Traffic sign recognition using deep learning

Lakshya Mehta

March 2024

Declaration I, Lakshya Mehta, confirm that the dissertation titled ”Traffic Sign Recognition Using Deep Learning” is my original work. It has not been submitted for any academic qualification elsewhere. The information presented in this dissertation adheres to academic standards and ethical guidelines. Proper referencing has been employed for all information obtained from external sources.

Acknowledgement This paper is dedicated to my parents, Pratap Singh Mehta and Laxmi Mehta, for their unwavering support and encouragement. Additionally, I dedicate this work to my supervisor for their invaluable guidance and assistance throughout the project.

Contents

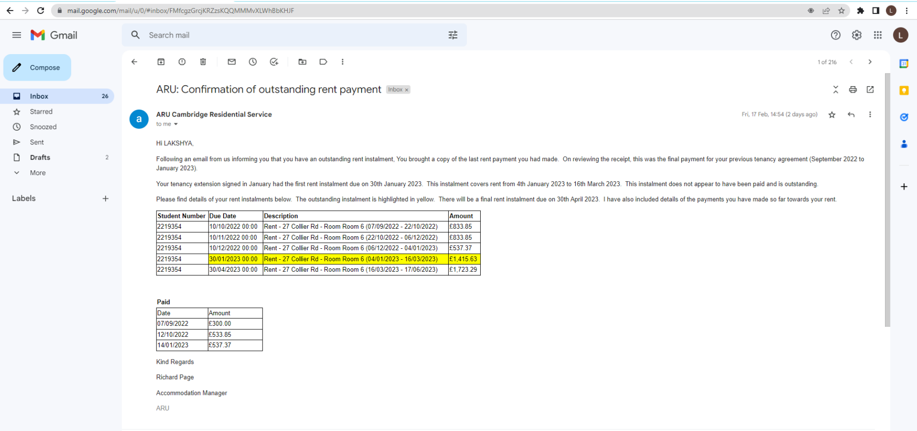
List of Figures

1 Abstract

Recognizing traffic signs has historically been challenging, but recent technological advancements have addressed these obstacles through various techniques and methodologies. Traffic sign recognition systems are vital for applications like self-driving cars and driver assistance systems, allowing vehicles to interpret and respond to road signs effectively. Image data for traffic signs often presents challenges such as variations in lighting conditions, occlusions, and the presence of other objects like vehicles and pedestrians. To overcome these challenges, techniques like data augmentation (e.g., random horizontal flipping) and post-processing methods (e.g., Non-Maximum Suppression) are employed. These methods enhance the diversity and accuracy of training data and refine detection results. Sophisticated object detection models like Faster R-CNN with ResNet-50 FPN backbone networks are utilized for detecting traffic sign objects. These models leverage deep learning algorithms to accurately detect and classify objects within images. The implemented methodology has shown promising results, achieving a mean Average Precision (mAP) of 46.7% and a recall rate of 46.5%, indicating the model's effectiveness in identifying and localizing traffic sign objects for improved road safety and vehicle functionality.

2 Ethics approval

The project was submitted to Mr Stiphen Chowdhury for approval on March 11, 2024, and received approval on March 11, 2024. It was classified as low risk with a status of Green.



ARU: Confirmation of outstanding rent payment

ARU Cambridge Residential Service
to me

Fri, 17 Feb, 14:54 (2 days ago)

Hi LAKSHYA,

Following an email from us informing you that you have an outstanding rent instalment, You brought a copy of the last rent payment you had made. On reviewing the receipt, this was the final payment for your previous tenancy agreement (September 2022 to January 2023).

Your tenancy extension signed in January had the first rent instalment due on 30th January 2023. This instalment covers rent from 4th January 2023 to 16th March 2023. This instalment does not appear to have been paid and is outstanding.

Please find details of your rent instalments below. The outstanding instalment is highlighted in yellow. There will be a final rent instalment due on 30th April 2023. I have also included details of the payments you have made so far towards your rent.

Student Number	Due Date	Description	Amount
2219354	10/10/2022 00:00	Rent - 27 Collier Rd - Room Room 6 (07/09/2022 - 22/10/2022)	£833.85
2219354	10/11/2022 00:00	Rent - 27 Collier Rd - Room Room 6 (22/10/2022 - 06/12/2022)	£833.85
2219354	10/12/2022 00:00	Rent - 27 Collier Rd - Room Room 6 (06/12/2022 - 04/01/2023)	£537.37
2219354	30/01/2023 00:00	Rent - 27 Collier Rd - Room Room 6 (04/01/2023 - 16/03/2023)	£1,415.63
2219354	30/04/2023 00:00	Rent - 27 Collier Rd - Room Room 6 (16/03/2023 - 17/06/2023)	£1,723.29

Paid

Date	Amount
07/09/2022	£300.00
12/10/2022	£533.85
14/01/2023	£537.37

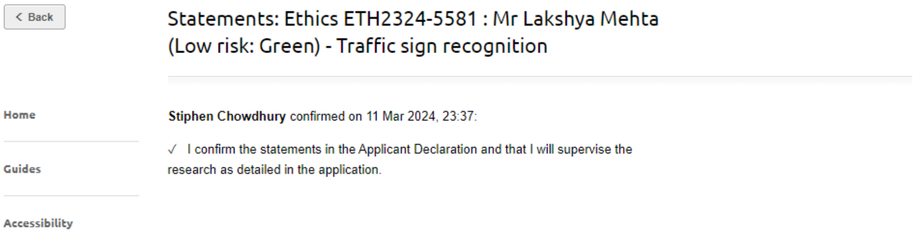
Kind Regards

Richard Page

Accommodation Manager

ARU

Figure 1: Ethics approval first image



Statements: Ethics ETH2324-5581 : Mr Lakshya Mehta (Low risk: Green) - Traffic sign recognition

Stiphen Chowdhury confirmed on 11 Mar 2024, 23:37:

✓ I confirm the statements in the Applicant Declaration and that I will supervise the research as detailed in the application.

Figure 2: Ethics approval second image

3 Introduction

3.1 Overview

Recognizing traffic signs poses a significant challenge, particularly in the context of emerging technologies. Fortunately, advancements in computer vision, specifically leveraging deep learning libraries, have effectively addressed this challenge. The application of deep learning technology has proven instrumental in the context of cameras in self-driving cars and driver-assistance systems, facilitating the recognition of various traffic signs. Object detection, a fundamental task in computer vision, involves identifying and localizing objects within images. In the context of traffic sign detection, this process is crucial for various applications such as autonomous vehicles, traffic surveillance systems, and self-driving cars. In traffic sign object detection, the primary objective is to accurately locate and classify traffic signs present in a scene captured by an image or video frame. This entails detecting the spatial coordinates of the signs within the frame and correctly identifying their semantic meaning. The significance of traffic sign detection lies in its role in ensuring safe and efficient navigation of vehicles on roads. Once a traffic sign is detected and classified, the vehicle's onboard systems can interpret the sign's meaning and take appropriate actions, such as adjusting speed, changing lanes, or coming to a stop, in accordance with traffic regulations and safety requirements.

Overall, traffic sign object detection plays a vital role in enhancing the perception capabilities of intelligent systems deployed in vehicles, contributing to improved road safety and efficiency in transportation systems. The complexity of traffic sign recognition arises from the vast array of signs, turning it into a multi-class classification problem. Furthermore, different countries exhibit variations in their traffic sign designs. Open-source datasets play a crucial role in training models for this task, with notable examples being Tsinghua Tencent 100k (TT100K), German Traffic Sign Recognition Benchmark (GTSRB), and German Traffic Sign Detection Benchmark (GTSDB). These datasets encompass diverse scenarios, including variations in lighting conditions, blur, angles, dimensions, and image quality. For this paper, the GTSDB dataset is specifically utilized, comprising 900 images. Of these, 600 images are employed for training the models, while the remaining 300 are reserved for testing the model's performance and accuracy. The traffic signs within the dataset are categorized into four distinct sub-categories: warning, informative, obligatory, and prohibited. An object detection model is applied to detect and classify these traffic signs, contributing to the design of a system aimed at assisting self-driving cars and drivers through the integration of deep learning-based computer vision technology.

3.2 Problem Background

Large datasets present both computational and processing challenges, as training models with extensive data requires substantial time and computational resources. Moreover, the high-quality images within these datasets demand more processing time for accurate detection and classification of traffic signs. The diversity in image characteristics, including varying angles, dimensions, blurring, and lighting conditions, further complicates the task of traffic sign recognition. Additionally, the presence of numerous other objects in the images, such as pedestrians, trees, railings, and vehicles, adds to the complexity of accurately classifying the traffic sign objects amidst the clutter.

Stallkamp et al. (2011) highlight these challenges, emphasizing the difficulty in correctly classifying traffic signs amidst the diverse image backgrounds and conditions. Furthermore, Houben et al. (2013a) discuss the intricacies of evaluating benchmark performance, particularly in comparison to assessing the classification stage of algorithms. This underscores the need for robust evaluation methodologies to gauge the effectiveness of traffic sign recognition systems accurately.

3.3 Research Aim

This paper aims to achieve accurate detection and classification of traffic signs within the images of the GTSDB dataset utilizing a computer vision object detection model, specifically the Faster R-CNN model. Additionally, the performance of both the proposed models will undergo thorough evaluation. To accomplish this objective, a combination of machine learning and deep learning libraries will be employed, along with their respective modules and functions, to fine-tune the hyperparameters of the models. Furthermore, these libraries will facilitate the visualization of both the data and results, offering insights into the effectiveness and efficiency of the proposed approach. Traffic signs play a crucial role in reducing accidents and ensuring road safety for both pedestrians and drivers. These signs convey important information about road conditions, regulations, hazards, and directions, helping road users make informed decisions while navigating through traffic. The timely and accurate identification of traffic signs is essential for effective decision-making on the road. Pedestrians rely on traffic signs to safely cross streets and navigate through urban areas, while drivers need to quickly interpret signs to comply with traffic laws and adjust their driving behavior accordingly. Moreover, the visibility and recognition of traffic signs are vital not only during the day but also at night and in adverse weather conditions. Adequate illumination, reflective materials, and clear markings are essential to ensure that signs remain visible and legible under various lighting conditions, including low light and darkness.By promptly identifying and understanding traffic signs, pedestrians and drivers can anticipate potential hazards, adhere to traffic regulations, and take appropriate actions to prevent accidents and ensure the smooth flow of traffic. Therefore, accurate and efficient traffic sign detection systems are indispensable for enhancing road safety and minimizing the risk of accidents on our roads.

3.4 Research Objectives

The objectives of this paper are as follows:

- To implement the Faster R-CNN object detection model.
- To successfully detect and classify traffic signs in the GTSDB dataset using the above-mentioned object detection models.

This paper aims to achieve two primary objectives related to traffic sign detection using the Faster R-CNN object detection model. The objectives are outlined as follows:

1. Implementation of the Faster R-CNN Object Detection Model: The first objective of this paper is to implement the Faster R-CNN (Region-based Convolutional Neural Networks) object detection model. Faster R-CNN is a state-of-the-art deep learning model used for object detection tasks, known for its accuracy and efficiency. This objective involves understanding the architecture and functioning of the Faster R-CNN model and implementing it using appropriate frameworks such as PyTorch or TensorFlow. The implementation process includes configuring the model architecture, training the model on annotated datasets, and fine-tuning the model parameters to optimize its performance.

