring the quality of the machine translation by comparing overlapping n-grams or word sequences generated from a corpus to manually created reference translations. The higher the BLEU score, the better overlap but it can be easy to get high scores for synonymous or paraphrase copying. After that, ROUGE score is also used on the basis of recall which means how much can our generated text hold that were present in provided reference during evaluating operation.

Comparison of before fine-tuning and after fine-tuning:

Instruction: Generate a description

Input: There is 1 knife in the right, 1 apple on the left, 1 person on the right.

Output: You can cut an apple with a knife on your right.

Output before fine-tuning: There is 1 knife in the right, 1 apple on the left, 1 person on the right. The person is pointing at the knife. The scene is set in a dark forest.

Here, we can see that the output was poorly organized and unable to provide a good description without fine-tuning.

Output after fine-tuning: You can cut an apple with a knife on your right.

But now after fine-tuning we can see that our output is providing the correct descriptive sentence .

Table 6.2: Llama3.2-3B-Instruct BLEU Score

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4
Llama3.2-3B-Instruct(Before Finetune)	0.1858	0.1053	0.066	0.0442
Llama3.2-3B-Instruct (After Finetune)	0.2982	0.196	0.1473	0.1175

Table 6.3: Llama3.2-3B-Instruct ROUGE Score

Model	ROUGE-1	ROUGE-2	ROUGE-L
Llama3.2-3B-Instruct (Before Finetune)	0.2572144958	0.08267532265	0.2102495145
Llama3.2-3B-Instruct (After Finetune)	0.3668390408	0.1617590308	0.3204431557

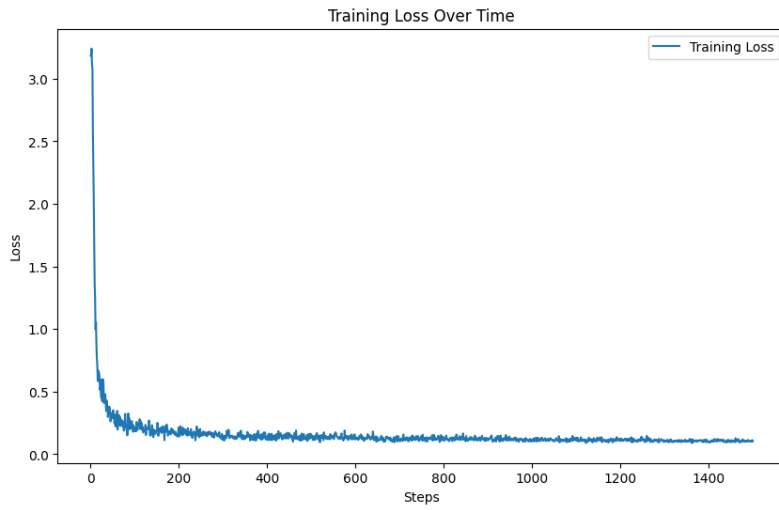


Figure 6.4: Steps vs Loss Curve for LLaMa3.2-3B-Instruct

6.4 Implications and Results



Figure 6.5: Output of YOLO and Faster R-CNN Model for Zebracrossing

Table 6.4: Generated Sentence for Zebracrossing from Llama

Detection from Yolo and Faster RCNN	1 car on the left, 1 zebracrossing on the center, 3 person on the center.
Response from LLAMA	You can cross safely at the zebra crossing on the center.

