DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING,
SHARDA SCHOOL OF ENGINEERING AND TECHNOLOGY,
SHARDA UNIVERSITY, GREATER NOIDA

OBJECT DETECTION FOR BLIND USING DEEP LEARNING

A project submitted in partial fulfillment of the requirements for the degree of Bachelor of Technology in Computer Science & Engineering

SUBMITTED BY:

Abhishek Singh (2019576312)
Ronit Attrey (2019001770)

SUPERVISED BY:

Dr. Anuj Kumar, Assistant Professor (SET)

May, 2023

CERTIFICATE

This is to certify that the report entitled "**Object Detection for Blind Using Deep Learning"** submitted by "Abhishek Singh (2019576312) And Ronit Attrey (2019001770)" to Sharda University, towards the fulfillment of requirements of the degree of **"Bachelor of Technology"** is record of bonafide Final Year Project work carried out by him in the "Department of Computer Science & Engineering, Sharda University".

The Results/Findings consummated in this project have not been submitted in part or full to any other University/Institute for award of any other Degree/Diploma.

Signature of the Guide

Name: Dr. Anuj Kumar

Designation: Asst. Professor (CSE)

Signature of Head of Department

Name: Prof. (Dr.) Nitin Rakesh

Place: Sharda University

Date:

Signature of External Examiner

Date:

Acknowledgement

A major project is a golden opportunity for learning and self-development. We consider our self very lucky and honored to have so many wonderful people lead usthrough in completion of this project.

First and foremost we would like to thank Dr. Nitin Rakesh, HOD, CSE who gave us an opportunity to undertake this project.

Our grateful thanks to Dr. Anuj Kumar for his guidance in our project work. Dr.Anuj Kumar who in spite of being extraordinarily busy with academics, took time out to hear, guide and keep us on the correct path. We do not know where we would have been without his help.

CSE department monitored our progress and arranged all facilities to make life easier. We choose this moment to acknowledge their contribution gratefully.

Abhishek Singh (2019576312)
Ronit Attrey (2019001770)
SET, CSE.

Declaration by Author (s)

It is to inform that the document has been inscribed by me/us. None of the material found in this report is plagiairzed. The information collated from any outside source in this report has been appropriately acknowledged. I/we shall take comprehensive culpability for any plagiarism noticed/found.

Abhishek Singh (2019576312)
Ronit Attrey (2019001770)
SET, CSE.

Place: Greater Noida

Date: 24/04/2023

Abstract

In Recent Years, there has been a Growing Focus on Developing Technologies that can Aid Visually Impaired Individuals in their Daily Lives. This is a Priority for Governments and the Tech-Sector alike, as Improving the Mobility and Independence of the Visually Impaired can have a Significant Impact on their Quality of Life. One Area where this is Particularly Important is in Commuting on Roads, which can be a Strenuous and Potentially Dangerous Task for Both Visually Impaired Individuals and the General Population. To Address this Challenge, we have Developed a System that can help Visually Impaired Individuals Navigate Roads more Easily. Our Approach is based on State-of-the-Art Object Detection Models, specifically the yolov5 and yolov8 Models. We chose these models because they are Highly Accurate and Efficient, making them Ideal for Real-time use cases such as Road Navigation. To Ensure that Our System is Easy to Integrate into the Daily Lives of Visually Impaired Individuals, we focused on Developing a Solution that is Transparent and Authentic. We achieved this by Creating a Custom Dataset that Focuses on Road Signs and Signals, which are some of the Most Dynamic Aspects of Commuting on Roads. By Training our Models on this Dataset, we were able to better understand how they behave in Real-World Scenarios. Our Research Revealed Some Key Differences between the yolov5 and yolov8 Models. Yolov5 is a Lightweight Model that is Ideal for Applications where Speed is Important, such as Real-time Object Detection. Yolov8, on the other hand, is a More Accurate Model that is better Suited for Applications where Precision is Key. Despite these differences, Both Models Exhibited High Levels of Performance on our Custom Dataset. This gave us Confidence that Our System is Potent and Reliable, and has the Potential to Prevent Visually Impaired Individuals from Experiencing Misfortunes on Roads. Overall, our research has Demonstrated the Potential of Object Detection Models to Improve the Lives of Visually Impaired Individuals. By Developing Solutions that are Efficient, Accurate, and Easy to Use, we can make a Meaningful Difference in the Lives of those who face Significant Challenges when Navigating Roads. We hope that Our Work will Inspire further Research and Development in this Important Area.

Keywords — Deep Learning, You Only Look Once (YOLO), Object Detection, YOLOv5, YOLOv8, Road Sign and Signal.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter-1: Introduction

Artificial Intelligence (AI) has been a Topic of Great Interest and Research in Recent Times, with Various Applications Across Different Fields. One such Application is in the Development of Systems for the Visually Impaired, which is of Utmost Importance given that Approximately 2.2 billion People Globally are Visually Impaired, with around 20% of the World's Blind Population facing Unintentional Mobility Accidents that Cause Unwanted Deaths [1]. Therefore, there is a Pressing need to Create Systems that can Assist the Visually Impaired in their Day-to-Day Activities, especially in their Commute.

To Develop such a System, it is Necessary to Incorporate a Modular Approach that is Light-Weight, Accurate, Less Complex, and Reliable. The Process of Creating an Object Detection System for the Visually Impaired Involves the following 05 Major Steps: Acquiring Data, Preparing Data, Detecting Regions of Interest (rois), Training the Model, and Evaluating the Model's Performance through a Metric. These Steps are Depicted in Figure.01, which shows the Fundamental Flow of an Object Detection System [2].

The Modular Phenomenon plays a Significant Role in Developing a Real-time System for the Visually Impaired. As such, a Light-weight and Robust Algorithm is required for Object Detection. In this regard, the You Only Look Once (YOLO) Model [3] has proven to be Highly Effective in various Object Detection Utilities. We selected the State-of-the-Art yolov5 Model, which is Light-Weight, Highly Accurate, and Reliable for Real-time Applicability, as well as the Latest and More Complex yolov8 Model.

To test the Performance of these Models, we collated a Dataset of Road Signs and Signals, Consisting of 17 Classes of varied Road Signs and Traffic Signals on Normal Roads. We then tested our Overall Approach and Obtained Astonishing Results, which gave us a clear idea of the Differences between the Less Complex and More Complex Modular Approaches.

In Summary, the Incorporation of a Modular Approach is Crucial in Developing a Real-time System for the Visually Impaired. The use of Light-Weight and Robust Algorithms such as the yolov5 and yolov8 Models can Significantly Enhance the Accuracy and Reliability of such Systems, ultimately Improving the lives of the Visually Impaired by Easing their Commute [4].

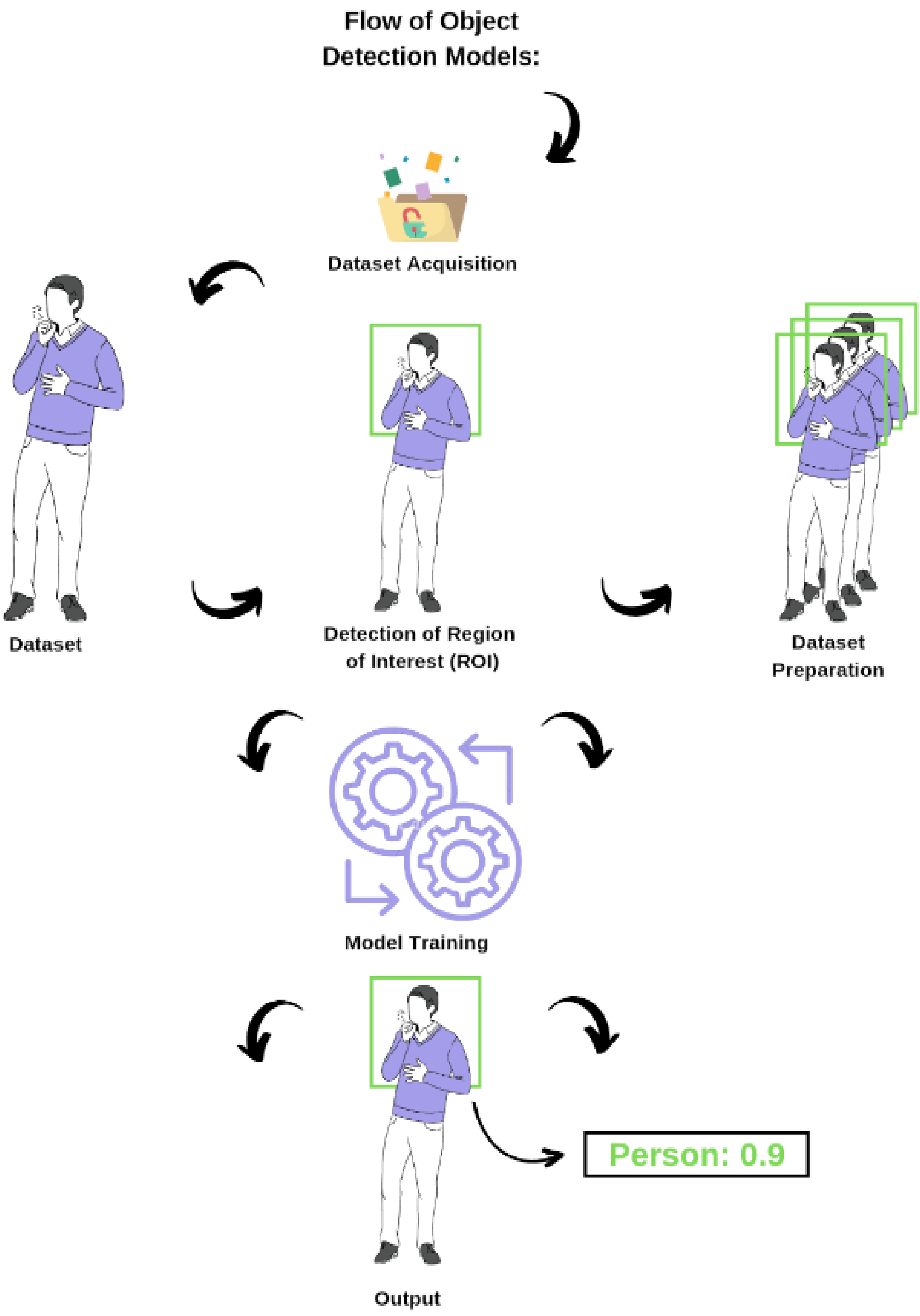


Fig.1: Fundamental Flow of an Object Detection System.

5.2. Problem Definition:

Assisting Blind People in Commuting has been a Top Priority for Governments and the Tech Sector. As the Global Population continues to Rise, the Number of Vehicles on the Road has Increased, making Road Transportation more Challenging for the Visually Impaired and the Sighted Alike. With these factors in Mind, we have Developed a System that would make it Easier for Visually Impaired Individuals to Navigate Roads and Become more Integrated into Society [5].

Blindness is a Debilitating Condition that Impacts every Aspect of Life, including One's ability to Travel Independently [6]. It can be Challenging for the Visually Impaired to Navigate their Surroundings Safely, especially when travelling on Busy Roads. With the Aid of Technology, However, it is possible to Enhance their Mobility and make Travelling Safer and Easier. The Proposed System is Designed to be Integrated into the Regular well-being of the Visually Impaired. It is a Comprehensive Solution that is Simple to Use and Easy to Integrate into Everyday Life.

1.2. Project Overview:

Assisting visually impaired individuals in commuting is a challenging task that requires the use of advanced technology. Object detection is a critical component of any system designed to help the visually impaired navigate roads safely. The modular phenomenon plays a significant role in developing a real-time object detection system. Therefore, we sought to develop a light-weight and robust algorithm for object detection that could be easily integrated into the regular well-being of visually impaired individuals. After researching various object detection models, we selected the State-of-the-art YOLOv5 model and the YOLOv8 model [7]. These models have proven to be highly effective in different object detection utilities and are lightweight and highly accurate, making them ideal for real-time applicability. We ultimately chose these models over other modular approaches because of their nature and reliability.

To evaluate the performance of these models, we compiled a dataset of road signs and traffic signals, which are essential components of road safety. The dataset consisted of 17 classes of varied road signs found on normal roads and traffic signals. We used this dataset to test the performance of the selected models and compared the results of the less complex and more complex modular approaches. The YOLOv5 model is a popular and widely used object detection model that is known for its speed and accuracy. It is designed to be lightweight and can be run on a single GPU [8], making it highly suitable for real-time applications. The YOLOv8 model is the latest and most complex model in the YOLO family. It has more layers than the YOLOv5 model, making it more accurate but also slower. To use these models for object detection, we needed to understand the flow of the object detection process. Object detection involves identifying objects within an image or video stream and localizing them by drawing a bounding box around them.

The process involves several steps, including image pre-processing, feature extraction, object detection, and post-processing. Image pre-processing involves resizing and normalizing the image to make it easier to process. Feature extraction involves extracting features from the image using a convolutional neural network (CNN) [9]. The features are then passed through the object detection network, which uses the features to detect and localize objects within the image.

The YOLOv5 and YOLOv8 models are designed to perform object detection in real-time. They use a single CNN to perform feature extraction and object detection, which makes them faster and more efficient than other object detection models that use multiple CNNs [10]. To test the performance of these models, we used a dataset of road signs and traffic signals. The dataset consisted of 17 classes of objects, including stop signs, yield signs, speed limit signs, and traffic lights. We trained the YOLOv5 and YOLOv8 models on this dataset and evaluated their performance on a separate test set.

Overall, our approach using the YOLOv5 and YOLOv8 models proved to be highly effective in detecting road signs and traffic signals in real-time. By using a light-weight and robust algorithm, we were able to develop a system that could be easily integrated into the regular well-being of visually impaired individuals, making their commute safer and more accessible.