2. Detection and Classification of Traffic Signs: The second objective is to utilize the implemented Faster R-CNN model to detect and classify traffic signs within the German Traffic Sign Detection Benchmark (GTSDB) dataset. The GTSDB dataset contains a collection of images with annotated traffic signs, making it suitable for training and evaluating object detection models. To achieve this objective, the implemented Faster R-CNN model will be trained on the GTSDB dataset to learn the features and characteristics of various traffic signs. During the training process, the model will be optimized to accurately detect and classify different types of traffic signs, including regulatory, warning, and informational signs.

Upon successful implementation and training of the Faster R-CNN model, the paper aims to evaluate its performance in terms of detection accuracy, classification accuracy, and overall efficiency. Various evaluation metrics such as Average Precision (AP), Average Recall (AR), and Intersection over Union (IoU) will be utilized to assess the model's performance objectively. Additionally, the paper will analyze the model's ability to generalize to unseen data and its robustness to variations in lighting conditions, occlusions, and other environmental factors commonly encountered in real-world scenarios.

In conclusion, this paper endeavors to contribute to the field of traffic sign detection by implementing and evaluating the Faster R-CNN object detection model for accurate and efficient detection and classification of traffic signs. By achieving the stated objectives, the paper aims to provide insights into the effectiveness of deep learning-based approaches for addressing real-world challenges in traffic sign recognition and enhancing road safety systems.

3.5 Research Scope

The evaluation of the model will involve assessing various parameters, primarily utilizing the COCO evaluation metrics. These metrics, including Average Precision (AP), True Positive (TP), False Positive (FP), and Average Recall (AR), provide valuable insights into the model's performance. To facilitate this evaluation process, we will leverage the pycocotools library, a powerful toolkit for importing and implementing these metrics within our codebase.

By integrating these evaluation metrics into our workflow, we can quantitatively measure the model's ability to detect and classify objects accurately. The Average Precision metric indicates the precision of object detection, while True Positive and False Positive metrics provide insights into the model's ability to correctly identify and distinguish between true and false detections. Additionally, the Average Recall metric helps assess the model's ability to recall relevant objects across the dataset.

Both of our object detection models will undergo rigorous testing using these metrics to assess their performance accurately. By comparing the results obtained from these evaluations, we can identify any strengths, weaknesses, or areas for improvement in each model. This comprehensive evaluation process will ensure that we have a thorough understanding of the capabilities and limitations of our models, enabling us to make informed decisions for further optimization and refinement.

3.6 Abridged Methodology

In this study, we will utilize deep learning libraries for various tasks including data visualization, data loading, data pre-processing, model importing, and visualizing the final results and predictions. To enhance the quality of training data and mitigate overfitting, we will employ data augmentation techniques. Following the application of data augmentation, the imported object detection models will undergo training using the augmented training data. Additionally, we will implement the Non-Maximum Suppression (NMS) method to regulate the number of bounding boxes predicted for each object during the construction process. Finally, the predictions' results will be displayed alongside the COCO evaluation metrics of the model and the predictions. In recent years, the field of computer vision has witnessed remarkable advancements, particularly in the domain of object detection, owing to the proliferation of convolutional neural networks (CNNs). Among the plethora of CNN-based models developed for object detection, several stand out as prominent choices, each offering unique advantages and characteristics. These models include Region-based Convolutional Neural Networks (R-CNN), Fast RCNN, Faster RCNN, Region-based Fully Convolutional Network (R-FCN), Single Shot Detector (SSD), and You Only Look Once (YOLO).

The selection of an appropriate object detection model is crucial and often depends on various factors such as the specific requirements of the application, computational efficiency, and the trade-off between accuracy and speed.

Therefore, evaluating these models comprehensively using a range of metrics is essential to make informed decisions. One of the fundamental metrics used for model evaluation is accuracy, which measures the overall correctness of the predictions made by the model. Precision and recall are additional metrics that provide insights into the model's performance. Precision quantifies the ratio of correctly predicted positive instances to the total number of positive predictions, while recall measures the ability of the model to correctly identify all positive instances from the dataset. Average precision (AP) and average recall (AR) are aggregate metrics that provide a consolidated assessment of the model's performance across different classes and thresholds. These metrics are particularly useful for evaluating the model's robustness and generalization capabilities across diverse datasets and scenarios. In addition to accuracy-related metrics, the computational efficiency of the model is also critical, especially for real-time applications and resource-constrained environments. Metrics such as running time and memory usage quantify the computational resources required by the model during inference. Understanding these metrics helps developers optimize the model's architecture and parameters to achieve the desired balance between accuracy and efficiency.

Furthermore, evaluating the performance of object detection models involves benchmarking against established datasets and standards. Common benchmark datasets such as COCO (Common Objects in Context) provide standardized evaluation protocols and benchmarks for assessing the performance of object detection models. These datasets contain annotated images with ground-truth bounding boxes and class labels, facilitating consistent and objective evaluation of different models. Ultimately, the choice of an object detection model depends on the specific requirements and constraints of the application. By conducting thorough evaluations using a diverse set of metrics and benchmark datasets, researchers and practitioners can identify the most suitable model for their needs and make informed decisions regarding model selection and deployment.

4 Literature Review

4.1 Traffic Sign Recognition

Traffic sign detection and recognition are essential components of various computer vision applications, particularly in the context of autonomous driving and driver assistance systems. This process typically involves the utilization of object detection models, wherein bounding box coordinates are utilized alongside target class information to identify and classify traffic sign objects. Several approaches have been explored by researchers, leading to successful detection and classification. Below, we outline two distinct approaches employed by researchers in this field.

Over the years, researchers have explored various methodologies for developing traffic sign detection systems. Traditional techniques such as Histogram of Oriented Gradients (HOG), Scale Invariant Feature Transform (SIFT), and

Local Binary Patterns (LBP) have been extensively studied. These methods rely on handcrafted features and conventional machine learning algorithms such as Support Vector Machines (SVM), Logistic Regression (LR), and Random Forests (RF) to detect traffic signs. However, with the advancement of computer vision and the emergence of Convolutional Neural Networks (CNNs), there has been a paradigm shift in traffic sign detection approaches. CNNs have demonstrated superior performance compared to traditional methods when applied to tasks such as object detection, including traffic sign detection. The CNN-based approach leverages the ability of deep learning models to automatically learn discriminative features directly from raw image data, eliminating the need for manual feature engineering. Empirical studies have shown that CNN-based models excel in traffic sign detection tasks, especially when evaluated on benchmark datasets such as the German Traffic Sign Recognition Benchmark (GTSRB). These datasets contain a diverse range of traffic sign images captured under different conditions, allowing researchers to evaluate the robustness and generalization capabilities of their models comprehensively. The CNN-based approach offers several advantages over traditional methods, including higher accuracy, improved scalability, and the ability to handle complex and diverse traffic sign images effectively. By leveraging deep learning techniques, researchers can develop more sophisticated and accurate traffic sign detection systems capable of meeting the stringent requirements of real-world applications such as autonomous driving, traffic surveillance, and road safety.

In summary, while traditional techniques such as HOG, SIFT, and LBP have laid the foundation for traffic sign detection, the advent of CNNs has revolutionized the field by enabling more accurate, robust, and efficient detection systems. The CNN-based approach represents a significant step forward in traffic sign detection research and holds great promise for future advancements in the field of computer vision and intelligent transportation systems. Several works have contributed significantly to the field of traffic sign detection, each proposing innovative methodologies to address the challenges associated with this task. Wang et al. introduced a method that combines coarse filtering modules based on Histogram of Oriented Gradients (HOG) with Linear Discriminant Analysis (LDA) and filtering modules utilizing HOG and Support Vector Machine (SVM) classifiers. Their approach, applied to the German Traffic Dataset, aims to efficiently detect traffic signs by leveraging both coarse and fine-grained feature extraction techniques.

Zang et al.(2016) proposed a hybrid approach that integrates a Local Binary Pattern (LBP) feature detector with an AdaBoost classifier to extract Regions of Interest (ROI) for initial coarse selection. Subsequently, cascaded Convolutional Neural Networks (CNNs) are employed to refine the ROI selection process and enhance traffic sign recognition accuracy. By combining traditional feature extraction methods with deep learning techniques, They aimed to improve both the efficiency and effectiveness of traffic sign detection systems.

Zhu et al.(2016) introduced a novel method based on a fully convolutional network (FCN) architecture, extending the concept of Region-based Convolutional Neural Networks (R-CNNs). Their approach utilizes an object proposal method called EdgeBox to generate region proposals, which are then fed into the FCN for traffic sign detection. By adopting a fully convolutional architecture, Zhu et al. aimed to achieve state-of-the-art results on the Swedish Traffic Signs Dataset, demonstrating the effectiveness of their method in handling complex traffic sign detection tasks.

These works represent notable contributions to the field of traffic sign detection, each offering unique insights and approaches to tackle the inherent challenges associated with this task. By combining traditional feature-based methods with advanced deep learning techniques, researchers continue to push the boundaries of traffic sign detection performance and pave the way for more robust and efficient detection systems in real-world applications.

4.2 Aproach A

Traffic sign recognition relies on a range of techniques within object detection models to enhance accuracy and reliability. One such technique is Non-Maximum Suppression (NMS), which eliminates redundant bounding boxes, ensuring that only the most relevant detections are retained. Additionally, data augmentation plays a crucial role by diversifying the dataset, thereby improving the model's ability to generalize to unseen scenarios.

Another important technique is Region of Interest (RoI) aligning, which ensures accurate feature extraction from detected regions, enabling the model to capture intricate details of traffic signs. Finally, fine-tuning pre-trained models is essential to adapt them to the specific task of traffic sign recognition. By fine-tuning the parameters of pre-trained models on traffic sign datasets, the model can learn to extract relevant features effectively.

These techniques, extensively studied and documented, have proven to be effective in achieving precise traffic sign recognition. By incorporating NMS, data augmentation, RoI aligning, and fine-tuning into object detection models, researchers can develop robust systems capable of accurately detecting and classifying traffic signs in diverse real-world environments.

4.2.1 Method 1

Liu & Zhang (2019) has described in their research nural networks in recognizing traffic signs due to various conditions such as occlusions, varying light conditions, resolutions, and illuminations present in the images. They note that traditional neural networks often rely solely on features from the topmost layers, resulting in the loss of edge pixels and spatial information crucial for accurate recognition. To address these limitations, Liu & Zhang propose a novel approach. They suggest combining higher resolution pixels from the initial convolutional layers with deeper layers containing stronger features. This fusion aims to construct a feature pyramid capable of detecting the semantic data of the target, thereby overcoming the disadvantages of traditional neural networks. To enhance the optimization of the neural network, introduced several techniques, including ROI aligning, soft-NMS, and an improved weighted cross-entropy loss function. These methods aimed to address challenges such as irregular circulation of categories and a small number of pixels in the target area. In region proposal-based networks, a ROI (Region of Interest) pooling layer was utilized to mitigate the issue of duplicated feature extractions. This layer processes the feature map and converts the corresponding region into a fixed-size ROI based on the coordinates of the provided candidate box. The obtained ROIs are then fed into the fully connected layers to detect and classify traffic signs accurately. The positional coordinates of candidate boxes obtained through model regression are predominantly floating-point numbers. However, fully connected layers require fixed-size inputs, posing a challenge due to the non-integer lengths of rectangular areas required for pooling. This necessitates two quantizing operations in the ROI pooling layer, impacting the semantic information within the pixels. As a solution, ROI aligning replaces ROI pooling, offering improved accuracy and detection rates without the need for quantization or breaking down candidate regions into smaller groups during mapping into feature maps. Additionally, the NMS method optimizes performance by adjusting overlapping windows to identify the best target box. Soft-NMS further refines this by applying a Gaussian weighting function to the score function, enhancing precision. Furthermore, the weighted cross-entropy loss function prioritizes important data by upgrading foreground pixels and masking background pixels. In the study's results, a recall rate of 96.5% and a precision rate of 91.03% were achieved, underscoring the effectiveness of the proposed techniques.

