Dissertation Title: Comparative analysis of YOLOv8, YOLOv10 and Faster R-CNN models for vehicle detection in low-light environment

Masters title: MSc Data Analytics

Name: Bimal Paudel

Year: 2025

Statement of compliance with academic ethics and the avoidance of plagiarism

I honestly declare that this dissertation is entirely my own work and none of its part has been copied from printed or electronic sources, translated from foreign sources, and reproduced from essays of other researchers or students. Wherever I have been based on ideas or other people texts I clearly declare it through the good use of references following academic ethics.
(In the case that is proved that part of the essay does not constitute an original work, but a copy of an already published essay or from another source, the student will be expelled permanently from the postgraduate program).

Name and Surname:
BIMAL PAUDEL

Date: 15.06.2025

Abstract

This study addresses the challenges of vehicle detection in low-light conditions and performs a comparative analysis by training state-of-the-art object detection models – YOLOv8, YOLOv10 and Faster R-CNN models on primary vehicle dataset and then secondary dataset. Initially, models trained on the primary dataset showed low mean Average Precision (mAP) and unstable performance, with YOLOv8 at 20% and the Faster R-CNN model at 9%. While data augmentation improved YOLOv8's accuracy by 5%, its impact on YOLOv10 and FRCNN was negligible. Crucially, the introduction of a larger, more diverse secondary dataset led to a dramatic increase in detection accuracy across all models; YOLOv8 and YOLOv10 achieved over 90% mAP@50, demonstrating superior performance and efficiency compared to Faster R-CNN. The research conclusively shows the importance of data quality and quantity for achieving high-performance low-light object detection and presents the accuracy and computational efficiency of YOLOv8 model as the most suitable for real-world deployment.

Keywords: Low-Light Object Detection, YOLOv8, YOLOv10, Faster R-CNN, Deep Learning

Acknowledgments

First and foremost, I would like to express my sincere gratitude to my supervisor Dr. Anuj Batta for his support and guidance throughout the course of this dissertation. His prompt responses whenever I required feedback were instrumental in completing this research. I am especially thankful to my brother Keshab Paudel and my friend Nayan Kandel – who generously dedicated their time to help me collect the primary dataset. I could not have completed this research without their invaluable assistance and unwavering support. Finally, I would like to thank my family for their constant encouragement and understanding during the demanding phases of this dissertation.

Table of Contents

Introduction

Vehicle detection plays an important role in intelligent transportation systems, traffic monitoring, and autonomous driving. However, detecting vehicles in low-light environments remains a significant challenge due to multiple factors such as poor illumination and low environmental contrast (Balaji et al., 2025). Traditional computer vision techniques often fail to perform reliably under these conditions, necessitating the use of advanced deep learning models capable of robust feature extraction.

Among the most prominent object detection architectures, YOLO and Faster R-CNN represent two of them, derived from the same technique of image processing i.e. convolution, to achieve the same goal. YOLOv10 (You Only Look Once) is a recent, one-shot object detector optimized for speed and efficiency. In contrast, Faster R-CNN, a two-stage detector, is known for its high accuracy, although at the cost of slower inference rate (Ren et al., 2016). While both models have demonstrated impressive performance under standard lighting, their effectiveness in adverse conditions such as low-light scenarios remains under-explored (Sivasubramanian et al., 2023).

This research aims to conduct a comparative analysis of YOLOv8, YOLOv10 and Faster R-CNN for vehicle detection in low-light environments. By evaluating their performance across key metrics—such as precision, recall, and inference time—this study seeks to identify the strengths and limitations of each model. The findings contribute to developing more resilient computer vision systems for real-world, low-visibility applications.

1.1 Trigger and Rationale

The reason for this research comes from a deep fascination with the evolving field of deep learning, particularly its applications in computer vision. As various developments in automation and object detection was observed, I became increasingly curious about the algorithms that drive these intelligent systems. The ability of machines to perceive and interpret visual data in real time, especially in challenging environments, was the inspiration to delve deeper into the technology behind it.

This curiosity led to the study state-of-the-art detection models like YOLO and Faster R-CNN, which have significantly influenced the field. By comparing the latest YOLO models with the widely adopted Faster R-CNN, this research aim to understand how these models perform in low-light conditions—an area with practical importance and ongoing research potential. This topic not only aligns with researcher's academic interests but also reflects where the future of intelligent vision systems is headed.