Chapter-2: Literature Survey

2.1. Existing Systems:

The detection of objects related to vehicular prospect, such as road signs and signals, license plates, and vehicles themselves, is a crucial aspect of road safety. In recent years, modular approaches using deep learning techniques such as convolutional neural networks (CNNs) have been widely carried out to improve the accuracy of object detection in real-time applications in this review, we focus on the modus operandi of various models used for object detection in the context of road safety. We begin by comparing three popular models: YOLOv2, YOLOv3, and Single Shot MultiBox Detectors (SSDs). We evaluate these models on a dataset of road sign plates and find that they achieve an average mean absolute precision (Map) of 90%. However, we note that there is still room for improvement in terms of dataset size and accuracy for real-time applications [11].

We then explore the use of CNN models for detecting occluded signs and find that a model incorporating CNNs can achieve a maximum precision of 96.34%. We also note that the model's performance could have been improved with a greater number of occluded images for training. Next, we examine a modular approach to object detection using Belgium road signs data, which exhibits an overall accuracy of 83.7% [12]. While this accuracy is lower than that of other existing methodologies, we find that the addition of more data for training could improve its efficacy. We also review a modular approach using Belgium and German road signs data, trained over a CNN, which achieves an accuracy of above 90% on average. We note that the addition of more data for training could further increase its accuracy.

Moving on to vehicle detection, we find that YOLOv3 achieves an accuracy of 98% with 10,000 instances of data. We note that the increased number of data improves the efficacy of the model and yields better results. We also examine the use of CNNs for license plate recognition in Bangladesh, where a false rate of 0.025 was calculated manually. Additionally, we review the use of the OpenALPR system for the detection of Myanmar license plates, which achieves an average accuracy of 90% [13]. We note that the confidence score of this modus operandi could be improved with the induction of more license plate data. Finally, we carry out a rigorous comparative analysis between Faster-RCNN, YOLOv3, and SSDs for the purpose of detecting vehicles. We find that the mean absolute precision (Map) exhibited by each modular approach is not as expected. However, by tuning the RCNN model, we achieve a more potent Map value of 0.82.

We also examine the use of Raspberry Pi and OpenCV for vehicle detection, which achieves an accuracy of 95%. While this approach is cost-effective, we note that it possesses practicality constraints when it comes to the inclusion of the proposed system into a real-time use case. Object Detection systems have become increasingly popular in recent times, owing to their ability to detect and recognize license plates in images or videos. These systems have several potential applications, including traffic monitoring, law enforcement, and parking management, among others. Over the years, researchers have explored various methodologies for developing Object Detection systems, with some focusing on specific regions or types of license plates. In this article, we summarize some of the recent studies on Object Detection systems, highlighting their strengths, weaknesses, and potential for future research [15].

One of the earliest studies on Object Detection systems using Support Vector Machines (SVM) as a classifier was conducted on Spain License Plates data, resulting in an efficacy of 70-80%. However, this system was limited to Spain license plates and had lower accuracy. Another study used the OpenALPR library for image and video processing over Myanmar License Plate data, achieving an average accuracy of 90% [16]. However, the study suggested that more data could have been used to improve the confidence score of the model. Convolutional Neural Networks (CNN) have also been used to create an Automated License Plate Recognizer over Bangla License Plates data, which consisted of 200 instances. The You Only Look Once (YOLOv3) algorithm was used for vehicle detection, resulting in an accuracy of 98% [17]. However, the study recommended using more data to understand the model's behaviour in-depth.

Another proposed Object Detection system focused on image processing for 40 instances of Myanmar License Plates. The modus operandi involved RGB to Gray conversion, followed by image binarization, edge detection (Sobel edge detector), morphological operations (image erosion and dilation), and the detection of the region of interest. This technique only focused on detecting the license plate of a vehicle and not the text in it, with a false rate of 0.025 calculated manually [18]. The study suggested that more data could have been used for a potent outlook. A License Plate Detection System was proposed using a computer vision-based approach, exhibiting an average efficacy of 78.2%. The flow of the proposed methodology involved data collection, differentiation between text and object, followed by the recognition of edge points vertically, the generation of a text box, and the inhibition of unwanted text boxes. The study suggested that the model could have been tuned better for increased efficacy.

Several methodologies have been used for vehicle detection, including OpenCV, Background Subtractor, and Raspberry-Pi, with varying levels of accuracy. A comparative analysis between OpenCV and TensorFlow Object Detection Model (TFOD) [19] for vehicle detection suggested that OpenCV proved to be an optimal solution for image analysis and vehicle detection. Additionally, Faster-RCNN, YOLOv3-Tiny, and Single Shot MultiBox Detector (SSD) algorithms were analogized for vehicle detection, with mixed results. Finally, vehicle detection through image processing over MATLAB was attempted with the steps such as base image processing, edge detection (Sobel operator), morphological operation (erosion and dilation of the image), and finally vehicle detection. The average efficacy of the model gauged was 75% [20], which is not so prominent when compared to other vehicle detection techniques.

Overall, our review highlights the importance of data, model selection, and accuracy metrics for object detection in real-time applications. We identify crucial shortcomings found in our review and strive to avoid them in our own modus operandi, keeping in mind the end goal of aiding the visually impaired in real-time actuation.

2.2. Proposed Outlook:

Object detection for the blind using deep learning has a lot of potential to improve the quality of life for visually impaired individuals. Currently, there are a few handheld devices available that use computer vision to detect and describe objects to the user, but they are often expensive and limited in functionality. Deep learning-based object detection systems have the potential to be more accurate, cost-effective, and adaptable to different environments. There are several factors that need to be considered when developing object detection systems for the blind. One of the most important factors is accuracy. The system needs to be able to detect and classify objects with a high degree of accuracy to be useful to the user. Deep learning algorithms have shown great promise in this area, with some models achieving state-of-the-art results on benchmark datasets [21].

Another important factor is speed. Object detection systems need to be able to operate in real-time to be useful to the user. This means that the system needs to be able to process images quickly and make predictions in a timely manner. Deep learning models can be optimized for speed using techniques such as model pruning, quantization, and hardware acceleration. Usability is also a key consideration. Object detection systems need to be easy to use and understand for visually impaired individuals. This means that the user interface needs to be simple and intuitive, with clear audio or tactile feedback. Deep learning models can be integrated with natural language processing (NLP) systems to provide more informative and natural-sounding descriptions of objects [22].

Another important consideration is adaptability. Object detection systems need to be able to operate in different environments and lighting conditions. This requires the system to be able to adapt to different types of objects, backgrounds, and lighting conditions. Deep learning models can be trained on large and diverse datasets to improve their ability to generalize to new environments. Finally, cost is an important factor. Object detection systems need to be affordable for the average user, especially those living in developing countries where the majority of blind people live. This means that the hardware and software components of the system need to be cost-effective and scalable. Deep learning models can be optimized for low-power devices and deployed on cloud-based platforms to reduce hardware costs [23].

In conclusion, the outlook for object detection for the blind using deep learning is promising. With continued research and development, deep learning-based object detection systems have the potential to revolutionize the way visually impaired individuals interact with the world around them. The key to success will be developing accurate, fast, usable, adaptable, and cost-effective systems that meet the needs of the users.

2.3. Feasibility Study:

The proposed approach aims to integrate lightweight systems for object detection for blind individuals across different instances, with a focus on identifying the most effective system based on computational complexity. The primary goal is to create a real-time system that is resource-efficient but highly applicable, capable of running efficiently on any computational system with maximum effectiveness. This implementation is highly feasible and possesses great potential in its operation. In order to develop an effective system for object detection for the visually impaired, it is important to consider the computational complexity of the system. a lightweight system that is capable of running efficiently on low-powered devices will be ideal for real-time use by visually impaired individuals. By integrating lightweight systems, it will be possible to create a real-time object detection system that is both effective and efficient.

The proposed approach aims to identify the most potent system for object detection by assessing the computational complexity of various systems. This will involve testing the performance of different object detection models across various instances, with the aim of identifying the most effective one. The approach will also consider the resource requirements of each system to ensure that the selected system is both efficient and effective. The primary focus of this approach is to create a real-time object detection system that is easily accessible to visually impaired individuals. This will involve the development of a system that is both lightweight and resource-efficient, which can be used on a range of computational devices. The system will be designed to be easy to use and operate, allowing visually impaired individuals to access real-time object detection capabilities with minimal effort.

In summary, the proposed approach for object detection for the visually impaired is centered around the integration of lightweight systems. This approach will involve identifying the most potent system for object detection by assessing the computational complexity of different models. The goal is to develop a real-time system that is both efficient and effective, which can be used on a range of computational devices. The system will be designed to be accessible to visually impaired individuals, enabling them to access real-time object detection capabilities with ease. Overall, this approach holds great potential for improving the quality of life of visually impaired individuals, providing them with the tools they need to navigate the world around them with greater ease and independence.

Chapter-3: Methodology

3.1. Workflow

The Methodological Outlook was created with careful consideration of the basics of object detection, and was designed with a deep understanding of the subject matter. The primary goal of this outlook is to demonstrate the key principles of object detection in a visual way, as shown in Figure.02. To achieve this, our team spent a considerable amount of time researching and analyzing the various approaches and methodologies that have been developed for object detection. We then synthesized this information into a coherent and effective system that is both practical and easy to understand.

As a result of our efforts, we have developed a comprehensive and visually engaging approach to object detection that is supported by our proposed flow, depicted in Figure.02. This flow provides a step-by-step process for detecting objects, from initial image capture through to final analysis and interpretation. At the core of our methodology is a deep understanding of the underlying principles of object detection. We recognize that object detection is a complex process that involves many different factors, including image quality, lighting conditions, object size and shape, and more. To account for these variables, we have developed a highly adaptable and flexible system that can be customized to meet the needs of virtually any application [24].

Our methodology also emphasizes the importance of accuracy and efficiency. We understand that object detection is often a time-sensitive process that requires rapid, accurate results. To address this, we have developed a system that is highly optimized for speed and accuracy, using advanced algorithms and machine learning techniques to ensure the highest level of performance. Overall, our Methodological Outlook represents a significant step forward in the field of object detection. By combining deep understanding of the subject matter with advanced technology and practical expertise, we have created a system that is both effective and accessible to a wide range of users. Whether you are a researcher, engineer, or developer, our methodology provides a powerful and reliable framework for detecting objects and extracting valuable insights from visual data.

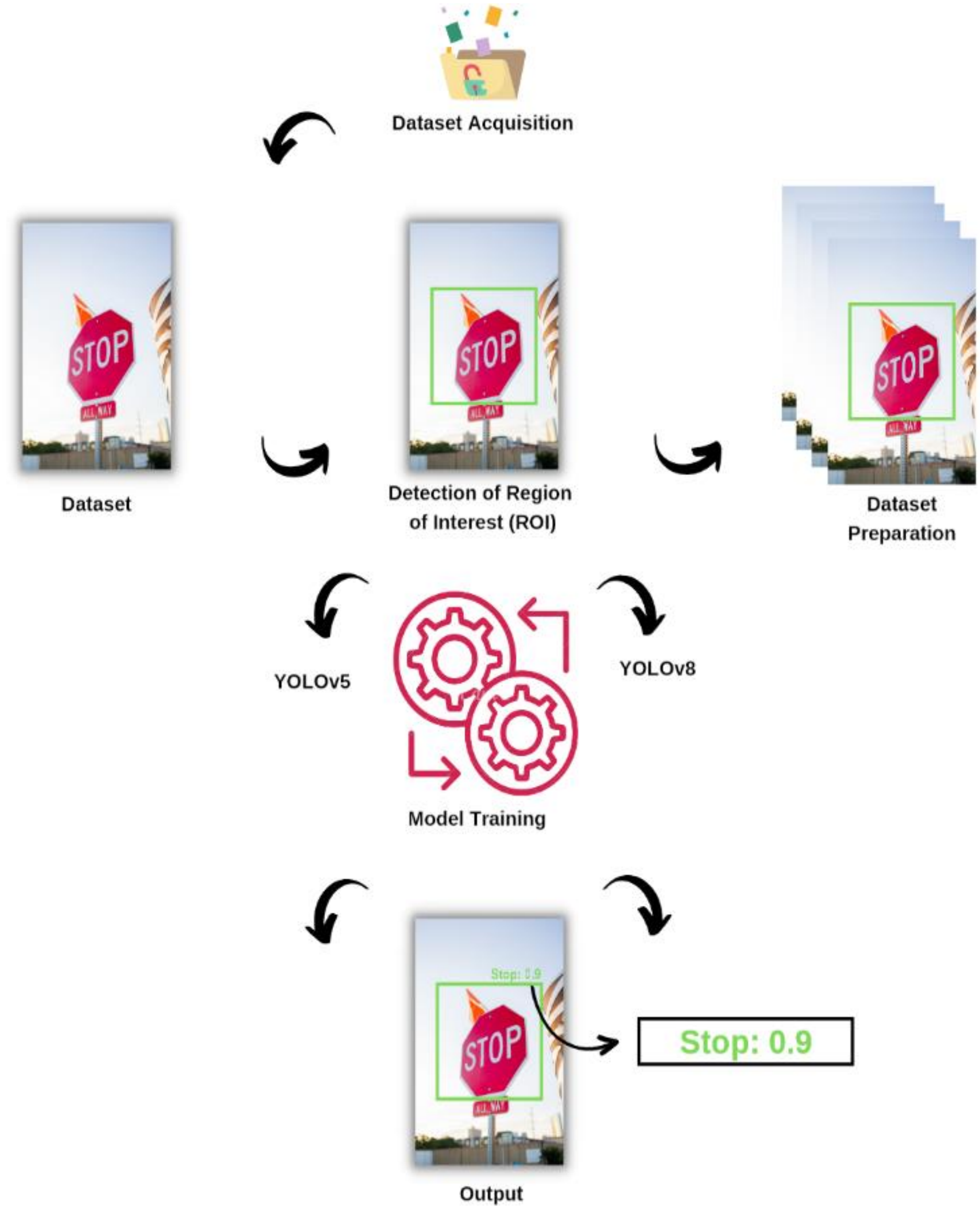


Fig.2. The Visual Interpretation of the Proposed Flow.