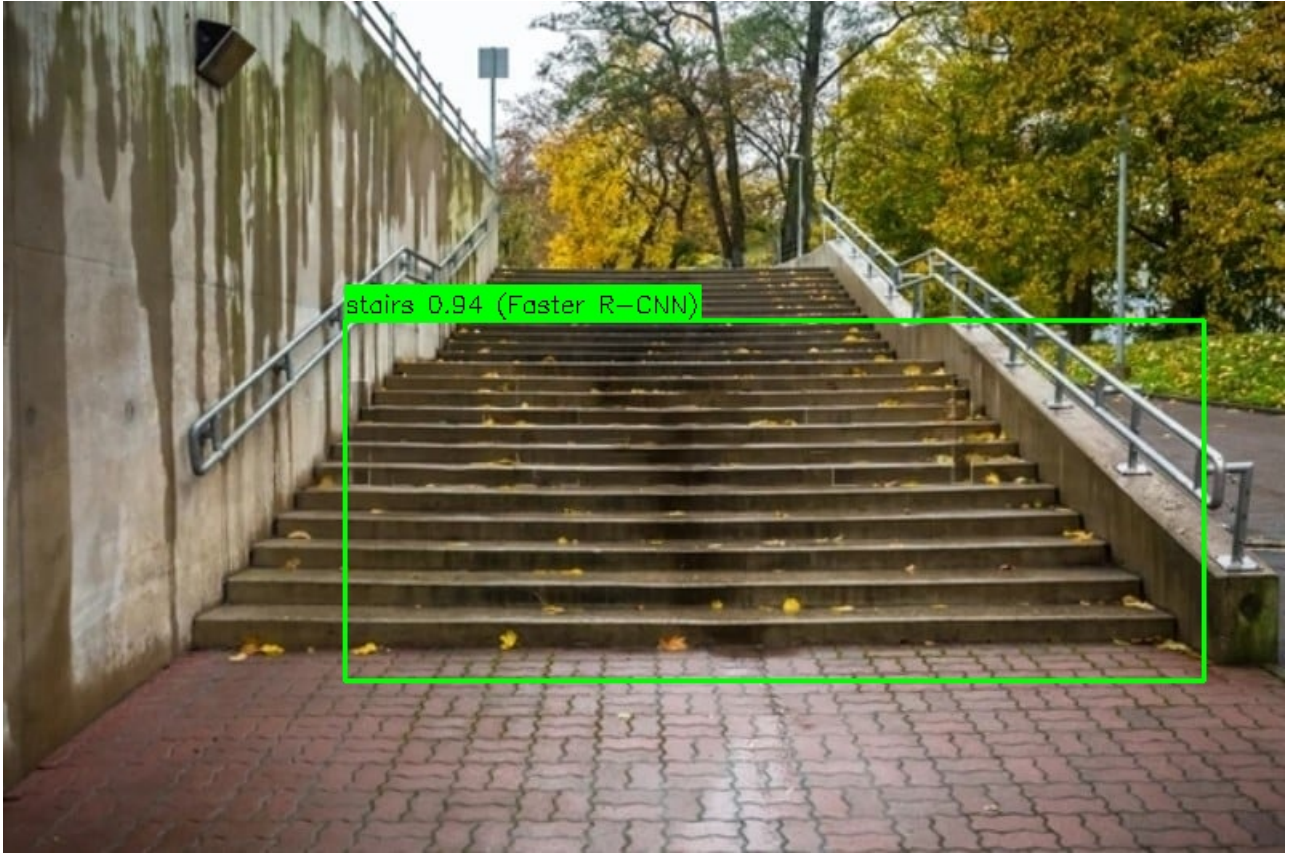


Figure 6.6: Output of YOLO and Faster R-CNN Model for Outdoor Stairs

Table 6.5: Generated Sentence for Outdoor Stairs from Llama

Detection from Yolo and Faster RCNN	1 stairs on the center.
Response from LLAMA	Please proceed to the stairs on the center as you walk.



Figure 6.7: Output of YOLO and Faster R-CNN Model for Indoor Stairs

Table 6.6: Generated Sentence for Indoor Stairs from Llama

Detection from Yolo and Faster RCNN	1 stairs on the center.
Response from LLAMA	Please proceed to the stairs on the center as you walk.



Figure 6.8: Output of YOLO and Faster R-CNN Model for Red Traffic Light

Table 6.7: Generated Sentence for Red Traffic Light from Llama

Detection from Yolo and Faster RCNN	1 person on the left, 1 red traffic on the right.
Response from LLAMA	A red traffic light is on your right, you can cross safely.