1.2 Research Questions

The following are the research questions this document aims to answer by the end of the project.

i. How do YOLOv10, YOLOv8, and Faster R-CNN compare in terms of detection accuracy precision, and recall for vehicle detection in low-light conditions?

ii. Which of the evaluated models show the most practical suitability for real-world low-light applications, such as autonomous driving and traffic management and surveillance?

iv. What is the performance difference between YOLOv8 and the newer YOLOv10, and does Faster R-CNN still maintain advantage in accuracy in comparison?

1.3 Aim and Objectives

The main aim of this research project is to analyze and compare various state-of-the-art models for object detection in the field of vehicles, especially in low-light environments. The analysis consists of evaluating their detection accuracy as well as real-world application. With such analysis, the research aims to offer insight into the current state of deep-learning algorithms and future direction with deep-learning and transformers powered developments.

To fulfill this aim, the following objectives were achieved during this project.

1. To collect and prepare a comprehensive dataset consisting of real-world low-light vehicle images.
2. To implement and fine-tune YOLOv10, YOLOv8, and Faster R-CNN models using the primary dataset initially and later curated the existing secondary dataset.
3. Evaluate the performance of the model in low-light conditions with the help of metrics such as mean average precision and inference time.

4.Train the models on larger secondary dataset sources to evaluate for further accuracy and speed of evaluation.

5. Present the findings of the training and compare them to answer research questions.

1.4 Methodology

This research adopts a mixed-method approach, incorporating both primary and secondary data. Initially, the primary research involves the collection of custom low-light vehicle images using a semi-controlled method, which serves as the initial training and testing dataset. This approach to collecting dataset helps to ensure the models are evaluated on realistic and relevant low-light scenarios (Watkins, 2023).

To enhance the scale and diversity of the data, secondary datasets—such as publicly available low-light and vehicle detection datasets—are integrated. This combination helps improve the robustness and generalization of the results. The models are trained using Python as the programming language and PyTorch, a popular deep-learning framework. Model performance is evaluated using standard metrics including mAP, precision, and inference time.

1.5 Synopsis

The project from this stage is broken down into four main chapters. First is Literature Review 1, which delves into the broader discussions regarding the progress in the field of computer vision and specifically object detection. It covers the historical methods of object detection and development of convolution neural network techniques from 2014 and the subsequent methods that followed it such as a single shot method YOLO and lately transformers architecture.

The next chapter is Literature Review 2 which goes deeper into object detection in the field of vehicle detection and applications specific to it such as autonomous driving and traffic management. The chapter also discusses the technologies specific to vehicle detection such as sensors-based approach and camera-based approach to data collection and evaluates both of their challenges and use-cases. The section presents various studies conducted in the field of vehicle detection and comparisons of various methodologies to identify gaps in the field one of which is the lack of comparison in vehicle detection using the newer YOLOv10.

The chapter to follow is the research methodology which describes the type of research along with the data collection techniques applied, the nature of the data and potential biases along with the limitations and challenges while collecting the data. After that Findings, Analysis and

Discussion section presents the full life cycle of the technical implementation in deep details and presents the findings of the respective technique and the final outcome.

Literature Review 1

2.1 Introduction

With the rapid growth in deep learning and computer vision, object detection has also been of equal importance. Its application ranges from as regular as facial recognition to agriculture pest monitoring, medical imaging, traffic and vehicle management, surveillance, etc. (Boesch, 2024). Facial recognition is very important in the modern context, either in a smartphone or offices as a security measure or as a surveillance device in high-risk environments (Ghenescu et al., 2018; Alamri, 2025). Similarly, object detection technologies have been used in the field of agriculture to monitor pest, the quality of crops, etc. Likewise, in medical fields, various models have been trained to identify tumors and other abnormalities in imaging to treat life threatening diseases before they become visible (Li et al., 2019). A significant challenge for its application is object detection under low lights and because of the issues such as high noise, bad contrast, and illumination, etc., consistently detecting accurate objects from images still comes with huge challenges (Shovo et al., 2024).