4.2.2 Method 2

Yashwanth et al. (2022) introduced the YOLOP model for traffic sign detection and recognition, where YOLOP stands for "You Only Look Once for Panoptic driving perception." This model features a standard encoder and four distinct decoders, each serving a specific function such as detecting traffic signs, delineating driving lanes, identifying various objects, and detecting and recognizing traffic signs. The efficacy of the model was evaluated using datasets including BDD100K (Berkley Deep Drive), GTSRB (German Traffic Sign Recognition Board), and GTSDB (German Traffic Sign Detection Board). The encoder of the YOLOP model comprises two distinct components: the backbone and the neck. The backbone is anchored by a YOLOv4 network renowned for its exceptional object detection capabilities. Leveraging YOLOv4 not only ensures robust object detection but also facilitates feature reuse and propagation, minimizing computational overhead. This backbone is shared among the remaining four encoders, optimizing computational efficiency. Additionally, the neck component merges features generated by the backbone network. It incorporates a Spatial Pyramid Pooling (SPP) layer to amalgamate features at different scales and a Feature Pyramid Network (FPN) layer to integrate features across various segmentation levels. This comprehensive integration of features enhances the model's ability to capture and understand the intricacies of the input data. The YOLOP network is equipped to classify objects at varying distances, ranging from close proximity to far distances. By fusing features across multiple scales, the network enhances its performance, ensuring accurate detection and recognition. The network consists of three decoders: the Detect head decoder, responsible for identifying bounding boxes and their associated classes using multi-scale feature maps; and the Drivable area segmentation and lane segmentation head decoders, which focus on segmentation tasks. Utilizing a segmentation-based U-Net network for predictions, these decoders effectively delineate drivable areas and lane boundaries, further bolstering the model's capabilities in understanding and interpreting complex visual scenes. The YOLOv5 algorithm was employed for object detection due to its exceptional speed in identifying objects. Leveraging this approach, the network attained a mean average precision (mAP) of 76.5% for identifying and recognizing traffic objects. Furthermore, the model achieved an intersection over union (IOU) of 91.5% and 70.5% for navigable area segmentation and lane detection, respectively. Additionally, outstanding results were obtained for traffic sign detection and recognition, with mAP scores of 99.4% and 94%, respectively. These impressive performance metrics underscore the efficacy and accuracy of the YOLOv5 algorithm in addressing the complex challenges of traffic sign detection and recognition.

4.2.3 Method 3

Shabarinath & Muralidhar (2020) proposed a Convolutional Neural Network (CNN) based on the VGGNet architecture, augmented with image preprocessing techniques. This system incorporates post-training quantization and pruning methods to optimize performance while minimizing computational overhead. Despite undergoing optimization processes, the model maintains an exceptional validation accuracy of 99.2%. The dataset utilized consists of images with dimensions of 32x32x3, featuring three color channels. To enhance model efficiency, the images undergo preprocessing steps, including conversion into single-channel grayscale representations. This conversion preserves pixel intensity, brightness, and complexion, ensuring consistency across the grayscale images. Furthermore, a local histogram equalization operation is applied to enhance image contrast, thereby improving feature visibility and aiding in traffic sign detection.

Post-training quantization techniques are employed to reduce the computational burden of the model without sacrificing accuracy. This optimization method minimizes the precision of weights and activations, leading to smaller model sizes and faster inference times. Additionally, pruning techniques are utilized to identify and eliminate redundant network parameters, further streamlining model efficiency. Despite these optimization processes, the model maintains a high level of accuracy, demonstrating its robustness and suitability for real-world applications. By leveraging a combination of preprocessing techniques and optimization methods, Shabarinath and Muralidhar's CNN-based system achieves superior performance in traffic sign detection tasks. These advancements contribute to the development of efficient and reliable traffic management systems, paving the way for safer and more intelligent transportation networks.

4.2.4 Method 4

Rajendran et al. (2019) opted for the RetinaNet algorithm for traffic sign detection and recognition due to its superior accuracy in predicting classes and detecting objects compared to other models such as YOLO and SSD. RetinaNet is equipped with a feature pyramid network and Focal Loss, which effectively reduce accuracy loss and enhance speed in object detection tasks. In their approach, the RetinaNet model is utilized for traffic sign detection, while a CNN classifier is employed for recognizing the detected traffic signs. The methodology was applied and evaluated on datasets including GTSDB (German Traffic Sign Detection Board) and GTSRB (German Traffic Sign Recognition Board), demonstrating the effectiveness of the RetinaNet-based approach in accurately detecting and recognizing traffic signs. This approach comprises three main components: a backbone network incorporating a feature pyramid network (FPN), a network for classifying objects, and a network for forming bounding boxes. The FPN is constructed entirely atop the ResNet-50 model. Operating on traffic images as input, it generates feature maps at different scales, thereby creating a feature pyramid. This feature pyramid is then divided into two pathways: the top-down pathway and the bottom-up pathway. The bottom-up pathway generates a pyramid of features containing feature maps of varying scales. The top-down pathway incorporates lateral connections and performs upsampling on the feature map. Leveraging lateral connections, the feature maps generated by both pathways are combined to produce feature maps of varying scales. The network responsible for generating bounding boxes takes candidate traffic sign boxes as inputs and generates bounding boxes around them. To mitigate regression issues, the generated bounding boxes are expanded by 25%. Subsequently, the expanded boxes are resized and cropped to a size of 48 x 48, facilitating classification. In the results, the proposed RetinaNet method achieved an accuracy of 96.46% and a mean Average Precision (mAP) of 96.7%.

4.2.5 Method 5

Tang et al. (2021) presented the Integrated Feature Pyramid Network with Feature Aggregation (IFA-FPN) as an enhancement to the Feature Pyramid Network (FPN). The IFA-FPN addresses FPN's challenges in managing highly imbalanced class structures and distributions. Three key methods were employed in the implementation of IFA-FPN. Firstly, a lightweight operation was introduced to improve the model's computational efficiency without compromising its performance. Secondly, Integrated Operation (IO) was utilized to mitigate the inequality in Region of Interests (ROIs) across different levels of the feature pyramid. Lastly, Feature Aggregation (FA) was integrated to enhance the feature maps' capability to represent features effectively, improving the model's overall performance.

The efficacy of the proposed IFA-FPN network was evaluated on several datasets, including STSD (Sweden traffic sign detection), TT100k (Tencent Tsinghua 100k), and GTSDB (German Traffic Sign Detection Board). The results showcased significant improvements, particularly when integrating the IFA-FPN method with Cascaded-RCNN. On the GTSDB dataset, the model achieved an impressive mean Average Precision (mAP) of 80.3%, demonstrating its effectiveness in accurately detecting and classifying traffic signs across different datasets.

4.2.6 Method 6

Liu et al. (2019) introduced the Multi-scale Region-based Convolutional Neural Network (MR-CNN), a novel architecture designed for traffic sign detection. The MR-CNN incorporates multiscale deconvolution operations to up-sample features extracted from deep convolutional layers. This enables the fusion of features from both deep and upper layers, resulting in a comprehensive feature map that captures multi-scale contextual information. Within the Region Proposal Network (RPN), the fused feature map enhances the resolution and semantic information related to small traffic signs. This improvement allows for more accurate detection and localization of small objects within the image. Moreover, outside the RPN, the fused features contribute to enhancing the overall feature representation, leading to improved performance in subsequent stages of the detection process.

The effectiveness of the MR-CNN architecture was evaluated using multiple datasets, including GTSDB, GTSRB, and TT100K. The model achieved impressive detection and recall accuracies, with reported values of 71.3% and 89.4%, respectively. These results demonstrate the efficacy of the proposed approach in accurately detecting traffic signs across diverse datasets, showcasing its potential for real-world applications in traffic management and autonomous driving systems.

4.2.7 Method 7

Ravindran et al. (2019) introduced the F-RCNN (Faster region-based Convolutional Neural Network) architecture, specifically tailored for the detection and classification of traffic signs. The model harnessed the power of transfer learning by leveraging pre-trained deep neural networks. Notably, in addition to detecting traffic signs, the approach incorporated the utilization of 'Tesseract' for text detection within the traffic sign images. To evaluate the performance of their approach, the researchers employed key metrics such as Mean Average Precision (mAP) and Frames Per Second (FPS). These metrics provided insights into both the accuracy and efficiency of the detection system. By measuring mAP, the model's ability to precisely locate and classify traffic signs was quantified. Concurrently, FPS assessed the speed at which the model could process images, crucial for real-time applications.

The effectiveness of the F-RCNN approach was rigorously tested using the GTSDB (German Traffic Sign Detection Benchmark) dataset. This dataset, representative of real-world scenarios, enabled comprehensive assessment of the model's performance across diverse traffic sign types and environmental conditions. Through their experimentation and evaluation, Ravindran et al. demonstrated the viability of their F-RCNN framework as a robust solution for traffic sign detection and classification tasks, offering promising prospects for practical deployment in real-world traffic management systems.

4.2.8 Method 8

Yan et al. (2022) proposed an enhanced version of the YOLOv4 model tailored specifically for traffic sign detection and classification tasks. Central to their approach was the integration of K-means clustering, a technique commonly used in unsupervised machine learning, to optimize the model's performance. The researchers conducted their evaluation using two widely recognized datasets in the field of traffic sign detection: the Chinese Traffic Sign Detection Benchmark (CCTSDB) and the German Traffic Sign Detection Benchmark (GTSDB). The utilization of these datasets allowed for a comprehensive assessment of the model's efficacy across different traffic sign types and environmental conditions.In their methodology, K-means clustering was employed to determine optimal anchor box values for the YOLOv4 model. Initially, the dimensions of the target frame served as inputs for the algorithm. Subsequently, K-means selected random centroids and assigned categories based on the distance of data points from these centroids. Through an iterative process, the algorithm refined the centroids until convergence, ultimately yielding the final anchor box values.

The enhanced YOLOv4 model yielded promising results, achieving a Mean Average Precision (mAP) of 90.42%. Furthermore, the model exhibited impressive recognition speed, processing images at a rate of 25.3 frames per second (fps). These outcomes underscore the effectiveness of the proposed approach in accurately detecting and classifying traffic signs while maintaining efficient computational performance, thereby demonstrating its potential for real-world applications in traffic management and autonomous driving systems.

4.2.9 Method 9

Al Khafaji & El Abbadi (2022) introduced a novel method that integrates the YOLOv5 (You Only Look Once) network with a Convolutional Neural Network (CNN) to enhance traffic sign detection and classification. This innovative approach was rigorously evaluated using two prominent datasets: the German Traffic Sign Recognition Benchmark (GTSRB) and the German Traffic Sign Detection Benchmark (GTSDB).