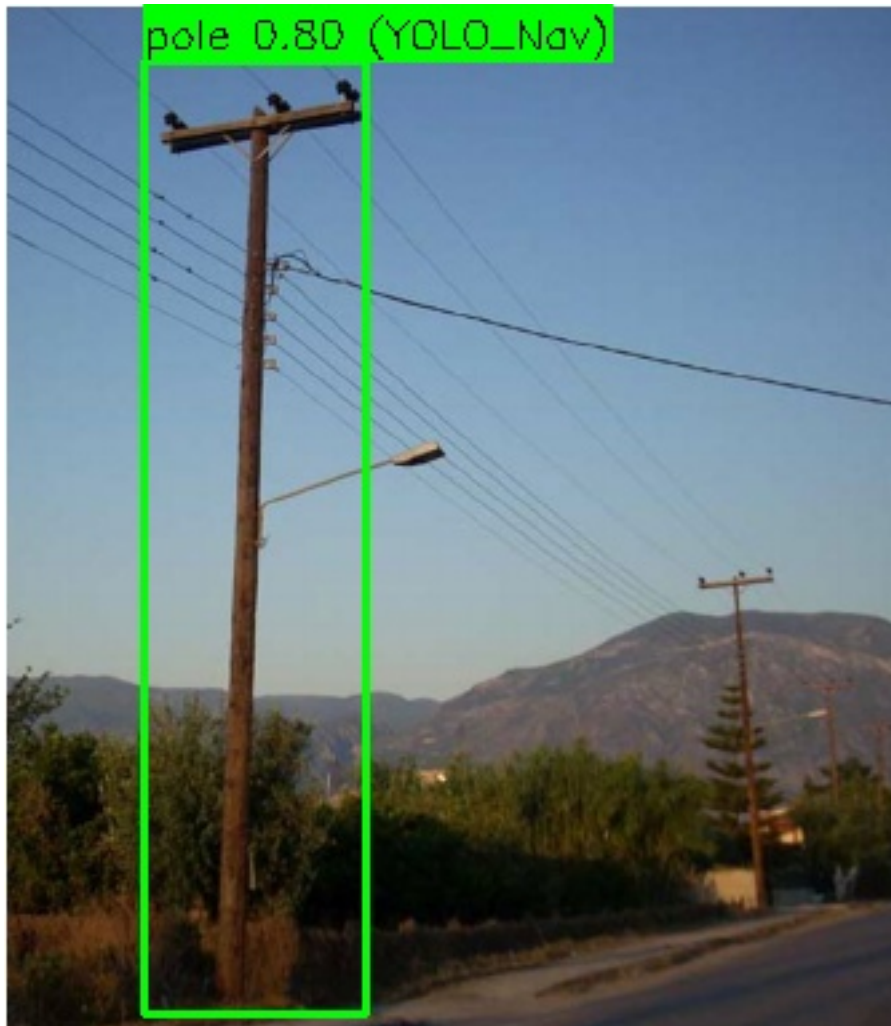


Figure 6.9: Output of YOLO and Faster R-CNN Model for Pole

Table 6.8: Generated Sentence for Pole from Llama

Detection from Yolo and Faster RCNN	1 pole on the left.
Response from LLAMA	Pole on the left, move right.



Figure 6.10: Output of YOLO and Faster R-CNN Model for Pedestrian Red

Table 6.9: Generated Sentence for Pedestrian Red from Llama

Detection from Yolo and Faster RCNN	1 ped red light on the center.
Response from LLAMA	Do not cross as the pedestrian red light on the center is on.