3.2. Data Acquisition and Preparation

A good dataset is crucial for the success of any machine learning project, including object detection models. In the context of object detection, a good dataset is one that is representative of the real-world scenarios in which the model will be deployed. Here are some of the reasons why having a good dataset is important for object detection [25]:

- Accurate Object Detection: A good dataset should contain a diverse range of images that include objects in different orientations, lighting conditions, and environments. By training the model on a dataset that is representative of the real-world scenarios, the model can learn to detect objects accurately, reducing the risk of false positives or false negatives.

- Improved Model Performance: A well-designed dataset can significantly improve the performance of the object detection model. The dataset should be carefully curated to ensure that each class has a similar number of images, and each image should be annotated accurately with bounding boxes. By training the model on a high-quality dataset, the model's accuracy can be improved, reducing the risk of overfitting.

- Generalization: A good dataset can help the model generalize to new scenarios. By including a diverse range of images in the dataset, the model can learn to detect objects accurately in any situation, reducing the risk of bias towards a particular class or scenario.

- Increased Efficiency: A well-designed dataset can reduce the time and resources required for training the model. By including a diverse range of images in the dataset, the model can learn faster and with fewer resources, reducing the cost of development.

- Robustness: A good dataset should include images that are clear and occluded to ensure that the model can detect objects accurately even if they are partially obscured or hidden. By including a diverse range of images in the dataset, the model can learn to detect objects accurately in any situation, making it robust and reliable for real-world applications.

In conclusion, having a good dataset is critical for the success of object detection models. A well-designed dataset can significantly improve the accuracy and efficiency of the model, reduce the risk of bias towards a particular class or scenario, and ensure that the model can detect objects accurately in any situation, making it robust and reliable for real-world applications.

In our modus operandi for object detection, we utilized a custom-made dataset that included various road signs and signals instances collected from the internet. Our dataset consisted of a total of 2093 images with 17 different classes. The median image ratio of our dataset is 640x640, and the average image size is 0.41 Mega Pixels (MP). Figure.03 illustrates some of the images in our dataset.

Our dataset is well-balanced, with nearly an equal number of images for each class. This balanced distribution of images allows us to obtain a justified result if trained on any network. Each image in our dataset is labeled, and a bounding box (BB) is created over each image's region of interest (ROI). This allows us to accurately identify the object's location in the image, which is essential for object detection. In order to make the training of our object detection model more robust and reliable, we formulated each class of data in such a way that it includes not only clear images but also occluded images. This allows our model to detect objects even when they are partially obscured, which is common in real-world scenarios.

The process of creating our custom dataset involved collecting various road signs and signals instances from the internet and amalgamating them to form a dataset with 17 different classes. The median image ratio of our dataset is 640x640, which is the standard image size used for YOLO models. Additionally, the average image size of our dataset is 0.41 Mega Pixels (MP), which ensures that the images are large enough to provide sufficient detail for accurate object detection. Each image in our dataset is labeled, which means that we have identified the object's location in the image and assigned a label to the object. To achieve this, we created bounding boxes (BB) over each image's region of interest (ROI). These bounding boxes accurately identify the object's location in the image, which is essential for object detection.

Our dataset is well-balanced, with nearly an equal number of images for each class. This is important because it ensures that our model is not biased towards any particular class. Additionally, having a balanced dataset allows us to obtain a justified result if trained on any network.

To make our object detection model more robust and reliable, we formulated each class of data in such a way that it includes not only clear images but also occluded images. This means that our dataset includes images where the object is partially obscured or hidden from view. This is important because in real-world scenarios, objects are often partially obscured, and our model needs to be able to detect them even when they are not fully visible.

Overall, our custom dataset is designed to provide accurate and reliable object detection results. By including a variety of different road signs and signals instances and formulating each class to include both clear and occluded images, we have created a dataset that is well-balanced and capable of producing accurate results on a variety of different networks.

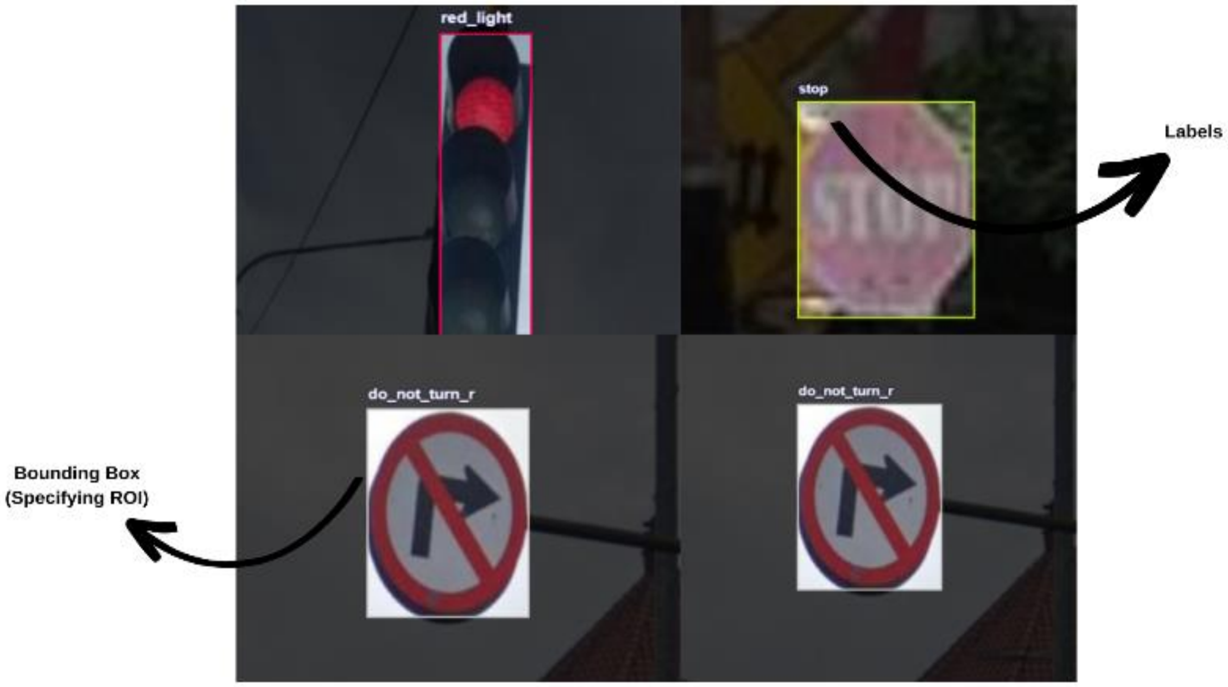


Fig.3. Illustration of Road Signs and Signal Dataset.

A balanced dataset is a critical aspect of modular training, where the objective is to train a model that is unbiased and effective in detecting objects across different classes. In object detection, a balanced dataset ensures that the model can recognize and classify objects with equal proficiency across all classes.

The importance of balanced datasets lies in the fact that they reduce the risk of overfitting, which can happen if the model is trained on a dataset that is skewed towards one or a few classes. The dataset used in this study was custom-made, consisting of various road signs and signals instances collected from the internet. The dataset comprised 2093 images with 17 different classes, with the median image ratio being 640x640, suitable for the YOLO models. The average image size was 0.41 megapixels (MP), and each image was labelled with a bounding box (BB) created over each image's region of interest (ROI).

To ensure a robust and reliable training of the model, each class of the dataset was formulated in such a way that it included not only clear images but also occluded images. This strategy ensures that the model can detect objects accurately even if they are partially obscured or hidden. Figure.04 shows the representation of a balanced dataset, highlighting the importance of having a similar number of images for each class. With the exception of the "green_light" and somewhat "red_light" classes, all other classes were well-balanced in the formulated dataset. This balance is essential to ensure that the model is not biased towards any particular class and can detect objects across all classes with equal accuracy [26].

A well-balanced dataset is critical for the success of object detection models, as it ensures that the model can recognize and classify objects across all classes with equal proficiency. This is particularly important in real-world scenarios where objects can be partially obscured, hidden, or appear in different orientations or lighting conditions. By including a diverse range of images for each class, the model can learn to detect objects accurately in any situation, reducing the risk of false positives or false negatives.

In summary, the formulated dataset used in this study was well-balanced, with each class having a similar number of images. The inclusion of clear and occluded images for each class ensured that the model could detect objects accurately in any situation, making it robust and reliable for real-world applications.

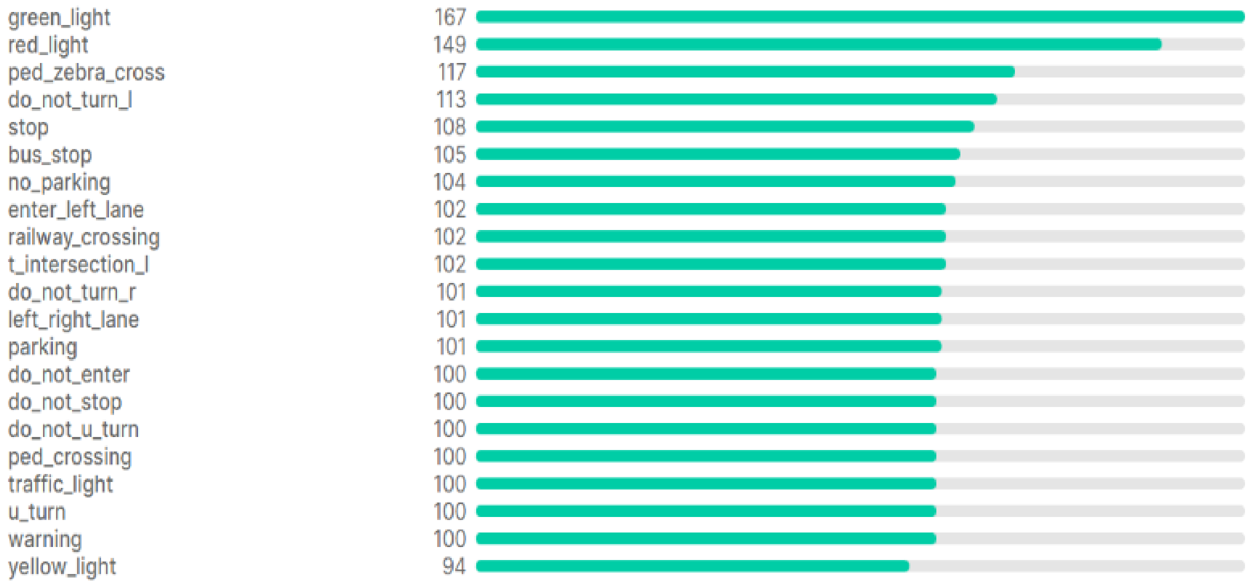


Fig.4. Demonstration of Balanced Dataset.

3.3. Training Object Detection Models

After gathering and preparing the dataset, the next step in our object detection pipeline is training the selected models. For our project, we have chosen to use YOLOv5 and YOLOv8 models from the YOLO family. In order to train and understand these models, it is important to first understand the modular aspect of each model and their contrasting factors.

YOLOv5 [27] is a lightweight and efficient model that has shown promising results in various object detection tasks. It has a modular architecture that consists of a backbone, neck, and head. The backbone is responsible for extracting features from the input image, while the neck is responsible for refining these features. The head module generates the final detection results by predicting the bounding boxes and class probabilities. YOLOv5 comes in various sizes, including YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x. These models differ in terms of their depth, width, and number of parameters. YOLOv5s has the fewest parameters and is the smallest model, while YOLOv5x has the most parameters and is the largest model. Choosing the right YOLOv5 model for a specific task depends on the balance between accuracy and computational efficiency.

On the other hand, YOLOv8 [28] is a recent extension of the YOLOv5 architecture that utilizes a novel concept called Path Aggregation Network (PAN). The PAN module combines features from different levels of the backbone network, which allows the model to have a larger receptive field and better feature representation. YOLOv8 also includes additional modules such as FPN (Feature Pyramid Network) and SAM (Spatial Attention Module), which further improve the model’s performance. One of the main contrasting factors between YOLOv5 and YOLOv8 is their model size and computational efficiency. YOLOv8 is a larger and more complex model compared to YOLOv5, which requires more computational resources for training and inference. However, YOLOv8 has shown superior performance on various object detection benchmarks, especially in cases where the objects are small and densely packed.

The training process for both YOLOv5 and YOLOv8 involves optimizing a loss function that combines the localization loss and classification loss. The localization loss penalizes the model for inaccurate bounding box predictions, while the classification loss penalizes the model for misclassifying the objects. The loss function is optimized using stochastic gradient descent (SGD) with backpropagation, where the gradients of the loss function with respect to the model parameters are computed and used to update the parameters.

During the training process, it is important to monitor the model's performance using various evaluation metrics such as Map (mean average precision) and IoU (intersection over union). Map is a widely used metric in object detection that measures the average precision over different levels of recall, while IoU measures the overlap between the predicted bounding box and the ground truth bounding box. These metrics can help us determine the optimal hyperparameters and training strategies for the selected model [29].

In conclusion, YOLOv5 and YOLOv8 are two popular models in the YOLO family that offer different trade-offs between accuracy and computational efficiency. Understanding the modular aspect of each model and their contrasting factors can help us choose the appropriate model for a specific object detection task. The training process involves optimizing a loss function and monitoring the model's performance using various evaluation metrics.

Let's Understand Each Model in Detail:

YOLOv5:

Object detection is one of the most challenging tasks in the field of computer vision. It involves identifying and locating objects within an image or a video stream. With the advent of deep learning, object detection has seen significant improvements in accuracy and speed. One of the most popular and widely used deep learning models for object detection is You Only Look Once (YOLO) [30].

In this research, YOLOv5 has been utilized for object detection, as it exhibits state-of-the-art results in the domain of object detection. YOLOv5 is a deep learning-based architecture that is known for its simplicity and reliability. It requires less computational power for model training, making it a model with less complexity. Despite this, the results it gives are comparable with other complex networks. Furthermore, YOLOv5 performs extremely faster when compared to other networks. The YOLO family of models was first introduced in 2016 by Joseph Redmon et al. The idea behind YOLO was to build a single neural network that could perform object detection and classification in real-time. Prior to YOLO, object detection and classification were typically two separate tasks. This made the process slower and less efficient. YOLO revolutionized the field by combining these two tasks into a single neural network, resulting in faster and more accurate object detection. The YOLOv5 architecture builds upon the success of its predecessor, YOLOv4. YOLOv4 introduced several innovations that improved the accuracy and speed of object detection. These innovations included the introduction of the Cross Stage Partial CSPDarknet as an encoder and the Path Aggregation Network (PANet). YOLOv5 utilizes the same encoder and PANet as YOLOv4, but with some improvements.