In their methodology, YOLOv5 was specifically trained on the GTSDB dataset to proficiently detect traffic signs within images. Meanwhile, the CNN component of the model underwent training on the GTSRB dataset, focusing on the classification aspect of traffic signs. To ensure compatibility and optimal performance, the GTSDB dataset was meticulously preprocessed before training. This involved converting the dataset into the YOLO format, resizing images to a standardized resolution of 640 x 640 pixels, and implementing noise removal techniques using median filtering.

The experimental results obtained from this approach yielded remarkable outcomes, showcasing a remarkable accuracy rate of 99.95% for both detecting and classifying traffic signs. This exceptional level of accuracy underscores the efficacy and robustness of the proposed methodology in accurately identifying and categorizing traffic signs within images. By seamlessly integrating YOLOv5 and CNN, this method offers a comprehensive solution for traffic sign detection and classification tasks, demonstrating its potential for real-world applications in traffic management systems and autonomous vehicles.

4.2.10 Method 10

Ibrahem et al. (2020) proposed a weakly supervised traffic sign detection system based on a convolutional neural network (CNN) framework, employing MobileNetv2 for traffic sign classification. The methodology comprises two stages aimed at efficient detection and classification of traffic signs. In the first stage, MobileNetv2 is employed to identify regions of interest within the images that require classification. Following this, in the second stage, MobileNetv2 is trained specifically to classify these identified regions. The architecture of MobileNetv2 consists of fully connected convolutional layers with 32 filters and 19 residual layers. It incorporates 3x3 convolutional layers with batch normalization and dropout layers for effective training.

To facilitate accurate classification, MobileNetv2 underwent fine-tuning on the German Traffic Sign Recognition Benchmark (GTSRB) dataset. This process enabled the model to predict traffic signs during classification on the German Traffic Sign Detection Board (GTSDB) dataset. In terms of performance, the model demonstrated promising results. It achieved a processing time of 55.04 milliseconds and attained a mean Average Precision (mAP) of 15.03 at an Intersection over Union (IoU) threshold of 0.5. These results indicate the effectiveness of the proposed weakly supervised approach for traffic sign detection and classification.

4.2.11 Method 11

Nacir et al. (2022) introduced a transfer learning model, YOLOv5, tailored for traffic sign detection and classification. This model underwent training using the GTSRB dataset for classification and the GTSDB dataset for detection. With 224 layers and 71,67,184 parameters, it occupies a compact size of 14.6 MBs. Data preprocessing involved augmenting the dataset with variations in lighting conditions, brightness, saturation levels, and occlusion levels to enhance model robustness.

The results showcased the effectiveness of the model, with a precision rate of 93.7% and a mean average precision of 94.5% achieved at an IoU threshold of 0.5. Additionally, the model exhibited a commendable recall rate of 93.8% and a minimum miss rate of 6.2%. These metrics underscore the model's ability to accurately detect and classify traffic signs, highlighting its potential for real-world applications in traffic management and autonomous driving systems.

4.2.12 Method 12

Yang et al. (2016) proposed a comprehensive two-part approach aimed at robust traffic sign recognition. The method consists of a detection section followed by a classification section, each tailored to handle specific challenges inherent in the task. In the detection section, the authors first transform coloured input images into probability maps utilizing a colour probability model. This transformation aids in highlighting potential regions of interest corresponding to traffic signs within the images. These probability maps serve as the basis for extracting traffic sign proposals, achieved by identifying extremal regions with significant probabilities indicative of potential signs. The extremal region extraction process helps in isolating candidate regions likely to contain traffic signs. Subsequently, a Support Vector Machine (SVM) trained on Colour Histogram of Oriented Gradients (HOG) features is employed to filter out false positives and classify the remaining proposals. The use of SVM allows for efficient discrimination between true positive traffic sign proposals and irrelevant objects or artefacts in the images. The integration of Colour HOG features provides the SVM with discriminative information essential for accurate classification.

Moving to the classification section, a Convolutional Neural Network (CNN) architecture is utilized to perform the task of traffic sign classification. The CNN network is designed to handle the complexities and variabilities present in traffic sign images, allowing for robust classification performance. To address the dataset's large number of classes, which consist of 43 individual traffic sign types, the authors group them into three superclasses. This grouping helps reduce the complexity of the classification task by consolidating similar signs into broader categories. To enhance the network's ability to handle variations in lighting and contrast conditions, the authors employ the Contrast Limited Adaptive Histogram Equalization (CLAHE) technique. CLAHE ensures that the network receives images with consistent contrast levels, thereby improving its generalization capabilities across diverse environmental conditions.

Despite the dataset's inherent challenges, including variations in lighting, contrast, and appearance, the proposed method achieves remarkable accuracy rates. Specifically, the prohibitory, mandatory, and danger classes, which are essential categories for traffic safety, attain high accuracies of 99.29%, 96.74%, and 97.13%, respectively. These results underscore the effectiveness of the proposed approach in accurately detecting and classifying traffic signs, thus contributing to improved road safety and intelligent transportation systems.

4.2.13 Method 13

Manzari et al. (2022) introduced a novel vision transformer architecture designed to address the challenges posed by the hierarchical structure of traffic signs in images. The proposed model leverages a pyramid transformer framework, which integrates local and global data provided by convolutions to form a feature pyramid. This hierarchical structure allows the network to learn from features at multiple scales, enabling it to better handle the imbalance in traffic sign sizes. The backbone network of the pyramid transformer is based on the Cascaded R-CNN architecture, a widely used framework for object detection tasks. By integrating the pyramid structure into the vision transformer, the model can generate multiscale feature maps, facilitating denser prediction operations.

The pyramid transformer consists of two main blocks: the Pyramid block and the normal block. At each stage of processing, the Pyramid block combines multiscale features and local information into tokens, enhancing the model's ability to capture fine-grained details across different scales. Additionally, a linear patch embedding layer is employed to introduce scale invariance, which is crucial for handling variations in traffic sign sizes. In contrast, the normal block serves to incorporate convolutional bias into the transformer architecture, ensuring that the model can effectively learn and adapt to the features extracted from the input images.

Experimental results on the GTSDB dataset demonstrate the efficacy of the proposed approach, achieving a mean Average Precision (mAP) of 77.3%. This performance highlights the effectiveness of the pyramid transformer in capturing multiscale features and addressing the challenges associated with traffic sign detection tasks.

4.2.14 Method 14

Zhang (2023) proposed leveraging the YOLOv3 algorithm as a robust solution for traffic sign detection and recognition tasks. The evaluation of YOLOv3 on the German Traffic Sign Detection Benchmark (GTSDB) dataset prioritized key performance metrics such as accuracy, recall rate, and average accuracy to assess its effectiveness.The architecture of YOLOv3 comprises three principal components: a feature layer fusion structure, Darknet-53, and Darknet53. These components are composed of 1x1 and 3x3 convolutional layers, supplemented with batch normalization and leaky ReLU activation functions. This architecture design enables YOLOv3 to effectively capture spatial dependencies and extract meaningful features from input images, facilitating accurate detection and classification of traffic signs. The evaluation results demonstrated impressive performance, with YOLOv3 achieving a mean average precision (mAP) of 98.1%. Additionally, the algorithm exhibited remarkable processing efficiency, capable of handling 69 frames per second (FPS) within a single second. These findings underscore the efficacy of the YOLOv3 algorithm in accurately detecting and identifying traffic signs in real-world scenarios, making it a promising solution for applications in traffic management and autonomous driving systems.

4.2.15 Method 15

In their study, Gámez Serna and Ruichek (2020) proposed a systematic approach comprising three key components: detection, refinement, and classification, aimed at improving traffic sign recognition. The detection and classification tasks are performed using the Mask R-CNN architecture, renowned for its robustness and accuracy in object detection tasks. Unlike traditional object detection models like Faster R-CNN, Mask R-CNN incorporates a mask branch, which enables precise localization of objects without the need for Region of Interest (RoI) pooling layers. Instead, it utilizes the RoIAlign layer, enhancing the model's efficiency in detecting traffic signs, particularly smaller objects. The refinement component of their methodology involves localization and various filtering processes to further enhance the accuracy of traffic sign detection. By incorporating these refinement techniques, the model can effectively filter out false positives and improve the localization of traffic signs in complex scenes. To evaluate the effectiveness of their approach, Gámez Serna and Ruichek utilized the German Traffic Sign Detection Benchmark (GTSDB) dataset, a widely used benchmark dataset for evaluating traffic sign detection algorithms. By leveraging this dataset, they were able to assess the performance of their methodology across a diverse range of traffic sign scenarios and environmental conditions.

Overall, their structured approach leveraging Mask R-CNN demonstrates promising results in traffic sign detection tasks. By combining precise localization, efficient detection, and effective classification techniques, their methodology offers a robust solution for real-world traffic sign recognition applications.

4.2.16 Method 16

Huang et al. (2017) introduced a novel dual-part approach for traffic sign recognition, combining feature extraction using Histogram of Gradient Variant (HOGv) with classification via Extreme Learning Machine (ELM). The HOGv feature extraction method effectively balances local and repetitive information, enhancing feature shapes crucial for traffic sign recognition. Subsequently, the ELM classifier, featuring a single hidden layer, is trained to map these features to the corresponding traffic sign classes. Notably, only the weights between the hidden and output layers are trained, with the input-to-hidden layer connection facilitating feature mapping. The pattern required for output weights is embedded within the cost function of the classifier, streamlining the training process.

This methodology was evaluated across three diverse datasets: the German Traffic Sign Recognition Benchmark dataset, Belgian Traffic Sign Classification dataset, and Revised Mapping and Assessing the State of Traffic Infrastructure (Revised MASTIF). Remarkably, the proposed approach achieved exceptional results, with a recognition rate of 99.09% on the GTSRB dataset. Moreover, it exhibited impressive efficiency, with a processing time of merely 3.2ms per frame. These results underscore the effectiveness and efficiency of the dual-part approach for traffic sign recognition tasks across various datasets.

4.2.17 Method 17

Zhu & Yan (2022) introduced a YOLOv5 model tailored for traffic sign recognition, leveraging its high accuracy and performance capabilities. The model was compared with the Single Shot Multibox Detector (SSD) to gauge its effectiveness. YOLOv5 comprises four key components: input, neck, prediction layer, and the neck part. In the input phase, images are resized to 608 x 608 x 3 dimensions. The backbone integrates a CSP module and a focus module, where the latter compresses the image's height and width. The CSP module further branches into CSP1X and CSP2X, catering to different segments of the network. The neck section combines the Feature Pyramid Network (FPN) and the Path Aggregation Network (PAN) structure. FPN operates in a top-down fashion, managing upsampling by transferring and merging information for predicted feature maps, while PAN operates in a bottom-up manner. In the prediction phase, YOLOv5 utilizes the Generalized Intersection over Union (GIoU) Loss function. In terms of results, YOLOv5 achieved 97.70% accuracy across all classes and a mean Average Precision (mAP) exceeding 90.00% for each class, at a frame rate of 30 FPS. In contrast, SSD attained an accuracy of 90.14% with a frame rate of 3.49 FPS.

4.2.18 Method 18

Sermanet and LeCun (2011) employed Convolutional Neural Networks (CNNs) for traffic sign classification, opting for CNNs over traditional methods like Histogram of Oriented Gradients (HOG) and Scale-Invariant Feature Transform (SIFT) due to CNNs' tailored design for such tasks. Their approach involved directly connecting features from the first and second stages to the classifier. Features from the first phase were extracted post-pooling and subsampling, with an additional subsampling layer leading to improved accuracies.