2.2 Object detection methods

Deep learning-based object detection methods can be classified into two main categories: CNN-based models and transformers-based models. The CNN based models can itself be separated into two types: one-stage and two-stage detector (Sun et al., 2024). The two-stage models first generates region of interest (RoI) to create candidate bounding boxes for the objects and in the next step, performs feature extraction to detect objects. R-CNN was one of the first such two stage models, presented by scholars from UC Berkley in 2013 in their paper titled ‘Rich feature hierarchies for accurate object detection and semantic segmentation’. They classified the problem into three different modules. First, they used a selective search method to identify region of interest in the given image. Secondly, they passed the RoIs through a convolution layer to extract a feature vector for each of the regions. Finally, the features are used to classify each region with a bounding box for each class. (Girshick, et al., 2014)

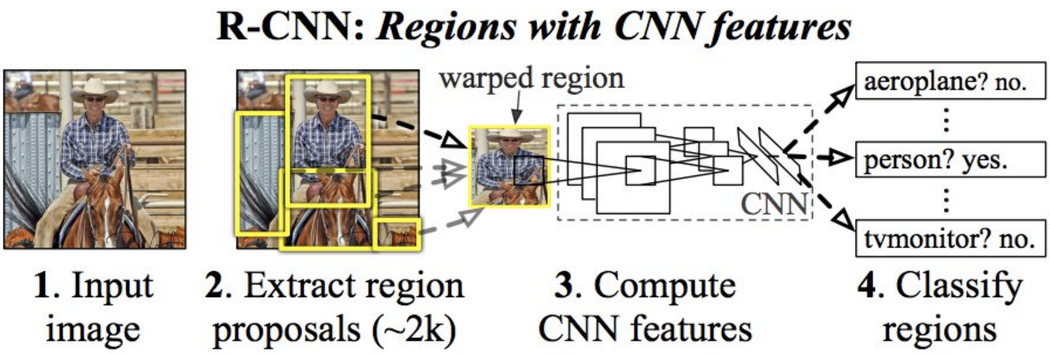


Figure 1: R-CNN architecture (Girshick, et al., 2014)

This architecture has been further improved with Fast R-CNN and Faster R-CNN which have modified the original and improved both the performance and speed of the models. Fast R-CNN improved on the computational bottleneck by passing the entire image through convolution instead of independent region propagation that is performed in its predecessor. Faster R-CNN further improves the architecture by replacing the selective search, used to identify the region of interests, with a region proposal network. (Ren, et al., 2016)

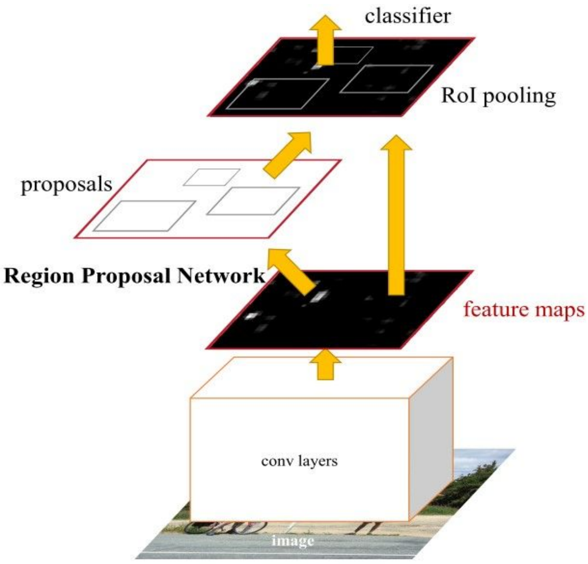


Figure 2: Faster R-CNN architecture (Ren, et al., 2016)

One-stage methods on the other hand do not use the RoI mechanism to generate bounding boxes and instead combine both the steps from above into a single stage. YOLO (You Only Look Once), Single Shot Detector (SSD), RetinaNet, etc. are some of the popular one-stage models used today. While not as accurate as two-stage detectors, because of the speed of inference, one-stage detectors such as the family of YOLO are very popular amongst the applications that prioritize speed and real-time analysis. YOLO works by first dividing an image into a grid of equal shape. Then convolution is applied to extract features in each of the grids. Intersection over unions is then applied over all the grids to identify the image candidates and those that are relevant over a specific threshold that is predetermined. Finally, non-max suppression is applied to remove the noisy bounding boxes with lower confidence scores and keep only the higher scores. This first version of YOLO was a huge improvement in terms of speed and computation but came with its challenges such as struggling to detect small or multi object images. Over the years, new versions have gradually improved upon the defects and by the time of writing this research in March 2025, there is a YOLOv10 that will be used in this paper. (Redmon, et al., 2016)