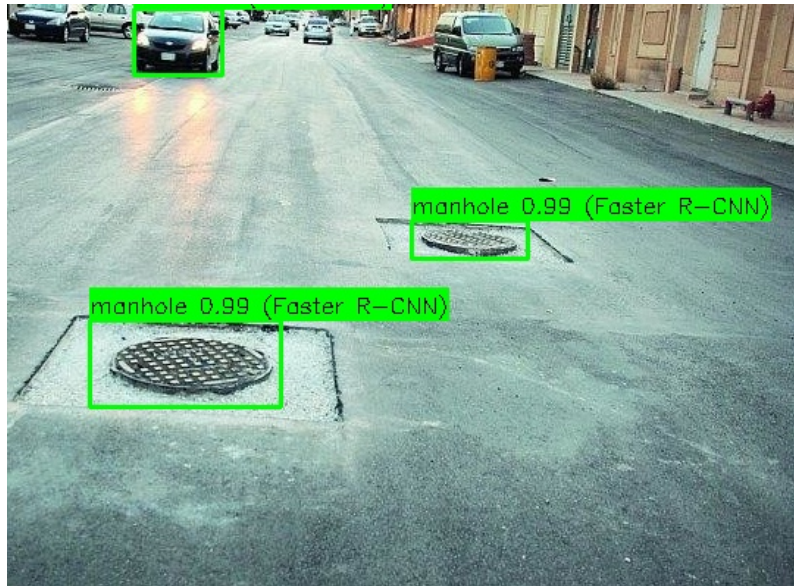


Figure 6.11: Output of YOLO and Faster R-CNN Model for Manhole without Person

Table 6.10: Generated Sentence for Manhole without Person from Llama

Detection from Yolo and Faster RCNN	1 manhole on the left, 1 car on the left, 1 manhole on the center.
Response from LLAMA	A manhole is on your left and another manhole is in front of you so, stay right.



Figure 6.12: Output of YOLO and Faster R-CNN Model for Manhole with person

Table 6.11: Generated Sentence for Manhole with Person from Llama

Detection from Yolo and Faster RCNN	1 manhole on the center, 2 person on the center.
Response from LLAMA	Manhole on the center, move left or right.

6.5 Discussion

In our paper a novel method of navigation for the visually impaired individual is presented using YOLOv11, Faster R-CNN along with Llama 3.2-3b Instruct. Although vast improvements have been made in assistive technologies, previous studies show that contexts are missing location and analysis of contextual information while the individual is using the device. Our focus was to build a system which not only detects obstacles but also provides location and detailed spatial arrangement and origin of sounds, providing them precise navigation capabilities.

Rapid object detection for timely task-related decision making in complex environments is done by integrating YOLOv11. This model showed high accuracy in navigation and feedback on the obstacles obstructing the path of a visually impaired individual. Faster R-CNN, on the other hand, improves classification accuracy by making more accurate decisions of whether a detected box contains the desired objects or is in the background. Navigation instructions are made context by the Llama 3.2-3B Instruct model creating clarity of surroundings and path directions help to empower the knowledge necessary for navigating safely. Such capability is important for better user confidence and smooth navigation. In addition, users can easily get exact location data by our system and make the correct decision for navigating. Results obtained through our work not only demonstrate the viability of integrating technologies to help visually impaired people, but also raise attention to general

challenges in AI-enabled solutions for Navigation. We intend to develop an even more precise manner for guiding navigation results in a robust and dependable way, which will bring about exceptional autonomy and mobility for blind users.




suppose u have to explain the navigating path to a blind person. so generate a navigating sentence that where he should go (must be in one line)



To reach the manhole, walk straight on a smooth paved road with some cracked areas, follow the road to your right until you feel a slight upward slope, then veer left at the curb edge until you hear the hollow sound of the open manhole ahead, located near a highway exit.