The first phase focused on extracting local information with fine details, while the second stage captured global shapes and structures of varying sizes. Preprocessed images were resized to 32 x 32 dimensions, resulting in a second-best accuracy of 98.97%. Furthermore, supplying the network with grayscale versions of the images enhanced accuracy to 99.17%. These results underscored the efficacy of CNNs in traffic sign classification, particularly in capturing both local and global features for accurate identification.

4.2.19 Method 19

Wei et al. (2018) proposed a transfer learning approach employing Convolutional Neural Networks (CNNs) for traffic sign detection and classification. The methodology involved training a deep CNN on extensive datasets initially, followed by training a Region CNN (RCNN) on a smaller subset of samples extracted from the original dataset. The RCNN, designed for object detection tasks, was subsequently utilized for the detection of traffic signs.

The transfer learning process leveraged a pre-trained model, initially trained on a similar task, such as a CNN trained on the CIFAR-10 dataset. With only 41 images available for stop sign detection, this approach demonstrated the effectiveness of utilizing pre-trained models to achieve robust performance even with limited training data. The RCNN model strategically focused on regions containing various objects within the images, significantly reducing computational complexity compared to traditional sliding window methods. Image preprocessing steps included resizing the input images to dimensions of 32 x 32 x 3 to facilitate efficient processing.

The evaluation of the proposed approach was conducted on the German Traffic Sign Detection Benchmark (GTSDB) dataset. The results showcased promising performance metrics, including a 95% recall rate and a 99% precision rate. Out of 100 signs present in the test dataset, 95 were correctly detected, with only 5 missed detections and 1 misclassification. The primary objective of this approach was to effectively detect and classify stop signs, demonstrating its potential utility in real-world applications such as autonomous driving systems and traffic surveillance. By harnessing the power of transfer learning and object detection techniques, Wei et al. (2018) provided a promising solution for accurate and efficient traffic sign detection tasks.

4.3 Aproach B

4.3.1 Method 1

Wang et al. (2013) proposed a robust traffic sign detection method leveraging a neural network framework integrating Histogram of Oriented Gradient (HOG) features and a coarse-to-fine sliding window approach. The objective was to achieve reliable detection performance even under challenging conditions such as poor lighting, occlusion, and low image resolution.

The methodology begins with feature extraction using the HOG method, which captures gradient orientation information within fixed-size windows across the image. These features are then fed into linear Support Vector Machines (SVMs) for classification, aiming to identify potential traffic sign regions. However, due to variations in scale and perspective, accurate classification poses challenges, especially for small or distant signs. To address this issue, the approach incorporates a two-stage filtering process. In the first stage, coarse filtering is applied to identify candidate Regions of Interest (ROIs) using small sliding windows. This initial filtering helps narrow down the search space, focusing on areas with higher likelihoods of containing traffic signs. Subsequently, a Non-Maximal Suppression (NMS) algorithm is employed to eliminate redundant neighboring ROIs and retain only the most relevant candidates.

One notable strength of the method lies in its ability to detect even the smallest signs effectively. This is achieved by combining the precision of fine filtering with the broader coverage of coarse filtering. By incorporating multiple scales and aspect ratios during the sliding window operation, the method ensures comprehensive coverage of potential sign regions across the image. Moreover, to accommodate the diverse categories and colors of traffic signs, the method adopts color HOG features. This entails computing histograms for each color channel and aggregating them collectively to form a comprehensive feature representation. By capturing both shape and color information, the model enhances recognition accuracy and robustness, particularly in scenarios involving signs with varying appearances.The proposed methodology was evaluated extensively across diverse datasets, demonstrating strong performance across different environmental conditions and sign variations. By effectively integrating feature extraction, classification, and filtering techniques, provided a comprehensive solution for traffic sign detection, offering promising results for real-world applications in traffic surveillance and autonomous driving systems.

4.4 Literature Review Summary

Traffic sign detection plays a critical role in various applications, including autonomous driving systems, traffic management, and road safety. As such, researchers continuously explore innovative methods and models to improve the accuracy and efficiency of traffic sign detection systems. Object detection models, particularly convolutional neural networks (CNNs) like Faster R-CNN, YOLO, and SSD, have dominated the field due to their ability to detect and classify objects with high precision. These models excel in handling complex scenarios and diverse traffic sign types, making them a popular choice among researchers. However, while object detection models have shown remarkable performance, there remains a significant opportunity to explore alternative methods and models beyond traditional CNN-based approaches. Traditional computer vision techniques, such as Histogram of Oriented Gradients (HOG), Scale-Invariant Feature Transform (SIFT), and Local Binary Patterns (LBP), offer potential avenues for traffic sign detection. These techniques focus on extracting meaningful features from images and can complement CNN-based approaches by providing additional context and information. Moreover, feature extraction methods, including edge detection, corner detection, and template matching, present alternative strategies for detecting traffic signs. These methods leverage specific image characteristics and patterns associated with traffic signs to identify and localize them within images. While feature extraction methods may lack the complexity and flexibility of CNNs, they can be computationally efficient and suitable for applications with resource constraints.

Hybrid models that combine elements of both traditional computer vision techniques and deep learning approaches represent another promising direction in traffic sign detection research. By leveraging the strengths of each method, hybrid models can achieve improved performance and robustness across different traffic sign detection scenarios. For example, researchers may explore integrating CNN-based feature extraction with classical machine learning algorithms for classification or refining object proposals generated by CNNs using geometric constraints. Diversifying research efforts beyond object detection models can provide several benefits to the field of traffic sign detection. Firstly, alternative methods and models offer new perspectives and insights into the problem domain, fostering innovation and creativity in algorithm development. Additionally, exploring a wider range of approaches enables researchers to address specific challenges and limitations associated with object detection models, such as computational complexity, dataset biases, and domain adaptation issues. Furthermore, alternative methods and models may offer more interpretable and explainable results, which are essential for applications requiring transparency and accountability, such as autonomous vehicles operating in real-world environments. By considering a diverse set of approaches, researchers can develop more robust and versatile traffic sign detection systems capable of addressing the complexities and nuances of real-world scenarios.

In conclusion, while object detection models remain at the forefront of traffic sign detection research, exploring alternative methods and models is essential for advancing the field. By embracing a diverse range of approaches, researchers can unlock new insights, improve performance, and develop more robust and adaptable traffic sign detection systems for various applications.

5 Methodology

5.1 Overview

The current approach to traffic sign detection predominantly revolves around the utilization of object detection models, which have proven to be effective in recognizing traffic signs. However, this approach, while reliable, may have limitations in terms of its complexity and sophistication. To overcome these limitations and further enhance the accuracy and efficiency of traffic sign recognition, it is essential to explore additional methodologies and strategies. One potential avenue for improvement is the fine-tuning of hyperparameters within existing object detection models. By fine-tuning parameters such as learning rates, batch sizes, and optimization algorithms, researchers can optimize model performance for specific traffic sign detection tasks. Additionally, exploring different model architectures, such as variations of convolutional neural networks (CNNs) or attention mechanisms, may offer new insights and improvements in accuracy. Another promising approach is leveraging transfer learning, where pre-trained models trained on large datasets are adapted to perform traffic sign recognition tasks. By transferring knowledge from domains with ample data to domains with limited data, transfer learning can facilitate faster convergence and improved generalization performance. Furthermore, considering the integration of other modalities, such as temporal data or contextual information, could lead to significant enhancements in traffic sign detection systems. Temporal data, such as video streams from traffic cameras, can provide valuable temporal context and motion information, enabling more robust and accurate detection of dynamic traffic signs. Likewise, incorporating contextual information, such as road layout, weather conditions, and surrounding objects, can improve the model's ability to recognize traffic signs in diverse and complex environments.

In summary, while object detection models serve as a strong foundation for traffic sign detection, further advancements can be achieved by exploring additional methodologies such as fine-tuning hyperparameters, experimenting with different architectures, leveraging transfer learning, and integrating other modalities. By embracing these strategies, researchers can push the boundaries of traffic sign recognition and develop more robust and efficient detection systems for real-world applications.

5.2 Research Framework

The detection of traffic signals in images involves a series of steps outlined below:

Step 1: Data Preparation Firstly, we need to prepare the data to meet the requirements of our model. This includes gathering a dataset containing images with annotated traffic signals. The annotations typically include bounding box coordinates indicating the location of each traffic signal in the image.

Step 2: Model Development and Training Next, we develop and train the model for traffic signal detection. This involves designing a neural network architecture suitable for detecting traffic signals in images. The model is trained using the annotated dataset prepared in the previous step. During training, the model learns to recognize and localize traffic signals in images by adjusting its parameters based on the annotated data. Training involves iteratively feeding batches of images and their corresponding annotations to the model, optimizing its parameters to minimize the difference between predicted and ground truth annotations. The trained model can then be evaluated and fine-tuned as needed to improve its performance on new, unseen data.

In the initial phase of our traffic sign detection project, we undertake data preparation as the primary step. This involves several key sub-steps to ensure that our data is properly formatted and ready for model training. First, we create a Python dictionary and populate it with image names as keys and corresponding traffic sign coordinates and class values as values. However, we observed that not all images contain sign coordinates upon inspecting the gt.txt file. As a result, we filter out images lacking this information and focus only on those with complete annotations for model training. These selected images are then segregated into a separate folder to streamline the training process. Subsequently, we design a custom dataset class in PyTorch to facilitate the loading of our prepared data. Additionally, we implement data augmentation techniques using PyTorch transforms to enhance the diversity and robustness of our dataset. Finally, we load the dataset by invoking the custom dataset class, ensuring that the data is effectively organized and ready for use in training our traffic sign detection model. This meticulous data preparation phase lays a solid foundation for the subsequent development and training stages of our model.

Parameter	**Value**
Optimizer	Stochastic Gradient Descent
Learning Rate Scheduler	CosineAnnealingWarmRestarts
Number of Epochs	10
Learning Rate	0.0005

Table 1: Model Training Parameters

In the beginning stages of our project on detecting traffic signs, we start by preparing our data, which is a crucial initial step. This process involves a series of important tasks aimed at ensuring that our data is well-organized and suitable for training our model. Firstly, we create a Python dictionary where we assign image names as keys and pair them with the corresponding coordinates and class values of the traffic signs. However, we encountered a challenge during this process when we discovered that not all images contained the necessary sign coordinates as we inspected the gt.txt file. Consequently, we decided to filter out the images lacking this vital information and focus solely on those with complete annotations for our model's training. These carefully selected images are then separated into a distinct folder to streamline our training efforts. Following this, we develop a customized dataset class using PyTorch to efficiently load our prepared data. Additionally, we integrate various data augmentation techniques using PyTorch transforms to enhance the diversity and resilience of our dataset. Finally, we load the dataset using the custom dataset class, ensuring that our data is well-prepared and ready for training our traffic sign detection model. This metic Raw Data Acquisitionulous approach to data preparation establishes a strong foundation for the subsequent stages of model development and training.

5.3 Raw Data Acquisition

The image data used for generating predictions consists of images in the Portable Pixmap (PPM) format. Alongside this dataset is an annotations file that includes coordinates indicating the bounding boxes and the target class for objects representing traffic signs. Importantly, all the data used in this project was sourced from the official website of the German Traffic Sign Detection Benchmark (GTSDB) dataset and is freely available as open-source. This dataset has been thoroughly examined and cited in various studies, including by Houben et al. (2013b).