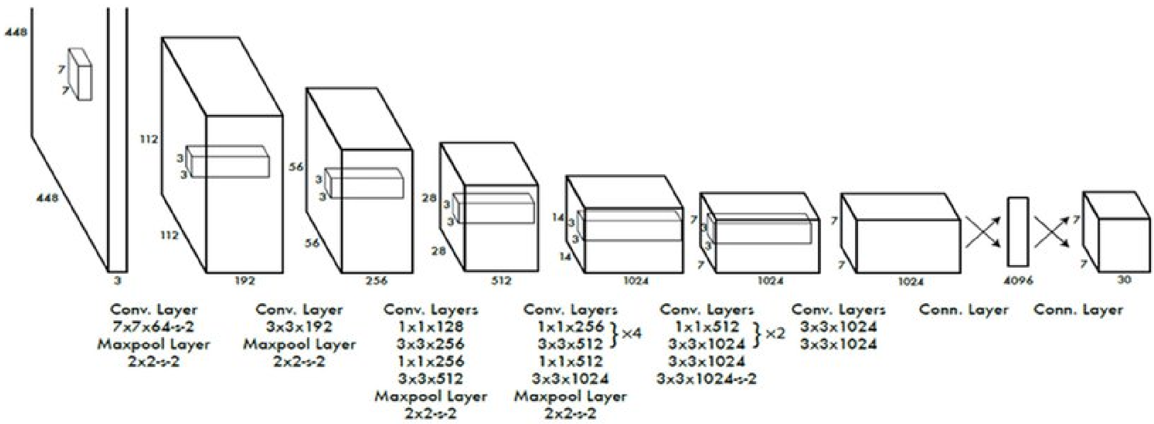


Figure 3: First YOLO architecture (Redmon, et al., 2016)

Transformers based detectors on the other hand is the latest architecture, compared to two mentioned mentioned above and entered object detection in 2020. After its success with amazing performance in the field of natural language processing, researchers started to implement various ways in which to take advantage of the attention architecture and introduced it in object detection (Shehzadi, et al., 2023). This architecture improves on the CNN based approach which spends time on post-processing step, to implement non-maximum suppression (NMS) to filter out the unnecessary bounding-boxes. So, the transformer structure: CNN backbone, encoder, decoder and prediction, removes NHS in post-processing (Sun et al., 2024). DETR, Vision transformer, Convolutional Vision Transformer, etc. are some of the most used models in object detection.

2.3 Object detection in low visibility environment

Most of the research done and progress made in various fields of computer vision have been on images of good quality and taken in good lighting (Dhabliya, 2023). However, practically, that is not usually the case. The visibility in images can be affected by many environmental factors surrounding it, such as bad lighting, moving objects, time of the day, etc. This can result in the images having high noise, bad contrast and blurred objects which all contribute to lower performance during object detection (Ashar, et al., 2024).

The main approach for object detection in low light images consists of two stages: image enhancement followed by object detection. The enhancement phase works as a preprocessing step to improve the visibility of the objects in the image before being passed into the object detection phase (Al-Refai, et al., 2025). Images can be enhanced in either classical methods or more recently, methods based on deep learning. Classical methods include gamma correction, histogram equalization, and Retinex whereas deep learning approaches such as Cycle-GAN focus on image-to-image translation (Wu, et al., 2023). Because of its better performance, image enhancement using deep learning for object detection is preferred in recent times. Researchers have experimented with traditional approach and deep learning approach and have found mixed results with both approaches even doing more harm than good in some cases.

One such research was done by Chen and Shah (2021) who compared two different enhancement techniques: brightness histogram normalization and Cycle-GAN, an advanced convolution-based method. After training the resulting images on Faster-RCNN model, they found that the basic histogram normalization performed worse than the original dataset whereas the Cycle-GAN performed better.