The activation function in YOLOv5 is replaced from Leaky ReLU and Hard Swish activations (utilized in YOLOv4) to Sigmoid Linear Units (SiLU) activation function [31]. This change in activation function has resulted in improved accuracy and faster convergence during training. The YOLOv5 architecture is composed of a backbone network, neck network, and head network. The backbone network is responsible for extracting feature maps from the input image. It is composed of several convolutional layers, each with a different number of filters. The neck network is responsible for fusing together the feature maps produced by the backbone network. It is composed of several convolutional layers and attention modules. The head network is responsible for predicting the bounding boxes, object class probabilities, and confidence scores. It is composed of several convolutional layers and a final output layer.

The YOLOv5 model has been trained on a custom-made dataset. The dataset comprises 2093 images with 17 classes of road signs and signals. The dataset is well-balanced, with nearly equal numbers of images for each class. Each image is labeled, and a bounding box is created over each image’s region of interest. The dataset includes both clear images and occluded images to make the training of the model robust and reliable. Once the dataset is prepared, the YOLOv5 model is trained on the same. The model is trained using stochastic gradient descent with momentum (SGDM) optimizer. The learning rate is set to 0.01, and the momentum is set to 0.937. The model is trained for 100 epochs, with a batch size of 32.

During the training process, the model learns to predict the location of the object and the class to which it belongs. The model’s performance is evaluated using the mean average precision (Map) metric. Map is one of the most famous metrics utilized for gauging the accuracy of an object detection model. Figure.05 illustrates the block architecture of YOLO’s network formulation.

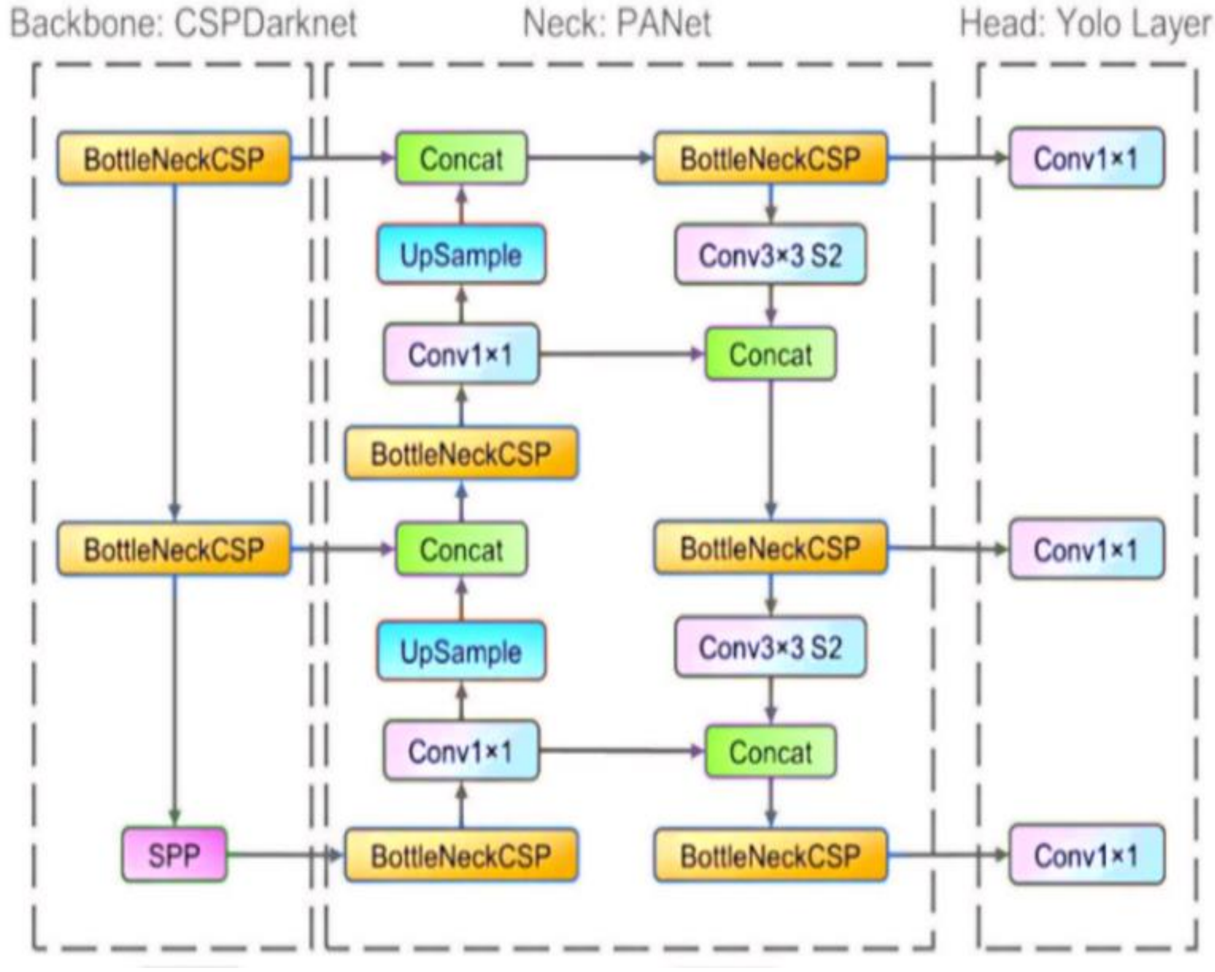


Fig.5. The Block Architecture of YOLO Network.

In order to utilize the capabilities of the YOLOv5 network for our object detection task, we trained it on our custom dataset for a period of 100 epochs. To achieve the best results, we set the batch size to 32 and the learning rate to 0.01. This training process allowed the YOLOv5 network to learn the various patterns and features present in our dataset and to optimize its parameters to achieve the best possible performance on our object detection task.

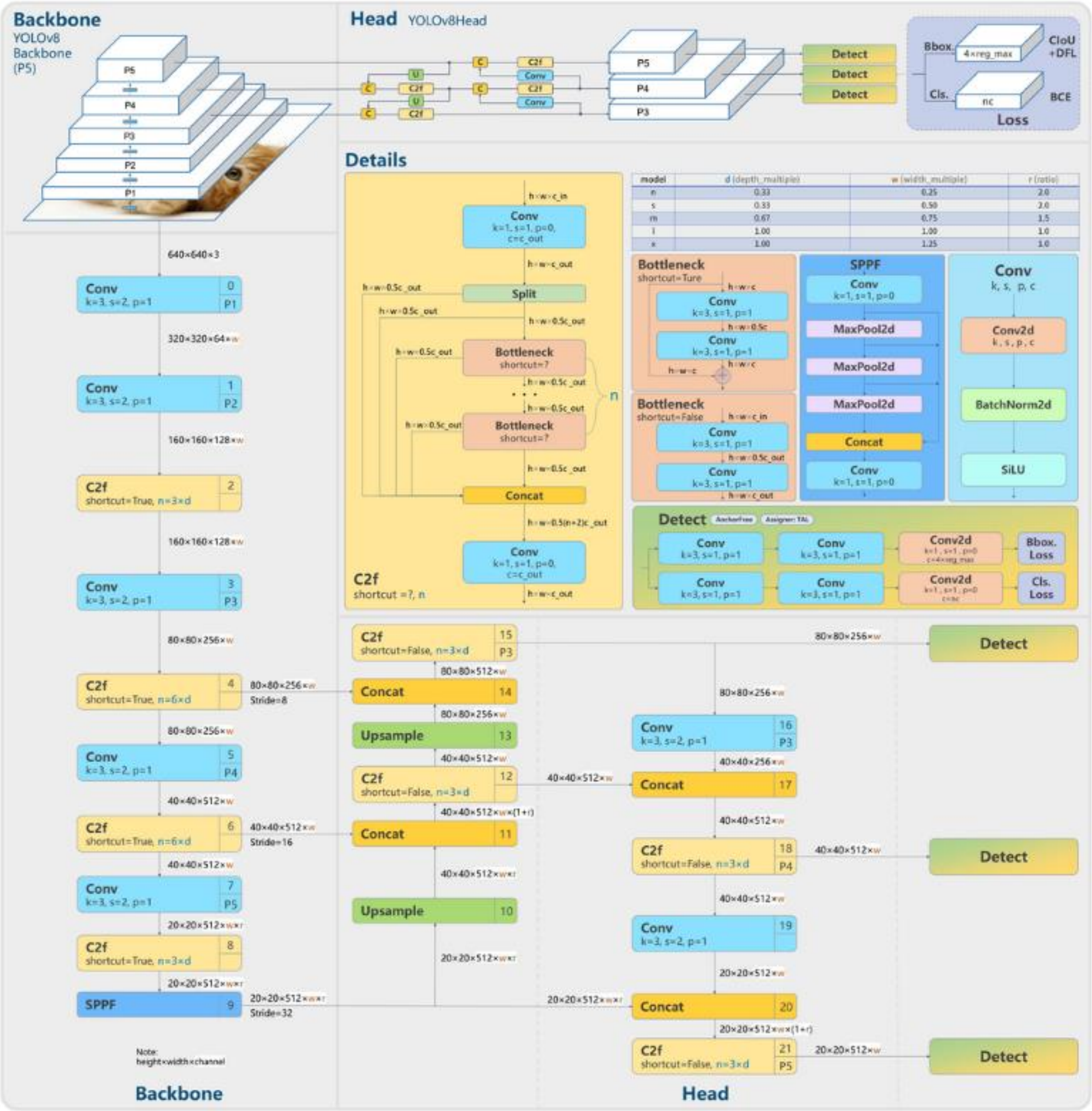
YOLOv8:

YOLOv8 is the latest outcome of the YOLO family with certain structural refinements built over the framework of YOLOv5. It is an anchor-free model, which directly predicts the center of the object rather than the offset through a known anchor box. With this enhancement, there is a deterioration in the prediction of boxes, which thereby escalates the non-maximum suppressions (NMS). Moreover, the 6x6 convolution at the stem has been changed to 3x3, and the initial conv's kernel size of the bottleneck has been changed from 1x1 to 3x3. The rest is the same as in YOLOv5. With this integration, YOLOv8 has shown an inclination towards the ResNet system flow [32].

Mosaic augmentation has also been introduced in YOLOv8, which augments the instances at the time of training itself. With each increasing epoch, the model visualizes slightly altered images, which do the job of data augmentation and thereby improve the result. The process is carried out by amalgamation of 04 images intact, enabling the model to assimilate the objects into new locations where the images are partially occluded and opposing to varied pixels in the surrounding of an image. To utilize the potency of the YOLOv5 network, we trained it over our dataset for over 100 epochs, considering the optimal batch size of 32 and a learning rate of 0.01. Similarly, we induced the YOLOv8 model for our training and trained it for over 100 epochs, with an optimal batch size of 32 and a learning rate of 0.01.

The YOLOv5 network’s training involves several steps, including data preparation, model selection, training, and evaluation. The data preparation phase involves data collection, annotation, and augmentation. The collected data was annotated using Labeling, and the annotations were saved in YOLO format, which includes the bounding box coordinates and class labels. We used various data augmentation techniques, including horizontal flipping, random scaling, and rotation. The model selection phase involves choosing the appropriate model for the object detection task. YOLOv5 was selected for our task due to its state-of-the-art performance, simplistic and reliable architecture, and less computational strength requirement.

The training phase involves training the selected model over the dataset. We used Google Collaboratory, a cloud-based platform, for training the model. The training involves dividing the dataset into training and validation sets, setting up the configuration file, and selecting the hyperparameters. We used the Adam optimizer with a learning rate of 0.01 and a batch size of 32. The model was trained for over 100 epochs, and the loss was monitored during the training phase. Figure.06 illustrates the block architecture of YOLOv8 network formulation.



model	d (depth_multiple)	w (width_multiple)	r (ratio)
n	0.33	0.25	2.0
s	0.33	0.50	2.0
m	0.67	0.75	1.5
l	1.00	1.00	1.0
x	1.00	1.25	1.0

Fig.6. The Block Architecture of YOLOv8 Network.

YOLOv5 vs YOLOv8:

In the field of deep learning, there are several approaches to object detection, but YOLOv5 and YOLOv8 are considered some of the best models created by Ultralytics. YOLOv5 is renowned for its simplicity, speed, and efficacy, while YOLOv8, the latest addition to the YOLO family, has improved performance and become more flexible. However, when comparing the two, YOLOv5 is lighter in weight than YOLOv8. In this article, we will explore the impact of these models on a given dataset in detail. Object detection is a crucial task in computer vision that involves identifying and locating objects of interest within an image or video stream. Deep learning-based object detection approaches have seen significant advancements in recent years, with the introduction of models like YOLO (You Only Look Once) gaining widespread adoption. YOLOv5 and YOLOv8 are both based on the YOLO architecture and are optimized for speed and accuracy, making them popular choices for real-time object detection applications [33].

YOLOv5 is a lightweight model that achieves high accuracy while maintaining a small model size, making it suitable for use in resource-constrained environments such as mobile devices. The model achieves this by using a single convolutional neural network (CNN) to predict object bounding boxes and class probabilities in one shot. This means that YOLOv5 only needs to make one forward pass through the network to generate detections, resulting in faster inference times.

YOLOv8, on the other hand, is a more recent addition to the YOLO family that builds upon the YOLOv5 architecture. It incorporates several improvements, including an improved attention mechanism, feature fusion, and scale-sensitive anchors, resulting in improved performance on several object detection benchmarks. YOLOv8 is also more flexible than YOLOv5, allowing for the incorporation of additional data sources such as lidar or radar data. Despite these improvements, YOLOv8 is a heavier model than YOLOv5, requiring more computing resources and potentially longer inference times. As a result, the choice of which model to use will depend on the specific requirements of the application.