5.4 Data Wrangling and Preprocessing

The annotations text file provided in the dataset served as the foundation for data organization. Each image name listed in the file was used as a key in a dictionary, with the corresponding bounding box coordinates and class names stored as the values for those keys. Subsequently, all images within the folder were relocated to a new directory, but only if their names were found within the aforementioned dictionary. This initial data wrangling process established the groundwork for subsequent steps, particularly data preprocessing.

Moving forward, data preprocessing entailed the definition of an object detector class responsible for reading the images, converting them to the PIL (Python Imaging Library) image format, and extracting the associated bounding box coordinates. To augment the dataset and diversify training samples, a technique called random horizontal flip was employed with a 50% probability, thereby increasing the quantity of training data available for model training.

5.5 Dataset

The dataset utilized for this project is the German Traffic Sign Detection Benchmark dataset, often referred to as GTSDB. It consists of 600 images, each accompanied by a ground truth text file named ”gt.txt.” This text file contains crucial information such as the coordinates of the bounding boxes encompassing the traffic signs present in the images. Notably, the dataset encompasses a wide array of traffic signs, comprising a total of 43 distinct classes. Upon acquiring the dataset from the designated source, it is organized within a main folder titled ”data.” Within this main folder, two primary components are found: the ”gt.txt” file and a subfolder named ”images.” The ”gt.txt” file serves as a repository for the ground truth annotations, facilitating the accurate localization of traffic signs within the images. On the other hand, the ”images” subfolder contains all the image files in the .ppm format. These image files are the focal point of the dataset, providing the visual data necessary for training and evaluating traffic sign detection models.

This structured organization of the dataset ensures easy access to both the ground truth annotations and the corresponding image data, streamlining the process of data preprocessing, model training, and evaluation. With this dataset structure in place, researchers and practitioners can effectively leverage the GTSDB dataset for various traffic sign detection tasks, ranging from algorithm development to performance evaluation and benchmarking.

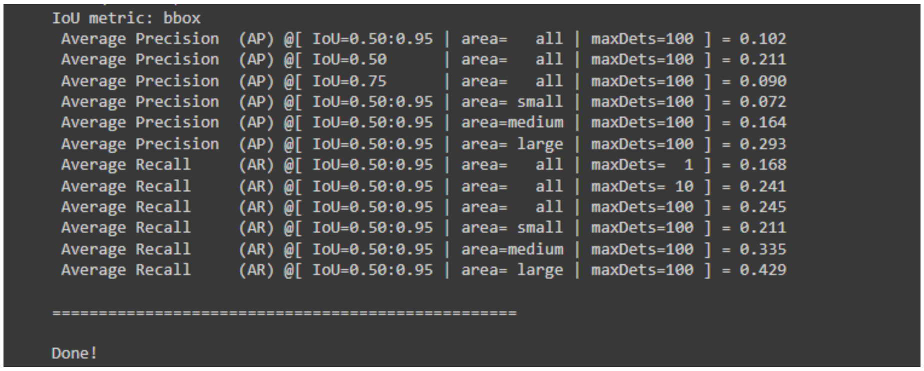
6 Evaluation

In evaluating our developed model, we employed COCO detection evaluation metrics, which offer robust standards for assessing object detection algorithms. The metrics utilized include Average Precision (AP) and Average Recall (AR), both crucial in quantifying the performance of object detection models.

Average Precision (AP) measures the precision of detection across various recall levels. It is calculated by computing the area under the precision-recall curve (AP curve). This metric provides insights into how well the model identifies objects across different confidence thresholds, with higher AP values indicating better performance. Average Recall (AR), on the other hand, quantifies the ability of the model to detect objects across all ground truth instances. It represents the average recall value obtained at predefined intervals of average precision, providing a comprehensive assessment of the model's detection capability.

By leveraging these COCO metrics, we gain a comprehensive understanding of our model's performance, enabling us to make informed decisions regarding its effectiveness in detecting objects within the given dataset.

In object detection evaluation, True Positive (TP) refers to the correct identification of objects by the model. Specifically, TP occurs when the Intersection over Union (IoU) between the predicted bounding box and the ground truth bounding box exceeds or equals a predefined threshold, indicating a success-



```
IoU metric: bbox
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.102
 Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.211
 Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.090
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.072
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.164
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.293
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.168
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.241
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.245
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.211
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.335
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.429

==================================================

Done!
```

Figure 3: : Sample of the evaluation output

ful detection. Conversely, False Positive (FP) describes cases where the model incorrectly identifies objects. FP arises when the IoU between the predicted bounding box and the ground truth bounding box falls below the threshold, signifying a false detection.

Average Precision (AP) is a metric that quantifies the accuracy of object detection by measuring the number of true positive detections among the predicted bounding boxes. It reflects the precision of the model across various confidence thresholds. Average Recall (AR), on the other hand, assesses the model's ability to detect objects by determining the proportion of true positive detections out of all possible positive instances. It provides insights into the model's overall detection performance across different thresholds. By understanding these definitions and metrics, we can effectively evaluate and analyze the performance of object detection models, enabling us to make informed decisions regarding their effectiveness and reliability in real-world applications.

The image below illustrates a sample evaluation of the model obtained during a specific iteration or epoch of training on the dataset.

The image above provides an illustration for the following discussion. The Average Precision (AP) at IoU (Intersection over Union) ranging from 0.5 to 0.95 for objects with a large area is calculated to be 0.800. This indicates that when the model identifies an object with a large area, it correctly matches the ground truth objects approximately 80% of the time. Similarly, the Average Recall (AR) at IoU ranging from 0.5 to 0.95 for objects with a large area is also calculated to be 0.800. This signifies that the model successfully detects around 80% of objects with a large area.

The evaluation process also includes analyzing the loss curves post-training to gain further insights into the model's performance. In this context, the loss curve serves as a crucial diagnostic tool, allowing us to assess the effectiveness of the model's training regimen and its ability to minimize errors.

The principle guiding our interpretation of the loss curve is straightforward: lower loss values indicate better model performance. By plotting the loss curve,

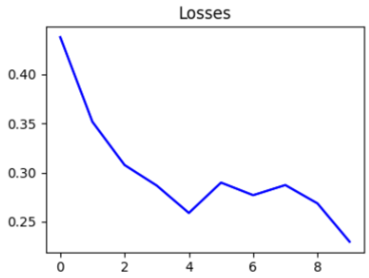


Figure 4: Loss Curve after training for 10 epochs

we can visualize how the loss metric evolves over the course of training epochs, providing a comprehensive overview of the model's learning trajectory and convergence.

The image below illustrates the loss curve obtained after successfully training the model for 10 epochs. Through this visualization, we can observe the trend of the loss metric over time, discerning whether the model exhibits consistent improvement, plateaus, or experiences fluctuations in performance. This analysis aids in determining the optimal training duration and identifying any potential issues such as overfitting or underfitting, thereby informing further adjustments or iterations in the training process. The loss curves we analyze include Loss Box Reg, Loss RPN Box Reg, Loss Classifier, and Loss Objectness. Each loss type serves a specific purpose in training our model. Loss Box Reg represents the regression loss for bounding box localization, guiding the model to accurately predict the coordinates of bounding boxes around objects. Loss RPN Box Reg focuses on the bounding box regression loss specifically for region proposal network (RPN) regions. Loss Classifier measures the loss incurred in classifying objects into different categories, aiding the model in correctly identifying the types of objects present in the image. Loss Objectness quantifies the loss related to predicting whether an object is present in a given region, helping the model distinguish between object and background regions effectively. Understanding these loss types is crucial for assessing the performance and training progress of our detection model.

Loss Box Reg assesses the accuracy of the model in predicting the bounding box around the true object, indicating how closely the predicted bounding box aligns with the ground truth. Loss RPN Box Reg evaluates the effectiveness of the network in generating region proposals, reflecting how well the model identifies potential object regions. Loss Classifier gauges the model's proficiency in classifying objects within the detected bounding boxes, providing insight into its ability to correctly label object types. Loss Objectness measures the network's performance in identifying bounding boxes containing objects, helping to distinguish between object and background regions. The respective plots

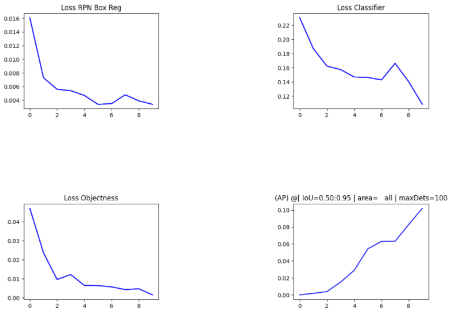


Figure 5: Various Loss Curves used in evaluation after training for 10 epochs

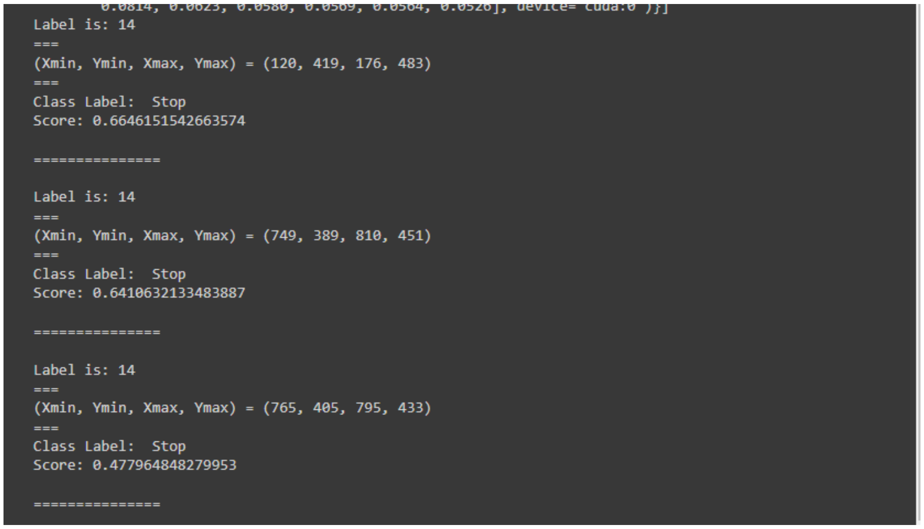
illustrating these loss metrics are presented below, generated after training the data on the model for 1000 epochs.

7 Results

The evaluation of our model using COCO metrics yielded insightful results. When considering the Average Precision (AP) values across a range of Intersection over Union (IoU) thresholds from 0.50 to 0.95, we found that our model achieved an accuracy of 10.2% in accurately predicting traffic sign objects with bounding boxes that overlapped ground truth boxes by 50% to 95% of their area. Similarly, the Average Recall (AR) values indicated that our model could correctly identify 16.8% of traffic sign objects within the entire image area with a maximum of 1 detection, under similar IoU conditions.

However, it's worth noting that our model's performance in precisely predicting objects similar to ground truth boxes, especially within a large area with a maximum of 100 detections, was considerably lower, with both AP and AR values reaching zero. Nonetheless, when focusing specifically on IoU thresholds of 0.50 and 0.50-0.95, covering a large area with a maximum of 100 detections, our model achieved promising AP and AR values of 21.1% and 42.9%, respectively.