Figure 6.13: Comparison with GPT

⚡ **Visual Question Answering demo**
 using [dandelin/vilt-b32-finetuned-vqa](#)
 🖼️ Visual Question Answering Examples ▾



where should the person taking the photo go for safe navigation Compute

road	0.163
street	0.074
left	0.029
car	0.022
fire hydrant	0.022

Computation time: 0.121 s

[</> View Code](#) [🖼️ Maximize](#)

Figure 6.14: Comparison with VQA

When comparing our system with other models such as GPT-4o and VQA (Visual Question Answering), the differences are clear. These models were tested by asking them to generate navigation instructions and object detection tasks. However, both models failed to deliver accurate results. For instance, GPT-4o and VQA often gave incorrect and unorganized navigation guidance, which can be critical when helping visually impaired individuals. In contrast, our system provides clear, accurate, and context-relevant instructions, making it a more reliable option for navigation assistance.

Chapter 7

Limitations and Future Work

7.1 Conclusion

In our study, we made a navigation system for the visually impaired people by using object detection models YOLOv11, Faster R-CNN with Llama 3.2-3B Instruct as well. This combination greatly provides the obstacle detection and awareness that people need to better move through their environment without any human assistance. The fine-tuned Llama 3.2-3B Instruct model achieved competitive performance metrics, a BLEU-4 score of .1175 and ROUGE-L of .3204. Demonstrating its capacity to generate accurate, contextually relevant navigation instructions with far less redundancy than other models. Integrating YOLOv11 helped in fast detection of obstacles, which is an important factor as timely decisions need to be taken in dynamic environments. Faster R-CNN improves this accuracy of class based recognition and hence delivers dependable information to the users about its surroundings. The model identifies obstacles, describes the scene and tells people where to go so they can make informed navigational decisions. This capability is essential for enhancing user confidence and safety while navigating complex and unfamiliar environments. Our results show the promise of such a combination to provide effective and practical solutions for visually impaired individuals. This work is particularly well placed to highlight how AI technologies are revolutionizing improving navigation accuracy, cost constraint and user confidence for a better living experience in persons who are blind or visually impaired.

7.2 Limitations

In our paper, we proposed an advanced assistance system based on YoloV11, LLaMa 3.2-3B Instruct and Faster R-CNN to help visually impaired people by achieving improved mobility. While these technologies hold significant promise, several limitations need to be addressed to fully realize their potential in real-world applications.

1. **Real-time Object Detection:** Real-Time Object Detection is a significant limitation to our current study, because it is difficult to apply LLaMA 3.2-3B and Faster R-CNN for real-time material detecting problems due to their model architecture limitations. However, as these models are relatively complex and computationally expensive, they simply cannot deliver the real-time instant reactions needed for most of today's applications (which require milliseconds).

LLaMA 3.2-3B: A large-scale language model based on transformers having billions of parameters has a very high latency when it comes to processing inputs and producing outputs from them. It is too expensive as a real-time data recovery component on devices without specialist hardware and computational power like powerful GPUs. Faster R-CNN also segments region proposals and classifies them through a two-step method that is very accurate at finding objects. This makes it less suitable for real-time detection, particularly in dynamic scenarios requiring immediate action. The computational burden of these models would be a major obstacle to meet the real-time performance requirements using this architecture, which requires adjusting itself for operations in changing environments.