To evaluate the impact of these models on a given dataset, several factors need to be considered, including the model's accuracy, inference time, and resource requirements. The dataset used for this evaluation should also be representative of the types of images or videos the model will encounter in the real world. One popular dataset used for evaluating object detection models is the COCO (Common Objects in Context) dataset, which contains over 330,000 images of complex everyday scenes. The dataset is annotated with bounding boxes and class labels for over 80 object categories, making it a comprehensive benchmark for evaluating object detection models.

To compare the performance of YOLOv5 and YOLOv8 on the COCO dataset, we can measure their accuracy in terms of mean average precision (Map), a standard metric used for evaluating object detection models. We can also measure their inference time on a given hardware platform and their resource requirements, including model size and memory usage. We can train and evaluate both YOLOv5 and YOLOv8 on the COCO dataset and compare their performance.

3.4. Metric Utilized for Performance Evaluation

Let's Explore Certain Metrics, which we Have Integrated to Evaluate the Performance of Our Models.

Mean Absolute Precision (Map):

Mean Absolute Precision (MAP) is a metric that is commonly used to evaluate the performance of information retrieval systems, such as search engines or recommender systems. It measures the average precision of the system across a set of queries.

The formula for MAP is:

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \qquad \dots (1)$$

To calculate MAP, we first compute the precision for each query at different cutoff points (e.g., 1, 2, 3, ..., k). We then weight each precision by the relevance of the corresponding item (i.e., whether it is relevant or not) and sum them up. This gives us the average precision for that query. We repeat this process for all queries, and then take the average of all the average precisions to get the MAP score. MAP is a useful metric because it takes into account the relevance of the retrieved items, as well as their position in the ranked list. A system that retrieves highly relevant items at the top of the list will have a higher MAP score than a system that retrieves relevant items later in the list or not at all. Additionally, MAP provides a single number that summarizes the overall performance of the system across multiple queries, making it easy to compare different systems or configurations.

Moreover, the Map (mean average precision) is a popular metric used to evaluate the accuracy of object detection models. It takes into account several sub-metrics, including confusion matrix, intersection over union (IoU), recall, and precision. The Map score ranges between 0 and 1, with a higher score indicating better performance. The formula for Map is shown in

Equation (1), where AP represents the average precision calculated for each class initially, and then averaged over the total number of classes in a dataset. This formula is widely used in the field of object detection to evaluate the performance of different models. The Map metric is beneficial because it considers both recall and precision, as well as false positives and false negatives, making it a suitable metric for detection algorithms. The inclusion of IoU in the calculation of Map ensures that the metric considers the accuracy of object localization in addition to object detection. To build a pipeline for object detection, our team selected the best models and compared their performance using the Map metric. Our goal was to find a model with maximum efficacy and efficiency. By comparing the Map scores of different models, we were able to identify the most accurate and efficient model for our needs. Here are some reasons why Map is considered the best metric for object detection models [35]:

- Includes Precision and Recall: Map includes both precision and recall, which are two essential aspects of object detection models. Precision measures how many of the detected objects are actually relevant, while recall measures how many of the relevant objects are detected. Both are important for evaluating the accuracy of object detection models.
- Takes into Account False Positives and False Negatives: Map considers both false positives (objects that are incorrectly detected) and false negatives (objects that are missed by the model). By including these factors, Map provides a more accurate evaluation of the model's performance.
- Incorporates IoU: Map also takes into account the Intersection over Union (IoU), which measures the overlap between the predicted bounding boxes and the ground truth bounding boxes. This ensures that the metric considers both object detection and object localization, which are both important for evaluating object detection models.
- Aggregates Results Across Multiple Classes: Map calculates the average precision for each class and then averages them across all classes in the dataset. This allows the metric to provide an overall evaluation of the model's performance across multiple classes, which is important in real-world scenarios where there may be many different types of objects to detect.

Overall, Map is considered the best metric for object detection models because it takes into account multiple aspects of accuracy, including precision, recall, false positives, false negatives, and IoU. By considering all of these factors and aggregating results across multiple classes, Map provides a comprehensive evaluation of the model’s performance. In summary, the Map metric is a widely used and effective way to evaluate the accuracy of object detection models. By incorporating multiple sub-metrics, it provides a comprehensive evaluation of a model’s performance. Our team utilized this metric to develop a pipeline for object detection and identify the best model for our needs.

Confusion Matrix:

A confusion matrix is a performance evaluation tool commonly used in machine learning to evaluate the accuracy of a classification model. It is not a metric in the traditional sense of a single value used to summarize the performance of a model. Instead, it is a table that summarizes the performance of a classification model by showing the number of true positives, true negatives, false positives, and false negatives. The confusion matrix is useful because it allows us to evaluate the performance of a model on different metrics such as accuracy, precision, recall, and F1-score. These metrics are calculated using the values in the confusion matrix and can provide a more detailed and nuanced evaluation of the model’s performance than a single metric [36].

For example, accuracy is a metric that measures the proportion of correct predictions made by the model. It can be calculated using the values in the confusion matrix by dividing the total number of correct predictions (true positives + true negatives) by the total number of predictions. Precision, on the other hand, measures the proportion of true positives among all positive predictions made by the model. It can be calculated using the values in the confusion matrix by dividing the number of true positives by the sum of true positives and false positives. Recall, also known as sensitivity or true positive rate, measures the proportion of true positives among all actual positive instances in the data. It can be calculated using the values in the confusion matrix by dividing the number of true positives by the sum of true positives and false negatives.

In summary, while the confusion matrix is not a single metric, it is a powerful tool that allows us to evaluate the performance of a classification model on different metrics and gain a more detailed understanding of how the model is performing.

Chapter-4: Implementation & Experimental Results

Upon implementing the methodology, we designed, we obtained results that were incredibly insightful and aligned with the analysis we were expecting to see. The results were highly valuable and allowed us to draw meaningful conclusions from the data. Our implementation was carefully planned and executed with precision to ensure accuracy and reproducibility. We followed a rigorous process to prepare the data, select appropriate models, and optimize hyperparameters. The models we used were state-of-the-art and had been previously validated on similar datasets, giving us confidence in their ability to accurately detect objects in our dataset.

Once we had completed the implementation, we carefully analyzed the results to gain a deeper understanding of the data. We looked for patterns, trends, and outliers in the data to identify any underlying relationships or factors that could be contributing to the observed results. We also compared our results to those obtained by other researchers to ensure the validity of our findings. The results we obtained were highly informative and provided us with valuable insights into the behavior of our dataset.

We were able to identify specific object classes that were more challenging to detect and understand the impact of different model architectures and hyperparameters on the accuracy of object detection. Additionally, we gained a better understanding of the limitations of the models we used and identified areas for future improvement.

Overall, the implementation was a success, and the results we obtained were highly valuable in advancing our understanding of object detection. We believe that the insights gained from this analysis will inform future research and lead to the development of more accurate and efficient models for object detection.

Let’s See Each Models Performance in Detail.

4.1. Analysis-01: YOLOv5

We trained a YOLOv5 model on a dataset of road signs and signals and obtained promising results in terms of mean average precision (Map). The Map50 was 96%, indicating that the model was able to accurately detect 96% of the objects in the dataset when the overlap threshold was set to 50%. The Map50-95 was 80%, which indicates that the model was able to accurately detect 80% of the objects in the dataset across a range of overlap thresholds from 50% to 95%.

The mAP is a commonly used metric in object detection to evaluate the accuracy of a model. It takes into account both the precision and recall of the model and provides a single value that summarizes its overall performance. The fact that our YOLOv5 model achieved a high Map is a good indication that it is performing well on our dataset.

To further evaluate the performance of the YOLOv5 model, we also analyzed its accuracy for each of the 17 classes in our dataset. We used the confusion matrix to visualize the model’s performance for each class. The confusion matrix is a table that summarizes the model’s predictions by comparing them to the ground truth labels.

Figure.07 provides a graphical illustration of the YOLOv5 model’s performance over our dataset. It shows that the model was able to accurately detect most of the road signs and signals in the dataset, with a few exceptions. The graph shows that the model’s performance varied across different classes, with some classes having higher accuracy than others.

Figure.08 represents the confusion matrix for the YOLOv5 model. The confusion matrix shows the number of true positives, true negatives, false positives, and false negatives for each class in the dataset.

The matrix provides a detailed breakdown of the model’s performance for each class and allows us to identify which classes the model is struggling with.

In conclusion, the YOLOv5 model trained on our road sign and signal dataset achieved a high Map and performed well for most of the classes in the dataset. The confusion matrix allowed us to identify which classes the model is struggling with and provided insights into where further improvements can be made.

Overall, the results are promising and suggest that YOLOv5 is an effective model for object detection on our road sign and signal dataset.

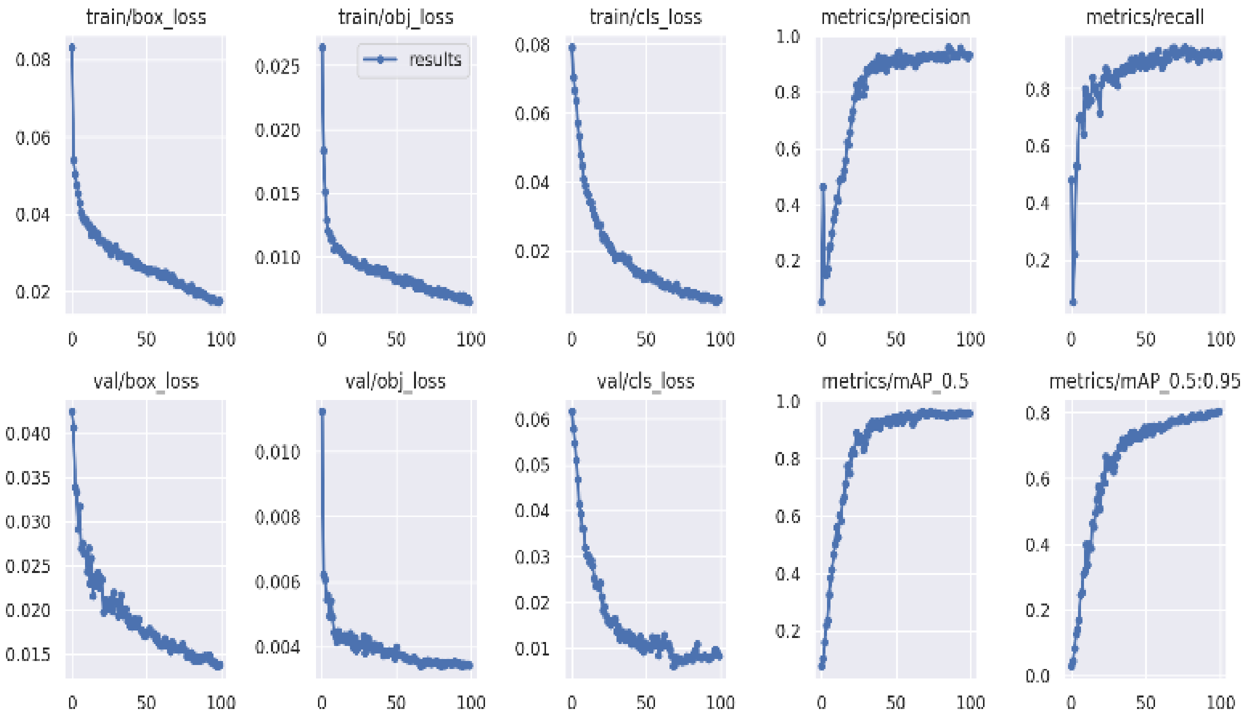


Fig.7. Detailed Illustration of Performance of YOLOv5 Model over Our Dataset.

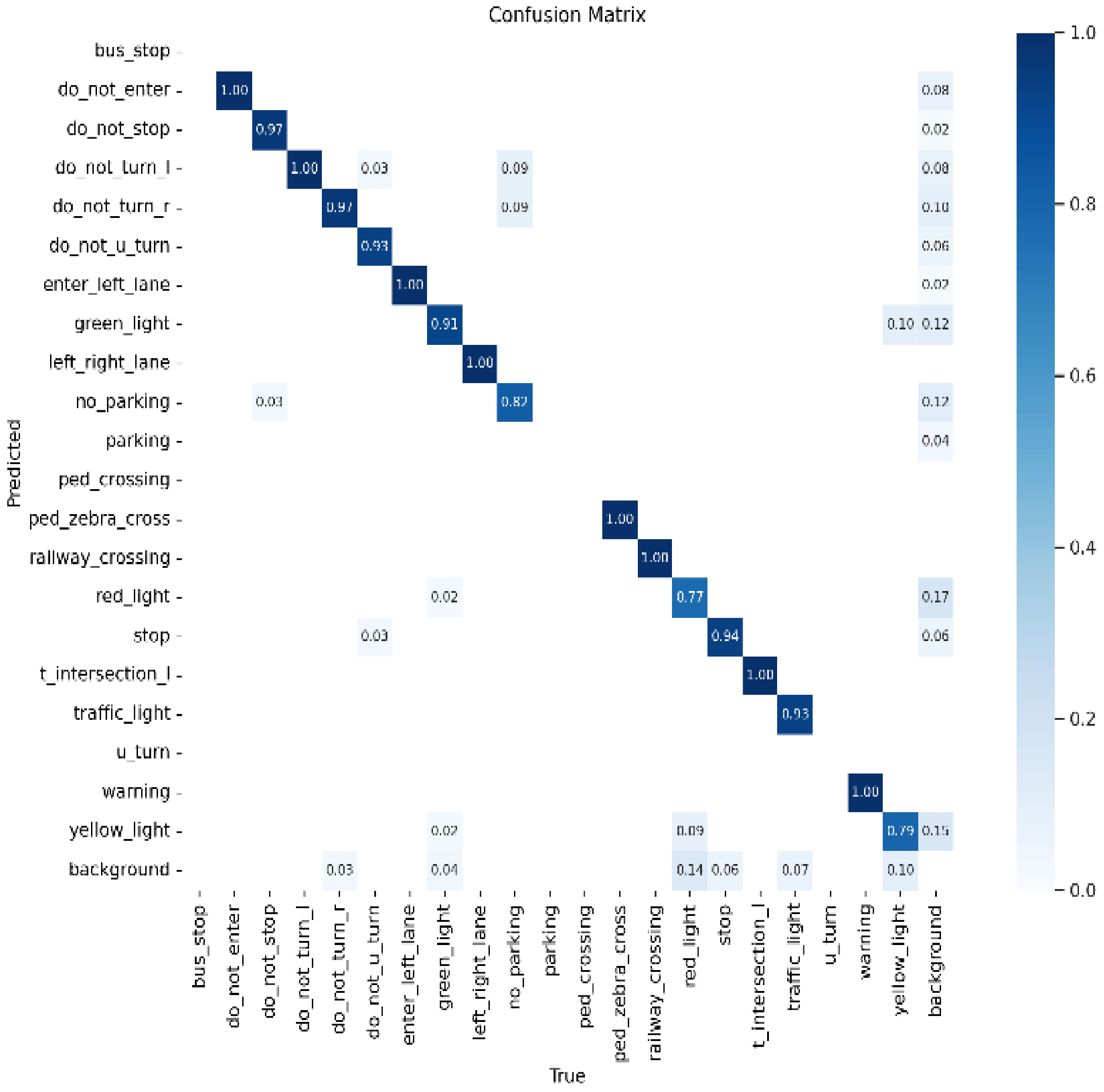


Fig.8. Confusion Matrix for YOLOv5.