Another study used Contrast Limited Adaptive Histogram Equalization (CLAHE) before using R-CNN, YOLOv5 and SSD algorithms to train models for object detection. They found that CLAHE based enhancements outperformed the normal histogram equalization and harmed the accuracy of detection and concluded that the time taken by RCNN model is a tradeoff and recommend YOLOv5 with ResNet-50 as backbone for low light object detection. (Sivasubramanian, et al., 2023)

Similarly, Al Sobbahia and Teklia (2022) in their paper presented a comprehensive comparison in which they used 10 different deep learning-based enhancement models to improve images followed by 4 different object detection models: YOLOv3, Mask R-CNN, RetinaNet and SSD. They observed that most of the enhancement models behaved similarly in terms of object detection and that visually improved images did not always correlate with improved object detection. They also observed that except for YOLOv3, most models performed better on the original low light images instead of the enhanced ones. More specifically they found that the YOLO model saw improvement when combined with enhancement models MBLLEN and DeepUPE on special classes of the dataset. This finding suggests that image enhancement models could be helpful for object detection if used in a way that preserves its features without distortion.

Irrespective of the image enhancement technique, researchers and scholars have used the above-mentioned state-of-the-art object detection methods in various fields to make progress in computer vision. There has been plenty of research and experiments done in a variety of ways for each model. Whether a study with a single model, modifying the architecture of a model by either adding or removing parts to achieve the desired results, or comparing the models and their different version with each other and evaluating their performance by using their statistical scores in the same dataset. Some of the research conducted on the various object detection techniques are described below.

Diwan, et al. (2022) used multiple versions of YOLO, up-to version 5 in their paper to train a dataset and compared it with two stage detectors such as Fast-RCNN. They found that using YOLO to identify the background and Fast-RCNN to detect objects produced results that outperformed the results as opposed to when each method individually. However, they did not present any findings in their paper to quantify and did not mention any detail about the low light environment. Also, because the YOLO method has many new versions now after their research, it is fair to assume that the performance of the newer versions would be comparatively better.

More recently, Bučko, et al. (2022) conducted research in which they used YOLOv3 and Sparse R-CNN to train a model using a dataset consisting of diverse weather conditions images, to detect potholes on roads. They observed that the YOLO model showed a high success rate with low computing time and hardware. However, they concluded the Sparse R-CNN model performed better in images with low light. Frnda, et al. (2024) extended the research by training their models on more recent YOLOv7 and Faster R-CNN to detect potholes on roads. They managed to improve the accuracy of Bucko et.al (2022) by balancing the dataset, increasing the less prevalent classes in the dataset synthetically using GAN technique. As a result, the Faster R-CNN model showed significant accuracy with mean Average Precision (mAP@50) for 'rainy' class at 0.82 and 'evening' at 0.89, which was significantly better than previous studies.

Abbas, et al. (2023) developed a traffic light management system using Faster R-CNN. They created an algorithm that classified the detected images, trained on self-generated local dataset, and classified them and based on their density, assigning the traffic light value with coordination with the central traffic servers. They achieved 95.7% accuracy which, compared to other similar projects, using different algorithms, is a better score. However, most of the literature I have studied thus far has suggested that two-stage methods like Faster R-CNN are not ideal for real-time environments such as traffic light management due to the slower nature of inference and computational overhead. Thus, the speed of detection could be improved with the use of a single-stage method that is not explored by the researchers.

The state-of-the-art algorithms and models have been used as a base model and modified by either removing or adding parts of architecture to achieve better results in the respective research fields. Han, et al. (2024) proposed a new object detection method named 3L-YOLO, based on the existing YOLOv8n by removing the need for image enhancing modules that are often required before detecting any objects. By testing on multiple benchmarks dataset such as ExDark, ExDark+ and DarkFace their mean average precision @50 improved by 2.7%, 4.3% and 1.4% respectively.

Ding, et al. (2024) proposed a lightweight YOLOv8n model for low-light underwater target detection in which they replaced the traditional convolution layers in backbone and neck networks with Ghost and GSconv modules. Training on RUOD dataset, the lightweight approach produced mAP@50 of 86.1% and mAP@50:95 of 60.8%, which are higher than that of original YOLOv8n algorithm which had 79.6% and 58.2% respectively.