2. **Measurement of Distance:** Our study lacks to give accurate measuring distances. The Faster R-CNN could only detect and classify objects accurately without accurate depth to objects compared to ground truth. Without this depth information, the system effectiveness can be substantially degraded as to detect a barrier or anything else. At least two user requirements are necessary. They are its existence and its distance. Individuals who are blind need a way to detect how far away from an obstacle they. Mitigating risk by measuring distance to Obstacle In a world with variable obstacle distances, real-time navigation requires accurate depth perception (which is confined by the monocular camera array of the current system). This will be very difficult as we cannot get an angle of the trajectory and future versions should consider including LiDAR, stereo vision or any other depth type sensing technologies to sort this out. This will improve context awareness and provide proper details about their surroundings.
3. **High Hardware Requirements for Real-Time Processing:** In devices where computational resources are the limiting performance factor, Faster R-CNN and LLaMA 3.2-3B Instruct have high hardware requirements that can limit their ability to process in real time. This makes implementing LLaMA 3.2-3B Instruct with its large-scale transformer architecture that depends on either specialized hardware or high-end GPUs for effective operation challenges to test in the region where potential users have portable, low-resource devices more commonly used by visually impaired people. While we can always improve the model using distillation or pruning, it gets into replacing accuracy with a large complex network. Similar to this, the two-stage of Faster R-CNN also requires heavy computing to give accurate results which makes it less suitable for low-power or mobile devices. Since It needs a lot of hardware optimizations it becomes difficult for all users to get their hands on the system.
4. **Limited Training on Diverse Environmental Conditions:** The models have been trained on a few datasets but they were not exposed to diverse environmental conditions. These real-world ambient variables affect thus performance of the respective object identification and navigation systems. The current system may fail under very harsh conditions such as heavy rain, fog or dim lightness that will cause the tests accuracy and image quality to decrease. The solution to this is to expand the system's repertoire, so that a broader range of scenarios can be handled and effective performance guaranteed in

different circumstances.

7.3 Future Work

Our study shows what we can achieve by leveraging YOLOv11, Faster R-CNN and LLaMA 3.2-3B Instruct to help visually impaired people to navigate through the physical world. However, the research identified many challenges in it that future work will need to address and improve.

1. **Real-Time Object Detection Improvements:** YOLOv11 and Faster R-CNN are well-known as popular object detection models, however these favorite model styles for improved real-time consistent recognition have also been plagued with many issues due to additional complexities. Our future work for The YOLOv11 and Faster R-CNN models will be further optimized for real-time performance. Hardware acceleration, quantization and model pruning etc. will be evaluated to reduce the computational cost/delay so that we can process as quickly as possible. The aim is to allow visually impaired individuals to better and more safely navigate in dynamic environments with fast-changing obstacles that might be unknowable at the time of model training.
2. **Managing Distance Measurement:** Our current study does not convey information about the distance from detected obstacles of a person, if an individual resides in close proximity to any obstacle or is distanced from it cannot communicate by technology. In a forthcoming work, we will employ such distance measurements via depth sensors to improve perception. This will allow visually impaired individuals to improve their decision making when considering navigation, as the distances of obstacles can be pinpointed instead of simply detected. Precise distance measurement is necessary to improve the safety features and real-time responsiveness of a system.
3. **High Hardware Requirements for Real-Time Processing:** The models YOLOv11, Faster R-CNN and LLaMA 3.2-3B Instruct are complex networks. It requires large computation operations. GTX GPUs or greater versions are needed to achieve an optimal performance for real-time processing of the data published from cameras. It slows down the navigation of blind people on small devices. Our future work will investigate lighter versions of these models and hardware optimization to reduce computational needs. We can place our system on final devices, allow these final nodes to use smaller and cheaper hardware, which allows us then that we go closer to the user in a larger part of industries because even here it is now possible.
4. **Limited Training on Diverse Environmental Conditions:** Our current model is limited, mostly based upon kind of weather condition, limited dataset and different area types. Our future works will concentrate on increasing the dataset to include a broader range of real-world scenarios, including low-light, bad weather, and rugged environments. This will help the system be more effective across a wide array of scenarios.
5. **Development of a Noble Navigation Device:** We will also develop a distinct navigation device for the visually impaired from scratch as our future

work. The real technique would be to view it all as a single device, and this is where the distance measuring combined with object detection in real-time actually becomes important. In order to solve this, we will try to create a new navigation device for the visually impaired individuals so that could help them in their daily life by making them able to navigate more easily simply which helps them move independently and safely.