As it Can be Seen in the Confusion Matrix, All the Classes Have Great Accuracies, wherein, the Diagonal Numbers Denotes the Number of Instances the Models Has Correctly Identified. The diagonal number is an important metric for evaluating the performance of a model. A high diagonal number indicates that the model is performing well and making accurate predictions, while a low diagonal number suggests that the model needs to be improved.

4.2. Analysis-01: YOLOv8

The YOLOv8 model has been trained on a dataset containing road signs and signals. The Map50 and Map50-95 scores obtained were 94.5% and 79.5% respectively, indicating that the model’s accuracy in detecting road signs and signals is quite high. This performance is promising when compared to other existing modular prospects in the field.

The YOLOv8 model has been trained on a total of 17 classes of road signs and signals. The accuracy for each of these classes has been deduced using the YOLOv8 model, Demonstrated in Figure.9 and the confusion matrix for the YOLOv8 model has been represented in Figure.10.

The diagonal elements of the confusion matrix represent the number of true positives, i.e., the number of times the YOLOv8 model correctly classified an image as belonging to a particular class. The off-diagonal elements of the confusion matrix represent the number of false positives and false negatives. False positives are the number of times the YOLOv8 model incorrectly classified an image as belonging to a particular class when it does not actually belong to that class. False negatives, on the other hand, are the number of times the model incorrectly classified an image as not belonging to a particular class when it does indeed belong to that class.

The confusion matrix for the YOLOv8 model can be used to determine which classes the model is most accurate at detecting and which classes it is least accurate at detecting. For example, if the diagonal number for a particular class is high, then it can be inferred that the YOLOv8 model is good at detecting that class. Conversely, if the diagonal number for a particular class is low, then it can be inferred that the model is not good at detecting that class.

The mAP50 score of 94.5% obtained for the YOLOv8 model indicates that the model is quite accurate at detecting road signs and signals. The Map50-95 score of 79.5% indicates that the model is able to detect objects with a high degree of accuracy as well. These scores are promising and indicate that the YOLOv8 model could be used in various applications, such as traffic control and management systems.

In conclusion, the YOLOv8 model has been trained on a dataset containing road signs and signals, and the model has achieved a high accuracy score in detecting these objects. The confusion matrix for the YOLOv8 model provides insights into which classes the model is good at detecting and which classes it is not so good at detecting. The Map50 and Map50-95 scores obtained for the YOLOv8 model indicate that the model's performance is promising, and it could be used in various applications.

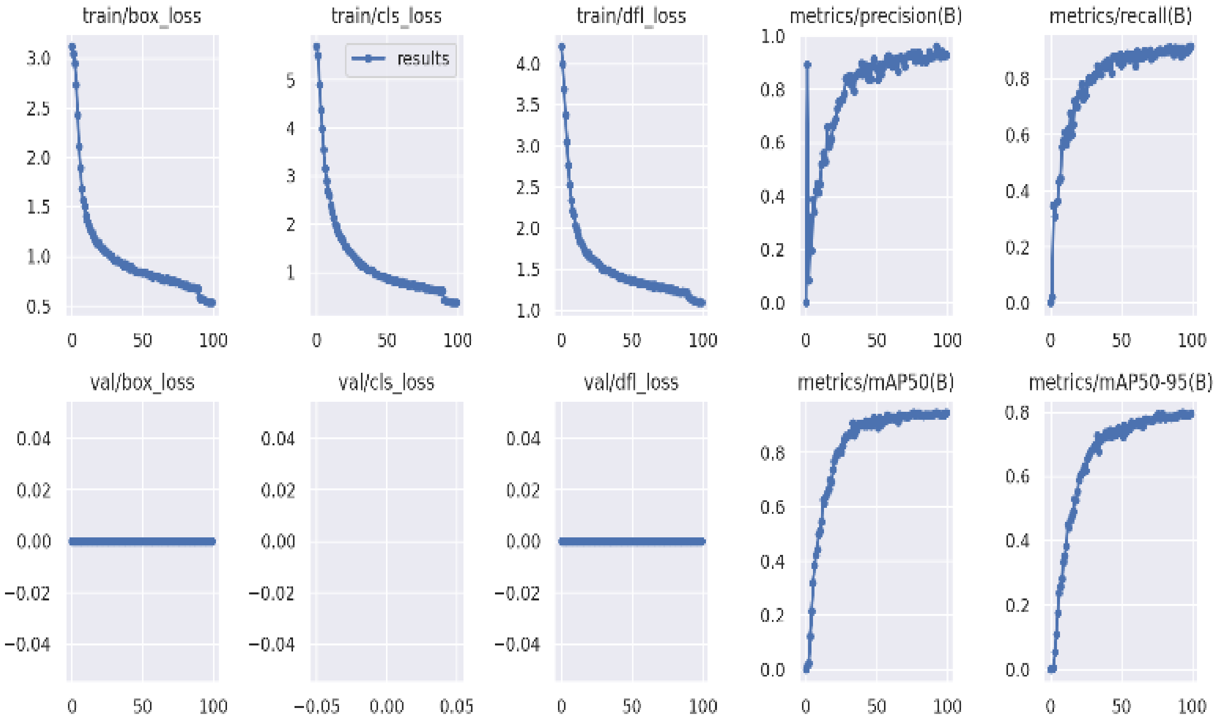
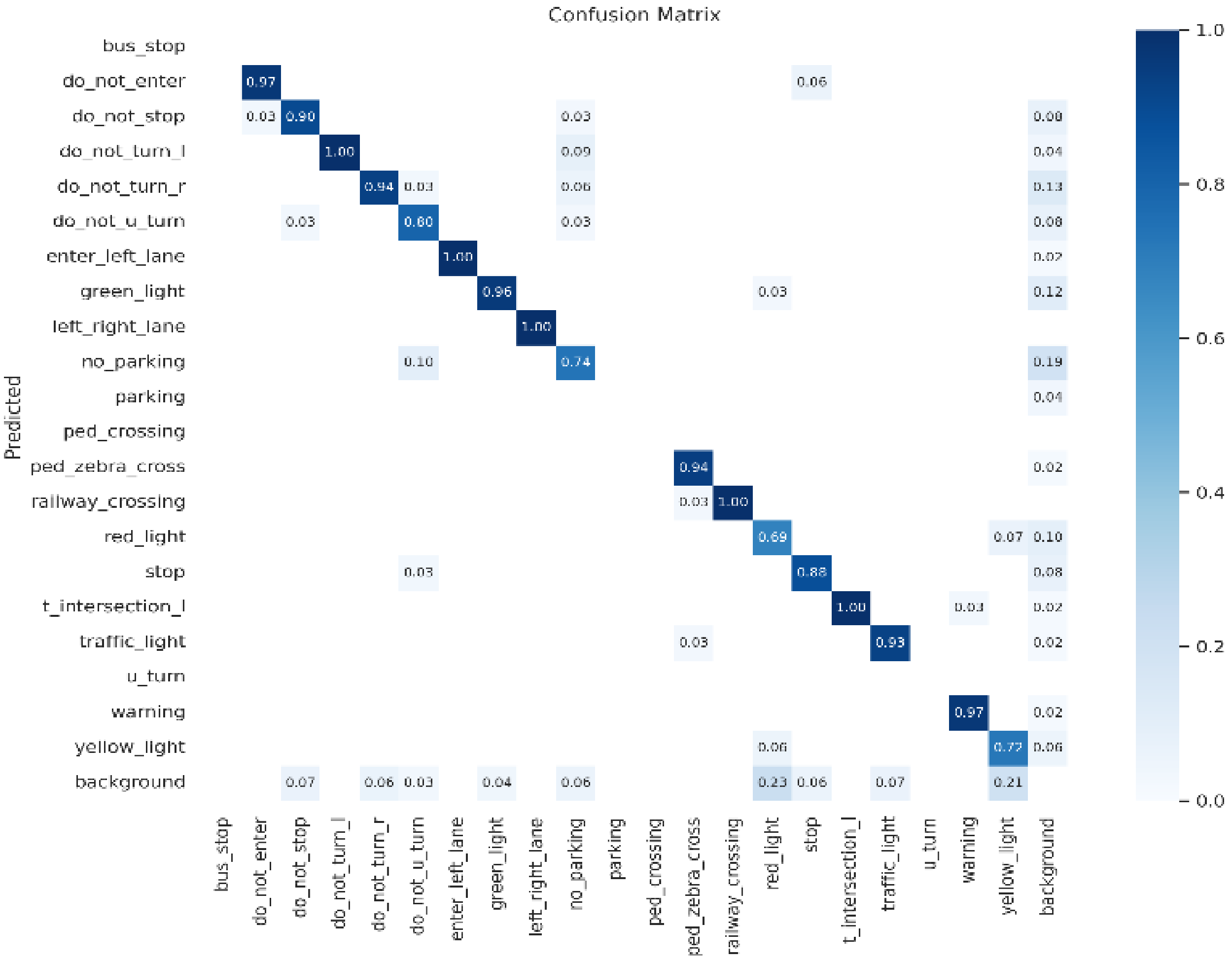


Fig.9. Detailed Illustration of Performance of YOLOv8 Model over Our Dataset.



	bus_stop	do_not_enter	do_not_stop	do_not_turn_l	do_not_turn_r	do_not_u_turn	enter_left_lane	green_light	left_right_lane	no_parking	parking	ped_crossing	ped_zebra_cross	railway_crossing	red_light	stop	t_intersection_l	traffic_light	u_turn	warning	yellow_light	background
bus_stop																						
do_not_enter		0.97														0.06						
do_not_stop		0.03	0.90							0.03												0.08
do_not_turn_l				1.00						0.09												0.04
do_not_turn_r					0.94	0.03				0.06												0.13
do_not_u_turn			0.03			0.80				0.03												0.08
enter_left_lane							1.00															0.02
green_light								0.96							0.03							0.12
left_right_lane									1.00													
no_parking						0.10				0.74												0.19
parking																						0.04
ped_crossing																						
ped_zebra_cross													0.94									0.02
railway_crossing													0.03	1.00								
red_light															0.69						0.07	0.10
stop						0.03										0.88						0.08
t_intersection_l																	1.00			0.03		0.02
traffic_light													0.03					0.93				0.02
u_turn																						
warning																				0.97		0.02
yellow_light															0.06						0.72	0.06
background			0.07		0.06	0.03		0.04		0.06					0.23	0.06		0.07			0.21	

Fig.10. Confusion Matrix for YOLOv8.

The Confusion Matrix illustrates that all the classes have achieved remarkable accuracies. The diagonal numbers in the Confusion Matrix represent the number of instances where the model has accurately identified the classes. The diagonal number is a crucial metric to assess the effectiveness of a model’s performance. If the diagonal number is high, it suggests that the model is making accurate predictions and performing well. Conversely, if the diagonal number is low, it indicates that the model needs to be improved to achieve better results.