The textual output of the predicted images provides detailed information,



```
         0.0814, 0.0623, 0.0580, 0.0569, 0.0564, 0.0526], device='cuda:0')}]
    Label is: 14
    ===
    (Xmin, Ymin, Xmax, Ymax) = (120, 419, 176, 483)
    ===
    Class Label:  Stop
    Score: 0.6646151542663574

    ===============

    Label is: 14
    ===
    (Xmin, Ymin, Xmax, Ymax) = (749, 389, 810, 451)
    ===
    Class Label:  Stop
    Score: 0.6410632133483887

    ===============

    Label is: 14
    ===
    (Xmin, Ymin, Xmax, Ymax) = (765, 405, 795, 433)
    ===
    Class Label:  Stop
    Score: 0.477964848279953

    ===============
```

Figure 6: Textual output of first image

including the label number, label name, predicted bounding box coordinates, and associated score, offering valuable insights into the model's predictions and performance.

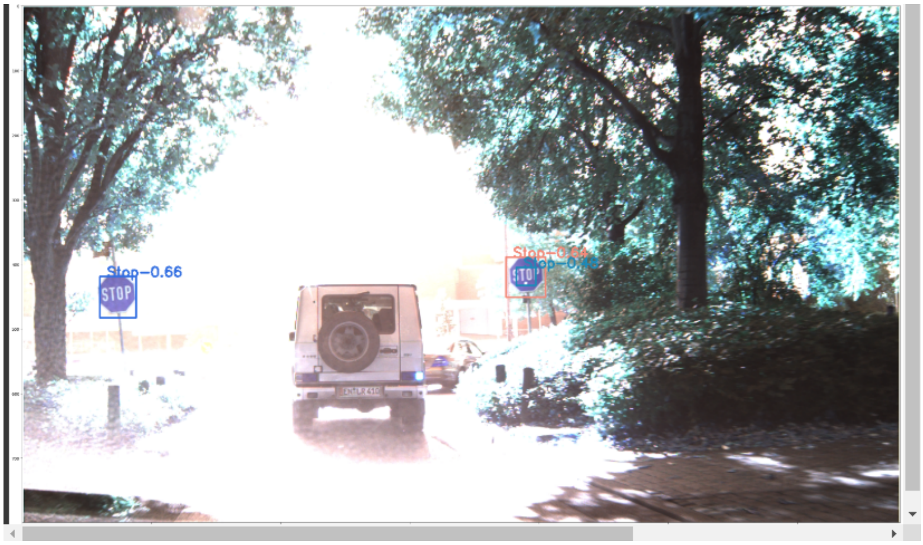


Figure 7: Correct detection and recognition

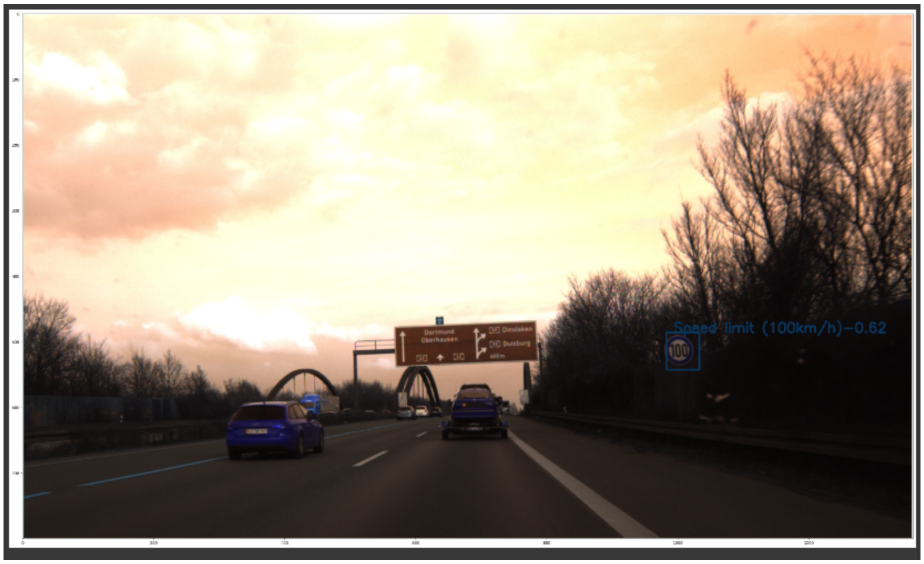
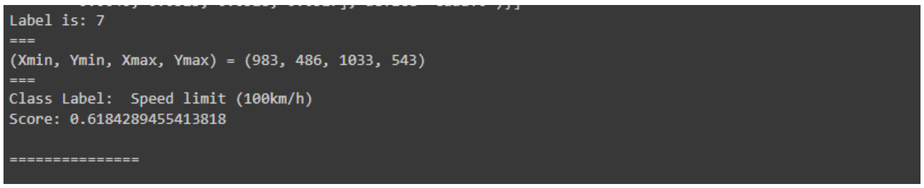


Figure 8: Correct detection and recognition



```
Label is: 7
===
(Xmin, Ymin, Xmax, Ymax) = (983, 486, 1033, 543)
===
Class Label:  Speed limit (100km/h)
Score: 0.6184289455413818

===============
```

Figure 9: Textual output of second image

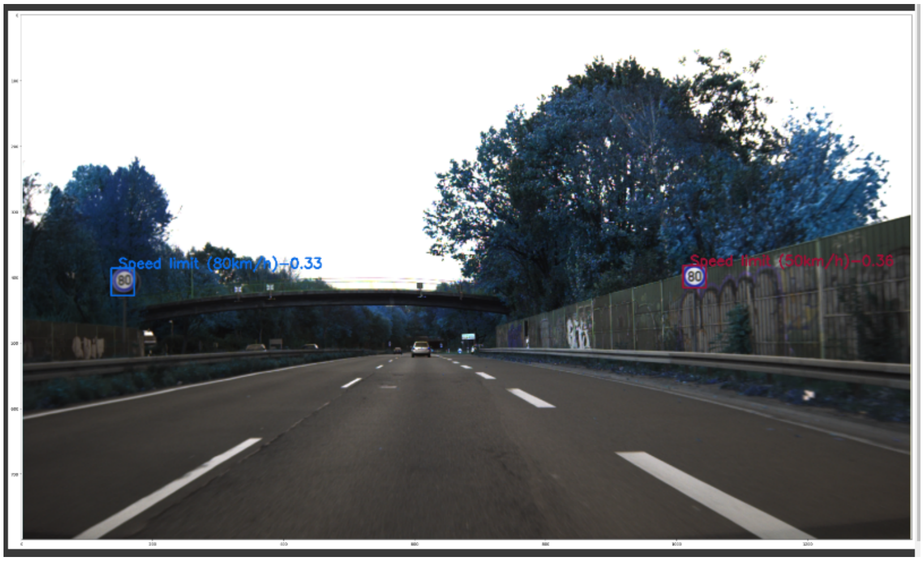
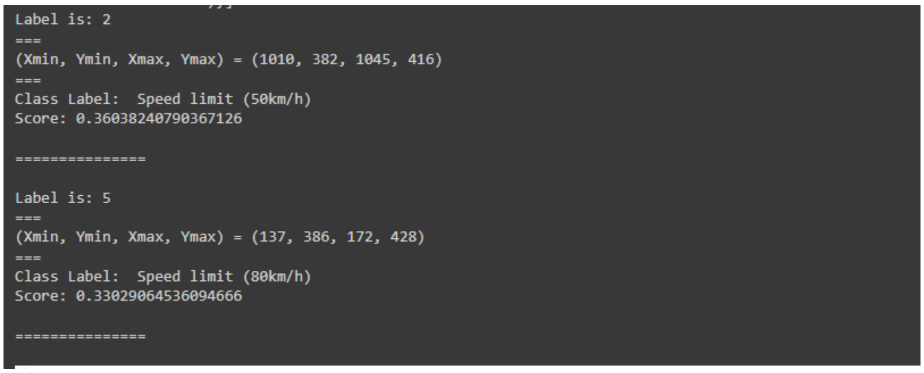


Figure 10: Example of correct detection and wrong classification



```
Label is: 2
===
(Xmin, Ymin, Xmax, Ymax) = (1010, 382, 1045, 416)
===
Class Label:  Speed limit (50km/h)
Score: 0.36038240790367126

===============

Label is: 5
===
(Xmin, Ymin, Xmax, Ymax) = (137, 386, 172, 428)
===
Class Label:  Speed limit (80km/h)
Score: 0.33029064536094666

===============
```

Figure 11: Textual output of third image

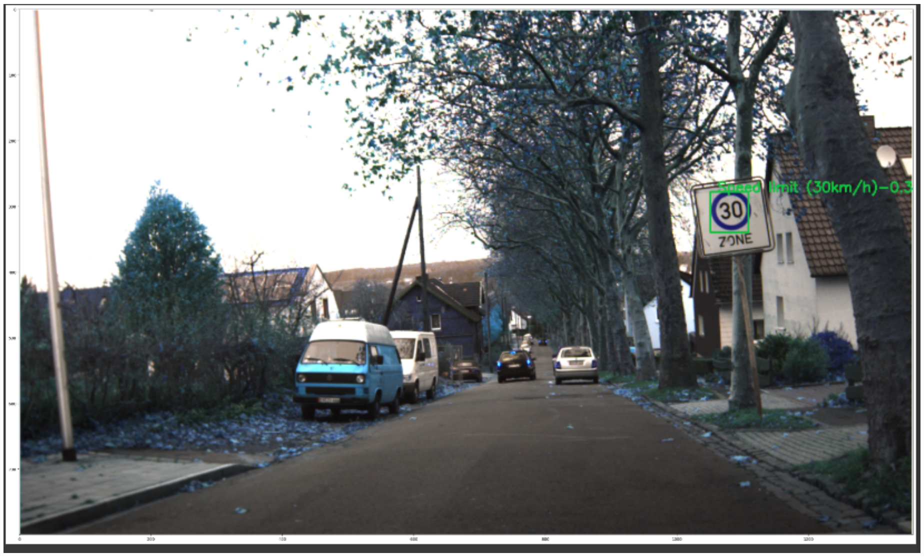
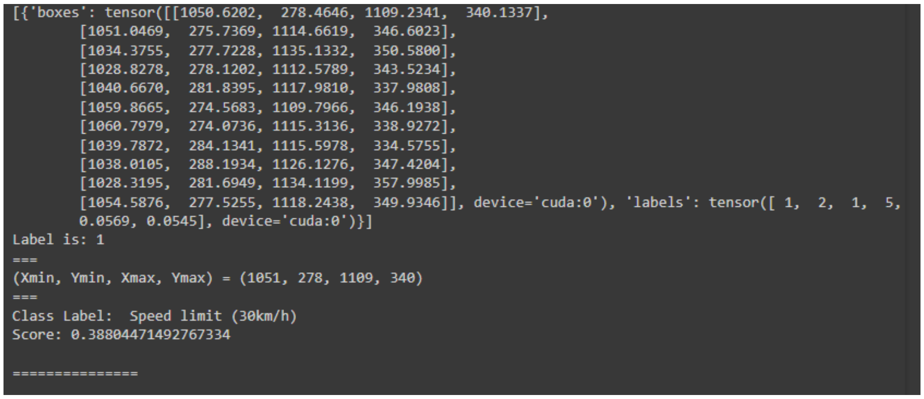