Bibliography

- [1] E. M. Nebot, “Sensors used for autonomous navigation,” in *Advances in Intelligent Autonomous Systems*, ser. International Series on Microprocessor-Based and Intelligent Systems Engineering, S. G. Tzafestas, Ed., vol. 18, Springer, 1999, pp. 81–105. DOI: 10.1007/978-94-011-4790-3_7. [Online]. Available: https://doi.org/10.1007/978-94-011-4790-3_7.
- [2] N. M. Barbour, “Inertial navigation sensors,” in *NATO RTO Lecture Series, RTO-EN-SET-116, Low-Cost Navigation Sensors and Integration Technology*, 2010. [Online]. Available: <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=9dba30cad95662bceb6c0fce6e6c8bc283742e9a>.
- [3] W. H. Organization, *Who releases new global estimates on visual impairment*, Retrieved from, 2012. [Online]. Available: <https://www.emro.who.int/control-and-preventions-of-blindness-and-deafness/announcements/global-estimates-on-visual-impairment.html>.
- [4] Q. Fan, L. Brown, and J. Smith, “A closer look at faster r-cnn for vehicle detection,” in *2016 IEEE Intelligent Vehicles Symposium (IV)*, Gothenburg, Sweden, 2016, pp. 124–129. DOI: 10.1109/IVS.2016.7535375. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/7535375>.
- [5] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, pp. 779–788. DOI: 10.1109/CVPR.2016.91.
- [6] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo, “Image captioning with semantic attention,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 4651–4659. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2016/papers/You_Image_Captioning_With_CVPR_2016_paper.pdf.
- [7] T. Grel, *Region of interest pooling explained*, In Data Science, Deep Learning, Machine Learning, Feb. 2017. [Online]. Available: <https://deepsense.ai/region-of-interest-pooling-explained/>.
- [8] M. Lokanath *et al.*, “Accurate object classification and detection by faster-rcnn,” *IOP Conference Series: Materials Science and Engineering*, vol. 263, p. 052028, 2017. DOI: 10.1088/1757-899X/263/5/052028. [Online]. Available: <https://iopscience.iop.org/article/10.1088/1757-899X/263/5/052028/meta>.
- [9] Y. Liu, “An improved faster r-cnn for object detection,” in *2018 11th International Symposium on Computational Intelligence and Design (ISCID)*, Hangzhou, China, 2018, pp. 119–123. DOI: 10.1109/ISCID.2018.10128. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/8695451>.