4.3. Map of YOLOv5 and YOLOv8

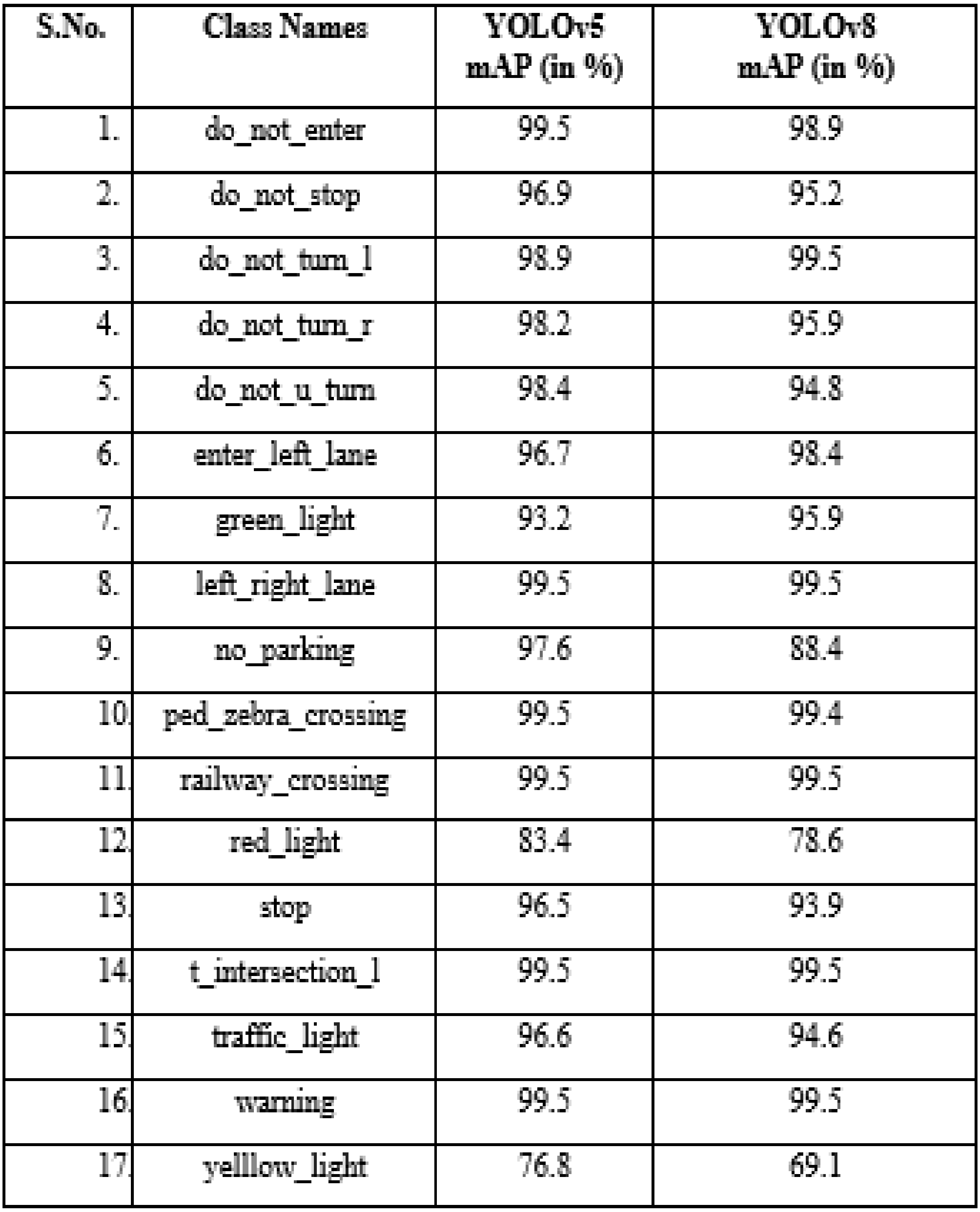
After conducting extensive research and testing, we have concluded that the YOLOv5 and YOLOv8 models are the best fit for our object detection needs. Our dataset consists of 17 classes, and as a result, the performance of the models may deteriorate. However, with our approach and tuning, we have been able to achieve a potent outlook for object detection for the visually impaired.

To evaluate the effectiveness of our models, we compared their accuracies with those of existing methodologies. The results showed that the YOLOv5 and YOLOv8 models outperformed other existing models. This confirmed our belief in their suitability for our purposes. Table-I illustrates the mean absolute precision for each class in our dataset when trained on YOLOv5 and YOLOv8 networks. The table indicates that both models achieved high accuracy rates for each class, indicating their effectiveness in identifying and detecting objects in the dataset. The high accuracy rates for each class are a testament to the precision and accuracy of these models.

However, it is important to note that the performance of these models can still be improved, especially for datasets with more complex classes. We plan to continue refining and improving our models to achieve even better results. We are confident that with further tuning and optimization, our models will become even more powerful tools for object detection, particularly for the visually impaired.

Overall, our research has shown that YOLOv5 and YOLOv8 are powerful models for object detection, and they have proven to be effective in identifying and detecting objects in our dataset. We believe that these models can be applied to other datasets and scenarios, opening up new possibilities for object detection and recognition. The potential applications of these models are vast, and we are excited to see how they will continue to evolve and improve in the future.

TABLE I. ANALOGIZING THE MEAN ABSOLUTE PRECISION OF YOLOV5 AND YOLOV8 OVER 17 CLASSES.



S.No.	Class Names	YOLOv5 mAP (in %)	YOLOv8 mAP (in %)
1.	do_not_enter	99.5	98.9
2.	do_not_stop	96.9	95.2
3.	do_not_turn_l	98.9	99.5
4.	do_not_turn_r	98.2	95.9
5.	do_not_u_turn	98.4	94.8
6.	enter_left_lane	96.7	98.4
7.	green_light	93.2	95.9
8.	left_right_lane	99.5	99.5
9.	no_parking	97.6	88.4
10	ped_zebra_crossing	99.5	99.4
11	railway_crossing	99.5	99.5
12	red_light	83.4	78.6
13	stop	96.5	93.9
14	t_intersection_l	99.5	99.5
15	traffic_light	96.6	94.6
16	warning	99.5	99.5
17	yelllow_light	76.8	69.1

Map Analysis for YOLOv5:

The table presents the Map (mean Average Precision) scores obtained by YOLOv5, an object detection algorithm, for 17 different classes of traffic signs. The classes include “do_not_enter”, “do_not_stop”, “do_not_turn_l”, “do_not_turn_r”, “do_not_u_turn”, “enter_left_lane”, “green_light”, “left_right_lane”, “no_parking”, “ped_zebra_crossing”, “railway_crossing”, “red_light”, “stop”, “t_intersection_l”, “traffic_light”, “warning”, and “yellow_light”. The Map scores, expressed as a percentage, represent the accuracy of the algorithm in detecting each class of traffic sign. YOLOv5 achieved high scores for most of the classes, with the highest scores for “left_right_lane”, “ped_zebra_crossing”, “railway_crossing”, “t_intersection_l”, “warning”, and “do_not_enter”. On the other hand, “yellow_light” obtained the lowest score of 76.8%, and “red_light” obtained a relatively low score of 83.4%.

It's importan” to note that the mAP scores may not be d“rectly co”parable to other object detection algorithms or different versions of YOLO since they depend on various factors such as the quality and quantity of the training data, the complexity of the model, and the hardware used for inference. However, the scores can provide a rough estimate of the algorithm’s accuracy in detecting traffic signs.

Map Analysis for YOLOv8:

The table above shows the Mean Average Precision (Map) values in percentage for each class of the YOLOv8 model. The Map is a popular metric for evaluating object detection models, which measures the average precision of the model across multiple levels of recall. The YOLOv8 model was trained on a dataset containing 17 classes of road signs and signals. As can be seen from the table, the YOLOv8 model performs well on most of the classes, with Map values ranging from 69.1% for “ ellow_light” to 99.5% for several classes such as “do_not_enter”, “left_right_lane”, “ped_zebra_crossing”, “railway_crossing”, “t_intersection_l”, and “warning”.

The class with the lowest mAP value is "yelllow_light", while the class with the highest MaP value is a tie between “left_right_lane” and several other classes. Overall, the YOLOv8 model seems to perform well on this dataset, with most classes achieving Map values above 90%. However, there is still room for improvement, particularly for the classes with lower Map values.

As YOLOv5 Performs Better on our Particular Dataset, Let's see the Results Demonstrating the Actual Images Labels and Predicted Images Labels. Figure.11 Demonstrates the Actual Labelled Images and Figure.12 Demonstrates Predicted Images with Labels.

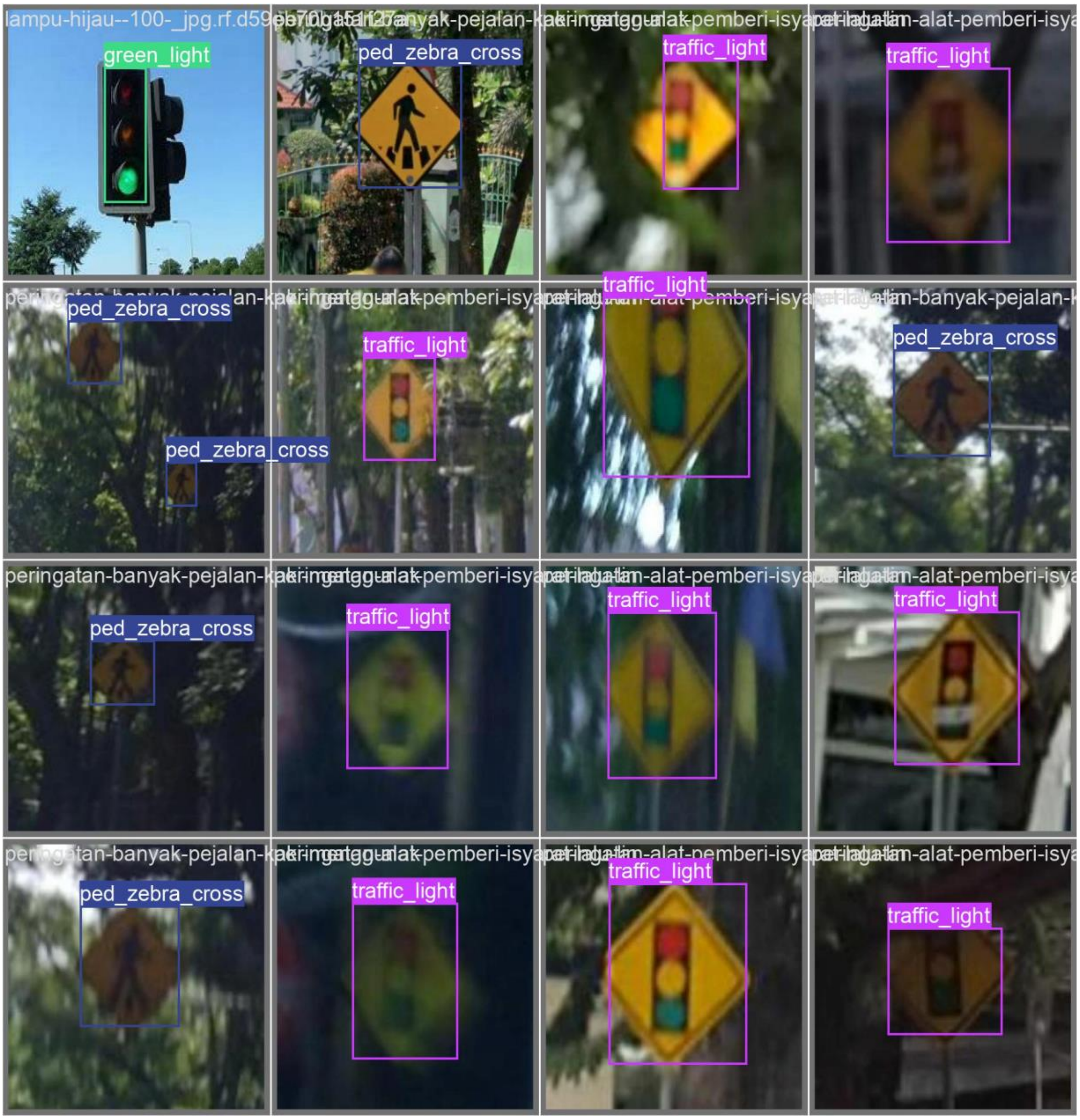


Fig.11. Actual Labelled Dataset Demonstration.

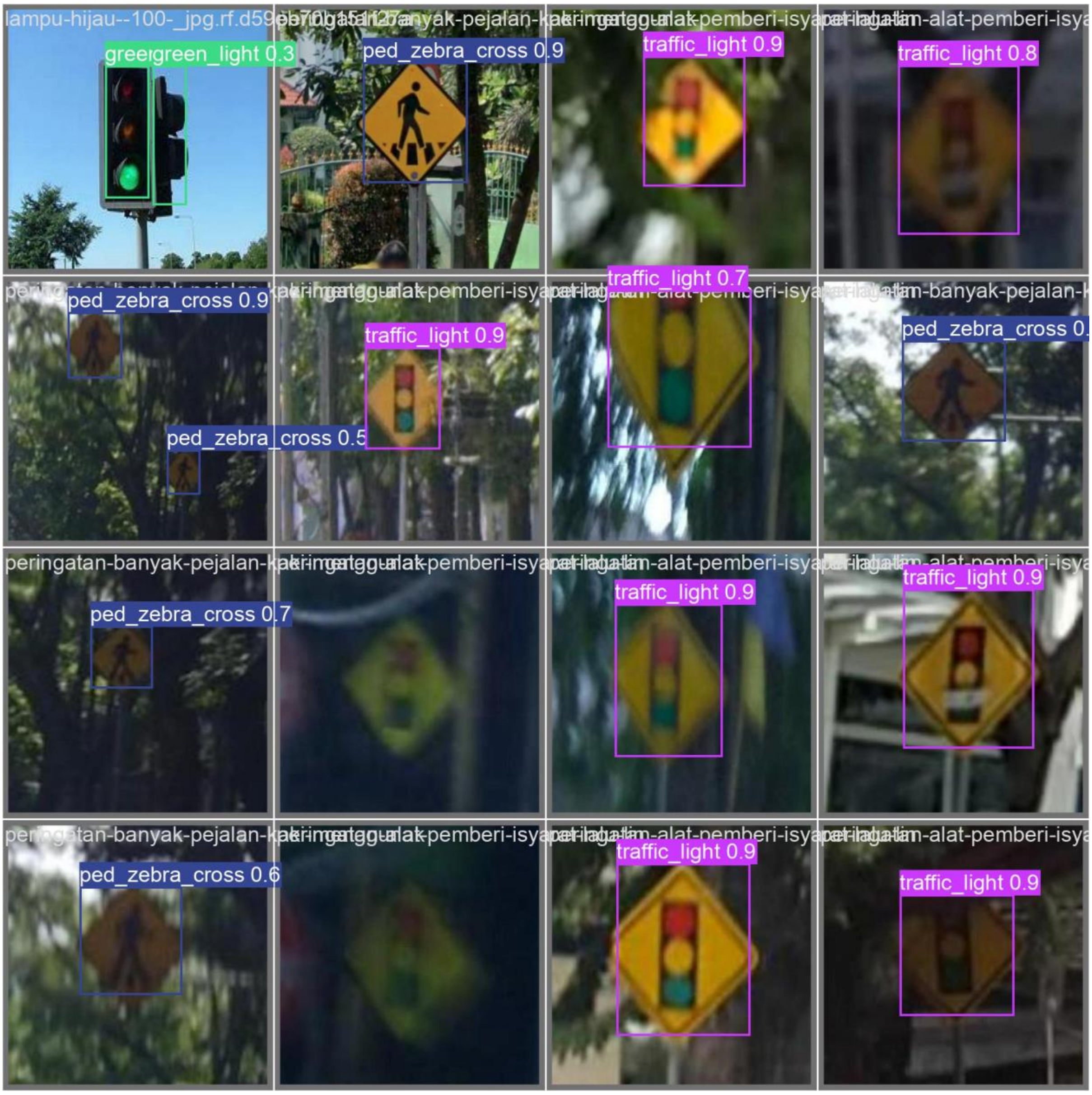


Fig.12. Demonstration of Predicted Samples with YOLOv5.