Figure 12: Correct detection and recognition



```
[{'boxes': tensor([[1050.6202,  278.4646, 1109.2341,  340.1337],
        [1051.0469,  275.7369, 1114.6619,  346.6023],
        [1034.3755,  277.7228, 1135.1332,  350.5800],
        [1028.8278,  278.1202, 1112.5789,  343.5234],
        [1040.6670,  281.8395, 1117.9810,  337.9808],
        [1059.8665,  274.5683, 1109.7966,  346.1938],
        [1060.7979,  274.0736, 1115.3136,  338.9272],
        [1039.7872,  284.1341, 1115.5978,  334.5755],
        [1038.0105,  288.1934, 1126.1276,  347.4204],
        [1028.3195,  281.6949, 1134.1199,  357.9985],
        [1054.5876,  277.5255, 1118.2438,  349.9346]], device='cuda:0'), 'labels': tensor([ 1,  2,  1,  5,
        0.0569, 0.0545], device='cuda:0')}]
Label is: 1
===
(Xmin, Ymin, Xmax, Ymax) = (1051, 278, 1109, 340)
===
Class Label:  Speed limit (30km/h)
Score: 0.38804471492767334

===============
```

Figure 13: Textual output of fourth image

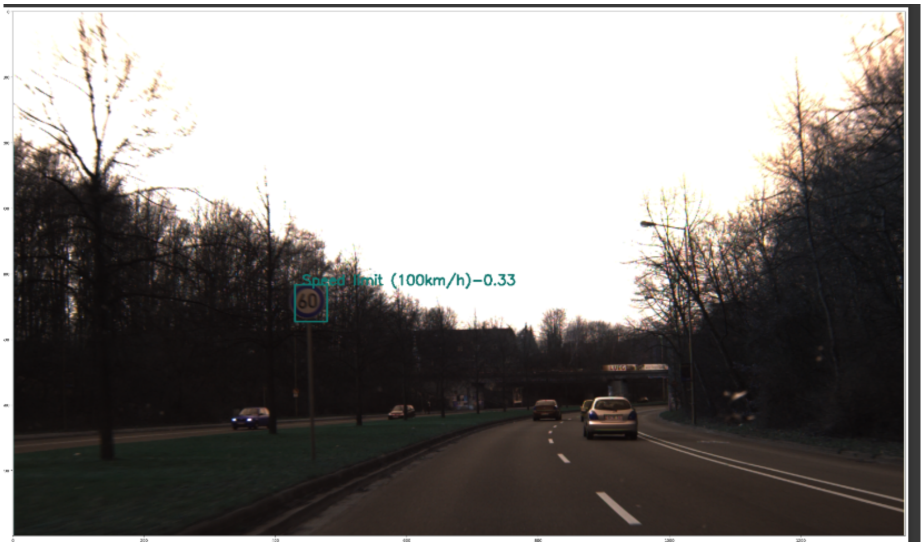


Figure 14: Example of correct detection and wrong classification

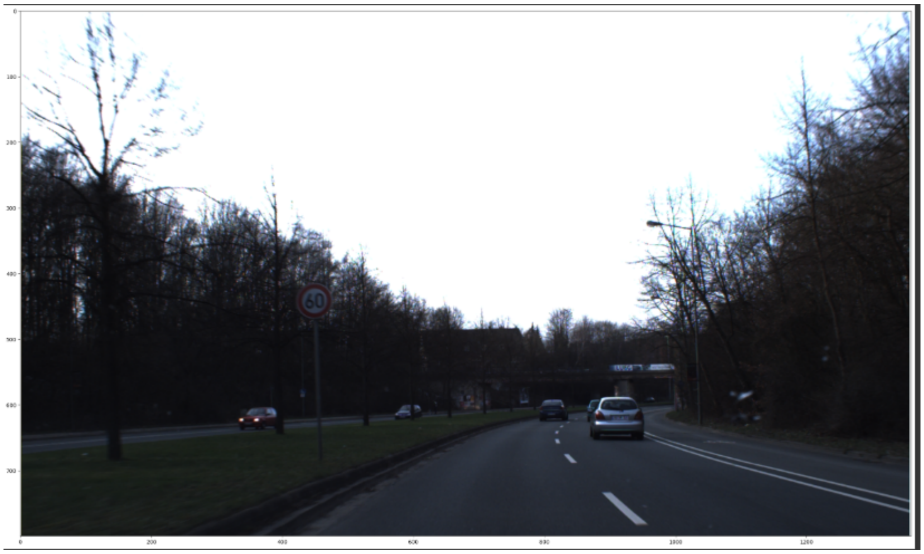


Figure 15: Real image

8 Conclusion and Future work

In this study, we conducted an experiment aimed at detecting traffic signs using the Faster Region-based Convolutional Neural Networks (Faster R-CNN) architecture. We employed the benchmark dataset of German traffic signs and utilized ResNet50 as the feature extractor for our model.

To evaluate the performance of our model, we employed COCO detection evaluation techniques, including average precision and average recall, along with analysis of various loss curves. Our findings can be summarized as follows:

a. The model exhibited suboptimal performance for areas classified as medium and small. This limitation may be attributed to the relatively small size of our dataset.

b. Analysis of the Loss Box Reg curve revealed that the model effectively fits bounding boxes tightly around detected objects.

c. Evaluation of the Loss RPN Box Reg curve indicated that additional training may be necessary to reduce loss further. This suggests that augmenting the dataset with more data could significantly enhance results.

d. Examination of the Loss Classifier curve demonstrated that the model excels in classifying objects within detected bounding boxes.

e. Analysis of the Loss Objectness curve revealed that the model performs well in detecting objects.

For future work, we plan to explore more advanced techniques such as Mask R-CNN, an improved version of Faster R-CNN, for traffic sign detection. Additionally, we intend to extend the training duration to further refine the model's performance. To achieve this, we aim to leverage more powerful GPUs to expedite the training process.

Moreover, we acknowledge that our model has yielded incorrect predictions in some instances. Therefore, we aim to address this issue and enhance the model's accuracy.

Furthermore, we aspire to implement real-time testing of our model by developing an interface that connects to a camera module. This will enable us to assess the model's performance in real-world scenarios, particularly when deployed on roads.

9 Refrence

Al Khafaji, Y. A. & El Abbadi, N. K. (2022), Traffic signs detection and recognition using a combination of yolo and cnn, in '2022 Iraqi International Conference on Communication and Information Technologies (IICCIT)', pp. 328–334.

G´amez Serna, C. & Ruichek, Y. (2020), 'Traffic signs detection and classification for european urban environments', IEEE Transactions on Intelligent Transportation Systems 21(10), 4388–4399.

Houben, S., Stallkamp, J., Salmen, J., Schlipsing, M. & Igel, C. (2013a), Detection of traffic signs in real-world images: The german traffic sign detection benchmark, in ‘The 2013 International Joint Conference on Neural Networks (IJCNN)’, pp. 1–8.

Houben, S., Stallkamp, J., Salmen, J., Schlipsing, M. & Igel, C. (2013b), Detection of traffic signs in real-world images: The German Traffic Sign Detection Benchmark, in ‘International Joint Conference on Neural Networks’, number 1288.

Huang, Z., Yu, Y., Gu, J. & Liu, H. (2017), ‘An efficient method for traffic sign recognition based on extreme learning machine’, IEEE Transactions on Cybernetics 47(4), 920–933.

Ibrahem, H., Salem, A. & Kang, H. S. (2020), Weakly supervised traffic sign detection in real time using single cnn architecture for multiple purposes, in ‘2020 IEEE International Conference on Consumer Electronics (ICCE)’, pp. 1–4.

Liu, J. & Zhang, C. (2019), A multi-scale neural network for traffic sign detection based on pyramid feature maps, in ‘2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)’, pp. 1851–1857.

Liu, Z., Du, J., Tian, F. & Wen, J. (2019), ‘Mr-cnn: A multi-scale region-based convolutional neural network for small traffic sign recognition’, IEEE Access 7, 57120– 57128.

Manzari, O. N., Boudesh, A. & Shokouhi, S. B. (2022), Pyramid transformer for traffic sign detection, in ‘2022 12th International Conference on Computer and Knowledge Engineering (ICCKE)’, pp. 112–116.

Nacir, O., Amna, M., Imen, W. & Hamdi, B. (2022), Yolo v5 for traffic sign recognition and detection using transfer learning, in ‘2022 IEEE International Conference on Electrical Sciences and Technologies in Maghreb (CISTEM)’, Vol. 4, pp. 1–4. PyTorch (2023), ‘TorchVision Object Detection Finetuning Tutorial x2014; PyTorch Tutorials 2.0.1+cu117 documentation — pytorch.org’, https://pytorch.org/tutorials/intermediate/torchvisiontutorial.html.[Accessed13 09 2023].

Rajendran, S. P., Shine, L., Pradeep, R. & Vijayaraghavan, S. (2019), Fast and accurate traffic sign recognition for self driving cars using retinanet based detector, in ‘2019 International Conference on Communication and Electronics Systems (ICCES)’, pp. 784–790.

Ravindran, R., Santora, M. J., Faied, M. & Fanaei, M. (2019), Traffic sign identification using deep learning, in ‘2019 International Conference on Computational Science and Computational Intelligence (CSCI)’, pp. 318–323.

Sermanet, P. & LeCun, Y. (2011), Traffic sign recognition with multi-scale convolutional networks, in ‘The 2011 International Joint Conference on Neural Networks’, pp. 2809–2813.

Shabarinath, B. B. & Muralidhar, P. (2020), Convolutional neural network based traffic-sign classifier optimized for edge inference, in ‘2020 IEEE REGION 10 CONFERENCE (TENCON)’, pp. 420–425.

Stallkamp, J., Schlipsing, M., Salmen, J. & Igel, C. (2011), The german traffic sign recognition benchmark: A multi-class classification competition, in

‘The 2011 International Joint Conference on Neural Networks’, pp. 1453–1460.

Tang, Q., Cao, G. & Jo, K.-H. (2021), ‘Integrated feature pyramid network with feature aggregation for traffic sign detection’, IEEE Access 9, 117784–117794.

Wang, G., Ren, G., Wu, Z., Zhao, Y. & Jiang, L. (2013), A robust, coarse-to-fine traffic sign detection method, in ‘The 2013 International Joint Conference on Neural Networks (IJCNN)’, pp. 1–5.

Wei, L., Runge, L. & Xiaolei, L. (2018), Traffic sign detection and recognition via transfer learning, in ‘2018 Chinese Control And Decision Conference (CCDC)’, pp. 5884–5887.

Yan, W., Yang, G., Zhang, W. & Liu, L. (2022), Traffic sign recognition using yolov4, in ‘2022 7th International Conference on Intelligent Computing and Signal Processing (ICSP)’, pp. 909–913.

Yang, Y., Luo, H., Xu, H. & Wu, F. (2016), ‘Towards real-time traffic sign detection and classification’, IEEE Transactions on Intelligent Transportation Systems 17(7), 2022–2031.

Yashwanth, S. D., Rao, S. V., Rakshit, Meharwade, Y. P. & Kivade, R. (2022), Autonomous driving using yolop, in ‘2022 IEEE North Karnataka Subsection Flagship International Conference (NKCon)’, pp. 1–6.

Zhang, X. (2023), Traffic sign detection based on yolo v3, in ‘2023 IEEE 3rd International Conference on Power, Electronics and Computer Applications (ICPECA)’, pp. 1044–1048.

Zhu, Y. & Yan, W. Q. (2022), ‘Traffic sign recognition based on deep learning’, Multimedia Tools and Applications 81(13), 17779–17791.

URL: https://doi.org/10.1007/s11042-022-12163-0