- [10] Y. Liu, “An improved faster r-cnn for object detection,” in *2018 11th International Symposium on Computational Intelligence and Design (ISCID)*, Hangzhou, China, 2018, pp. 119–123. DOI: 10.1109/ISCID.2018.10128. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/8695451>.
- [11] C. Cao *et al.*, “An improved faster r-cnn for small object detection,” *IEEE Access*, vol. 7, pp. 106 838–106 846, 2019. DOI: 10.1109/ACCESS.2019.2932731. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/8786135>.
- [12] S. Herdade, A. Kappeler, K. Boakye, and J. Soares, “Image captioning: Transforming objects into words,” in *Advances in Neural Information Processing Systems (NeurIPS 2019)*, 2019, pp. 11 794–11 804.
- [13] L. Huang, W. Wang, J. Chen, and X.-Y. Wei, “Attention on attention for image captioning,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 4634–4643. [Online]. Available: https://openaccess.thecvf.com/content_ICCV_2019/html/Huang_Attention_on_Attention_for_Image_Captioning_ICCV_2019_paper.html.
- [14] W. Wu, Y. Yin, X. Wang, and D. Xu, “Face detection with different scales based on faster r-cnn,” *IEEE Transactions on Cybernetics*, vol. 49, no. 11, pp. 4017–4028, 2019. DOI: 10.1109/TCYB.2018.2859482. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/8435966>.
- [15] E. J. Hu, Y. Shen, P. Wallis, *et al.*, “Lora: Low-rank adaptation of large language models,” *arXiv preprint arXiv:2106.09685*, 2021, Submitted on 17 Jun 2021 (v1), last revised 16 Oct 2021 (this version, v2).
- [16] I. V. S. L. Haritha, M. Harshini, S. Patil, and J. Philip, “Real time object detection using yolo algorithm,” in *2022 6th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, Coimbatore, India, 2022, pp. 1465–1468. DOI: 10.1109/ICECA55336.2022.10009184. [Online]. Available: <https://ieeexplore.ieee.org/document/10009184>.
- [17] M. A. Rahman, S. Siddika, M. A. Al-Baky, and M. J. Mia, “An automated navigation system for blind people,” *Bulletin of Electrical Engineering and Informatics*, vol. 11, no. 1, pp. 201–212, 2022. DOI: 10.11591/eei.v11i1.3452. [Online]. Available: <https://doi.org/10.11591/eei.v11i1.3452>.
- [18] A. B. Atitallah, Y. Said, M. A. Ben Atitallah, *et al.*, “Embedded implementation of an obstacle detection system for blind and visually impaired persons’ assistance navigation,” *Computers and Electrical Engineering*, vol. 108, p. 108 714, 2023. DOI: 10.1016/j.compeleceng.2023.108714. [Online]. Available: <https://www.sciencedirect.com/science/article/abs/pii/S0045790623001386>.
- [19] B. Pydala, T. P. Kumar, and K. K. Baseer, “Smart_eye: A navigation and obstacle detection for visually impaired people through smart app,” *Journal of Applied Engineering and Technological Science (JAETS)*, vol. 4, no. 2, pp. 992–1011, 2023. DOI: 10.37385/jaets.v4i2.2013. [Online]. Available: <https://doi.org/10.37385/jaets.v4i2.2013>.
- [20] K. M. Rahman, *Object detection using yolov5 from scratch with python — computer vision*, 9 min read, Oct. 2023. [Online]. Available: <https://medium.com/@KaziMushfiq1234/object-detection-using-yolov5-from-scartch-with-python-computer-vision-cfb6b65f540b>.

- [21] A. F. Gad and J. Skelton, *Faster r-cnn explained for object detection tasks*, Updated on August 28, 2024, 2024. [Online]. Available: <https://www.digitalocean.com/community/tutorials/faster-r-cnn-explained-object-detection#references>.
- [22] L. A. O. Gonçalves, R. Ghali, and M. A. Akhloufi, “Yolo-based models for smoke and wildfire detection in ground and aerial images,” *Fire*, vol. 7, no. 4, p. 140, 2024. DOI: 10.3390/fire7040140. [Online]. Available: <https://doi.org/10.3390/fire7040140>.
- [23] G. Jocher and J. Qiu, *Ultralytics yolo11*, version 11.0.0, 2024. [Online]. Available: <https://github.com/ultralytics/ultralytics>.
- [24] U. Kallakuri, B. Prakash, A. N. Mazumder, H.-A. Rashid, N. R. Waytowich, and T. Mohsenin, “Atlas: Adaptive landmark acquisition using llm-guided navigation,” *arXiv*, 2024. [Online]. Available: <https://openreview.net/forum?id=VhpxzSWTWj>.
- [25] “Llama 3.2: Handling both text and images,” 2024. [Online]. Available: <https://apidog.com/blog/llama-3-2-ai-multimodal-processing/>.
- [26] P. Wu, Y. Mu, K. Zhou, J. Ma, J. Chen, and C. Liu, “Camon: Cooperative agents for multi-object navigation with llm-based conversations,” *arXiv*, 2024. [Online]. Available: <https://arxiv.org/abs/2407.00632>.
- [27] F. Xie, J. Zhang, and S. Schwertfeger, “Intelligent lidar navigation: Leveraging external information and semantic maps with llm as copilot,” *arXiv*, 2024. [Online]. Available: <https://arxiv.org/abs/2409.08493>.
- [28] A. M. Hasan, K. Samsudin, A. R. Ramli, R. S. Azmir, and S. A. Ismaeel, “A review of navigation systems (integration and algorithms),” *Department of Computer and Communication System Engineering, Universiti Putra Malaysia*,