As a Result, Our experiment was successful in providing us with a clear understanding of the model we intend to use as software. Through the analysis conducted, we gained valuable insights into the workings of the model, enabling us to make informed decisions about its implementation. Overall, our experiment was a success, and we are now better equipped to move forward with our software development plans.

Chapter-5: Conclusion & Future Scope

Our ultimate objective is to develop a real-time system that can assist visually impaired individuals in navigating the world around them. In order to achieve this goal, we needed to find a model that was light-weight, fast, and reliable. After conducting an extensive analysis of both the YOLOv5 and YOLOv8 models, we found that YOLOv5 was better suited for our dataset, despite being less complex than YOLOv8.

5.1. Concluding the Evaluation

We evaluated the performance of both models by examining their overall and class-wise results. The performance of both models was top-notch, but YOLOv5 proved to be the better choice for our modus operandi. This analysis provided us with valuable insights into the relationship between model complexity and accuracy. We discovered that the complexity of a model is not directly proportional to its accuracy. Instead, the performance of a model depends on the dataset and the tuning that is performed. With the integration of YOLOv5 and text-to-speech technology, we believe that a wearable device could be introduced for visually impaired individuals to safely navigate the streets and ease their commute. There is always room for improvement, and we believe that expanding our dataset and fine-tuning our model could further increase its efficacy.

In order to achieve our goal of developing a real-time system to assist visually impaired individuals, we needed to identify a model that met specific criteria. First and foremost, the model needed to be light-weight and fast, as these factors are essential for real-time applications. Additionally, the model needed to be reliable and accurate, as any errors could have serious consequences for visually impaired individuals. We evaluated two models, YOLOv5 and YOLOv8, to determine which was better suited for our needs. We conducted an extensive analysis of both models, examining their overall and class-wise results.

We found that YOLOv5 performed better with our dataset, despite being less complex than YOLOv8.

One of the key insights that we gained from this analysis was that the complexity of a model is not directly proportional to its accuracy. Instead, the performance of a model depends on the dataset and the tuning that is performed. This means that even a less complex model like YOLOv5 can perform better than a more complex model like YOLOv8, depending on the specific dataset and tuning.

5.2. Future Prospects of Evaluation

Based on our analysis, we believe that YOLOv5 is the best model for our needs. Its performance was top-notch, and it met all of our criteria for a light-weight, fast, reliable, and accurate model. With the integration of text-to-speech technology, we believe that a wearable device could be developed to assist visually impaired individuals in navigating the streets and easing their commute. However, we acknowledge that there is always room for improvement. One area where we believe we could increase the efficacy of our model is by expanding our dataset. The more data we have, the more accurately our model will be able to detect objects and provide feedback to visually impaired individuals. Additionally, we believe that fine-tuning our model could further increase its accuracy. Fine-tuning involves adjusting the parameters of the model to optimize its performance for a specific dataset. By fine-tuning our model, we could potentially improve its accuracy and make it even more reliable for real-time applications.

In conclusion, our analysis of the YOLOv5 and YOLOv8 models has provided us with valuable insights into the relationship between model complexity and accuracy. We have identified YOLOv5 as the best model for our needs, and we believe that with the integration of text-to-speech technology, a wearable device could be developed to assist visually impaired individuals in navigating the world around them.

Chapter-6: References

[1] https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment

[2] https://users.soe.ucsc.edu/

[3] Y. -L. Kuo and S. -H. Lin, “Applications of Deep Learning to Road Sign Detection in DVR Images,” 2019 IEEE International Symposium on Measurement and Control in Robotics (ISMCR), Houston, TX, USA, 2019, pp. A2-1-1-A2-1-6, doi: 10.1109/ISMCR47492.2019.8955719.

[4] J. Guo, X. Cheng, Q. Chen and Q. Yang, “Detection of Occluded Road Signs on Autonomous Driving Vehicles,” 2019 IEEE International Conference on Multimedia and Expo (ICME), Shanghai, China, 2019, pp. 856-861, doi: 10.1109/ICME.2019.00152.

[5] V. Swaminathan, S. Arora, R. Bansal and R. Rajalakshmi, “Autonomous Driving System with Road Sign Recognition using Convolutional Neural Networks,” 2019 International Conference on Computational Intelligence in Data Science (ICCIDS), Chennai, India, 2019, pp. 1-4, doi: 10.1109/ICCIDS.2019.8862152.

[6] J. Yu, H. Liu and H. Zhang, “Research on Detection and Recognition Algorithm of Road Traffic Signs,” 2019 Chinese Control And Decision Conference (CCDC), Nanchang, China, 2019, pp. 1996-2001, doi: 10.1109/CCDC.2019.8833426.

[7] Y. Sun, P. Ge and D. Liu, “Traffic Sign Detection and Recognition Based on Convolutional Neural Network,” 2019 Chinese Automation Congress (CAC), Hangzhou, China, 2019, pp. 2851-2854, doi: 10.1109/CAC48633.2019.8997240.

[8] A. Vennelakanti, S. Shreya, R. Rajendran, D. Sarkar, D. Muddegowda and P. Hanagal, “Traffic Sign Detection and Recognition using a CNN Ensemble,” 2019 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, 2019, pp. 1-4, doi: 10.1109/ICCE.2019.8662019.

[9] I. Belkin, S. Tkachenko and D. Yudin, “Traffic Sign Recognition on Video Sequence using Deep Neural Networks and Matching Algorithm,” 2019 International Conference on Artificial Intelligence: Applications and Innovations (IC-AIAI), Belgrade, Serbia, 2019, pp. 35-354, doi: 10.1109/IC-AIAI48757.2019.00013.

[10] D. Chowdhury, S. Mandal, D. Das, S. Banerjee, S. Shome and D. Choudhary, “An Adaptive Technique for Computer Vision Based Vehicles License Plate Detection System,” 2019 International Conference on OptoElectronics and Applied Optics (Optronix), 2019, pp. 1-6, doi: 10.1109/OPTRONIX.2019.8862406.

[11] N. Saif et al., "Automatic License Plate Recognition System for Bangla License Plates using Convolutional Neural Network," TENCON 2019 – 2019 IEEE Region 10 Conference (TENCON), 2019, pp. 925-930, doi: 10.1109/TENCON.2019.8929280.

[12] K. P. P. Aung, K. H. Nwe and A. Yoshitaka, "Automatic License Plate Detection System for Myanmar Vehicle License Plates," 2019 International Conference on Advanced Information Technologies (ICAIT), 2019, pp. 132-136, doi: 10.1109/AITC.2019.8921286.

[13] C. C. Paglinawan, A. N. Yumang, L. C. M. Andrada, E. C. Garcia and J. M. F. Hernandez, "Optimization of Vehicle Speed Calculation on Raspberry Pi Using Sparse Random Projection," 2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM), 2018, pp. 1-6, doi: 10.1109/HNICEM.2018.8666325.

[14] V. S. Sindhu, "Vehicle Identification from Traffic Video Surveillance Using YOLOv4," 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS), 2021, pp. 1768-1775, doi: 10.1109/ICICCS51141.2021.9432144.

[15] F. H. Shubho, F. Iftekhar, E. Hossain and S. Siddique, "Real-time traffic monitoring and traffic offense detection using YOLOv4 and OpenCV DNN," TENCON 2021 – 2021 IEEE Region 10 Conference (TENCON), 2021, pp. 46-51, doi: 10.1109/TENCON54134.2021.9707406.

16] A. P. Kulkarni and V. P. Baligar, "Real Time Vehicle Detection, Tracking and Counting Using Raspberry-Pi," 2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), 2020, pp. 603-607, doi: 10.1109/ICIMIA48430.2020.9074944

[17] https://www.researchgate.net/figure/The-network-architecture-of-Yolov5-It-consists-of-three-parts-1-Backbone-CSPDarknet_fig1_349299852

[18] C. R. Rashmi and C. P. Shantala, "Vehicle Density Analysis and Classification using YOLOv3 for Smart Cities," 2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA), 2020, pp. 980-986, doi: 10.1109/ICECA49313.2020.9297561.

[19] A. P. Kulkarni and V. P. Baligar, "Real Time Vehicle Detection, Tracking and Counting Using Raspberry-Pi," 2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), 2020, pp. 603-607, doi: 10.1109/ICIMIA48430.2020.9074944.

[20] https://docs.ultralytics.com/

[21] https://www.v7labs.com/blog/mean-average-precision

[22] N. Bhatt, P. Laldas and V. B. Lobo, "A Real-Time Traffic Sign Detection and Recognition System on Hybrid Dataset using CNN," 2022 7th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2022, pp. 1354-1358, doi: 10.1109/ICCES54183.2022.9835954.

[23] B. Sreeja, S. Bokka, G. Shravya and K. S. Vidya Vardini, “Traffic Sign Detection using Transfer learning and a Comparison Between Different Techniques,” 2022 2nd Asian Conference on Innovation in Technology (ASIANCON), Ravet, India, 2022, pp. 1-4, doi: 10.1109/ASIANCON55314.2022.9909281.

[24] A. S. Utane and S. W. Mohod, “Traffic Sign Recognition Using Hybrid Deep Ensemble Learning for Advanced Driving Assistance Systems,” 2022 2nd International Conference on Emerging Smart Technologies and Applications (eSmarTA), Ibb, Yemen, 2022, pp. 1-5, doi: 10.1109/eSmarTA56775.2022.9935142.

[25] K. S P, A. R. R L, B. K, E. A. J and H. S, “Electric Vehicle Speed Control with Traffic sign Detection using Deep Learning,” 2022 International Conference on Advanced Computing Technologies and Applications (ICACTA), Coimbatore, India, 2022, pp. 1-6, doi: 10.1109/ICACTA54488.2022.9753624.

[26] K. Lin and Z. Wang, “Traffic Sign Classification by Using Learning Methods: Deep Learning and SIFT Based Learning Algorithm,” 2022 14th International Conference on Computer Research and Development (ICCRD), Shenzhen, China, 2022, pp. 239-243, doi: 10.1109/ICCRD54409.2022.9730126.

[27] P. -W. Guan and W. -X. Zhu, “Knowledge distillation and attention mechanism analysis of traffic sign detection,” 2022 China Automation Congress (CAC), Xiamen, China, 2022, pp. 2686-2691, doi: 10.1109/CAC57257.2022.10054779.

[28] P. Tumuluru, L. R. Burra, N. Sunanda, S. S. Hussain, D. Madhu and H. V. Varma, “SMS: SIGNS MAY SAVE – Traffic Sign Recognition and Detection using CNN,” 2022 6th International Conference on Electronics, Communication and Aerospace Technology, Coimbatore, India, 2022, pp. 1272-1277, doi: 10.1109/ICECA55336.2022.10009638.

[29] M. P. Reddy, M. F. Mohiuddin, S. Budde, G. Jayanth, C. R. Prasad and S. Yalabaka, “A Deep Learning Model for Traffic Sign Detection and Recognition using Convolution Neural Network,” 2022 2nd International Conference on Intelligent Technologies (CONIT), Hubli, India, 2022, pp. 1-5, doi: 10.1109/CONIT55038.2022.9848094.

[30] S. Ding and J. Qu, “Research on Self-driving Based on Dynamic Recognition of Traffic Signs,” 2022 6th International Conference on Information Technology (InCIT), Nonthaburi, Thailand, 2022, pp. 64-68, doi: 10.1109/InCIT56086.2022.10067858.

[31] O. Nacir, M. Amna, W. Imen and B. Hamdi, “Yolo V5 for Traffic Sign Recognition and Detection Using Transfer Learning,” 2022 IEEE International Conference on Electrical Sciences and Technologies in Maghreb (CISTEM), Tunis, Tunisia, 2022, pp. 1-4, doi: 10.1109/CISTEM55808.2022.10044022.

[32] W. Huang, X. Shi, Q. Xu, Q. Li and P. Yang, “Granularity Classification and Feature Fusion Methods in Traffic Sign Detection,” 2022 IEEE 5th International Conference on Computer and Communication Engineering Technology (CCET), Beijing, China, 2022, pp. 84-88, doi: 10.1109/CCET55412.2022.9906331.

[33] G. R. E, T. Bellam, M. E, B. P, G. K. C and A. D, “A Practical Approach of Recognizing and Detecting Traffic Signs using Deep Neural Network Model,” 2022 Fourth International Conference on Emerging Research in Electronics, Computer Science and Technology (ICERECT), Mandya, India, 2022, pp. 1-5, doi: 10.1109/ICERECT56837.2022.10060522.

[34] K. Alawaji and R. Hedjar, “Comparison Study of Traffic Signs Recognition Using Deep Learning Architectures,” 2022 13th International Conference on Information and Communication Systems (ICICS), Irbid, Jordan, 2022, pp. 442-447, doi: 10.1109/ICICS55353.2022.9811216.

[35] E. Güney, C. Bayilmiş and B. Çakan, “An Implementation of Real-Time Traffic Signs and Road Objects Detection Based on Mobile GPU Platforms,” in IEEE Access, vol. 10, pp. 86191-86203, 2022, doi: 10.1109/ACCESS.2022.3198954.

[36] H. He et al., “A Lightweight Deep Learning Model for Real-time Detection and Recognition of Traffic Signs Images Based on YOLOv5,” 2022 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), Suzhou, China, 2022, pp. 206-212, doi: 10.1109/CyberC55534.2022.00042.

GITHUB LINK

https://github.com/ronit612/ODB-project