Li, et al. (2022) proposed a novel lightweight object detector based on Transformers, L-DETR, by modifying the original DETR in which they reduce the parameters by 26 and 46 percent for backbone of ResNet50 and ResNet18 respectively. They present improved findings on lower computational overhead suggesting a better size to be deployed on the cloud. More interestingly, they also suggest that different activation functions significantly affect different normalization methods, and it is important to experiment with to achieve better results when both are being used.

Another research focused on vehicle-type recognition methods using an improved Faster R-CNN model. The researchers improved the model by taking features from each convolution layer as opposed to the original Faster R-CNN which only takes the features from the last layer. As a result, they achieved a mAP that was 1.7% higher than the traditional Faster R-CNN. However, due to the addition of new feature layers as well as using all the layers, the computational and recognition time increased. Hence, it can be argued that such small improvements in accuracy is not a good enough trade-off for increased computational overhead. (Bai, et al., 2024)

Reddy et al. (2024) comparatively evaluated different convolution neural network-based object detection models, like what this research wants to accomplish. They compared Faster R-CNN, YOLOv3 and SSD (Single Shot MultiBox Detector) in the context of vehicle detection. They found that SSD outperformed both the other methods in terms of speed and and mean average precision and also found that all three achieved accuracies of over 99%. One reason for that could be the size of dataset they used, which only consisted of just over 600 images and presence of similar data. This meant that the model could have over-fitted during training and may not reproduce such accuracy if the number of images were increased significantly. Secondly, the research does not mention the nature of the environment in images i.e. low-light, clear weather, rainy, etc. This could mean the dataset did not have enough diversity for the models to converge properly, resulting in such high accuracy.

Moussaoui et al. (2024) enhanced automated vehicle identification by using YOLOv8 along with OCR (Optical Character Recognition) to detect and identify license plates accurately. They used the YOLOv8 to extract license plate from the original image and used OCR to identify its origin. While the accuracy in OCR is beyond the scope of this project, the detection model produced 99% accuracy. As with the research by Reddy et al. (2024) from above, the small nature of the dataset along with lack of diversity of the environments in which the images were taken are likely the factors for such high precision.

Zhao et al. (2023) in their paper 'DETRs beats YOLOs on Real-time object detection', propose a transformer-based solution to take advantage of Transformer's ability to not require NMS while maintaining the high computational costs that come with it in the form of RT-DETR. This novel approach improves on the speed and performance that lacks with the original DETR but still has flaws in detecting small objects and the researchers do not mention about the performance on data with low lights.

Geetha et al. (2024) performed a comparative analysis of the more recent YOLO models: YOLOv8 and YOLOv10 in vehicle detection. They used a publicly available dataset consisting of 1321 images with five different classes: car motorcycle, truck, bus, and bicycle. They found that the new v10 model performed better in detecting bicycles and trucks, compared to v8, which was better at detecting cars. They conclude that the YOLOv10 is more versatile, and the performance was equivalent, and the choice depended on the computational resource and dataset.

Based on the literature, researchers have been using the state-of-the-art algorithms for object detection such as YOLO and Faster R-CNN in various fields to detect objects in low visibility. It can be observed that many detections are dependent on the field of application and there is no single perfect way to approach the problem at hand. Based on the data, application of correct enhancement before detection training with specific training algorithms produces better results.

2.4 Literature Gaps

This dissertation will use the Geetha et. al (2024) YOLO comparison paper from above as a baseline and extend to fill the gaps. While they have done a detailed analysis of YOLO v8 and v10, they mention the lack of diversity in their dataset such as bad weather and low lights environment as potential gaps in the project, along with the examination of computational efficiency. While the computational efficiency comparison will be outside the scope of this project, the gap with low light and different weather scenarios will be explored. By extending the research to include images with different visibility labels to a dataset, a more detailed analysis and observation could be performed based on the results of the different models.

More specifically, during this research, the results of YOLO v8 will be compared with those of YOLO v10 for a dataset of vehicle images, taken at multiple lighting settings. Additionally, the same dataset will also be used to train on a two-stage model, Faster R-CNN, to compare the performance compared with the YOLOv10 model.

2.5 Summary

The recent growth in artificial intelligence has had a significant impact in deep learning and computer vision. This part of the document first discussed the fields in which computer vision has been of use and secondly, the challenges of application regarding working with images was discussed. Then the research highlighted the different architectures and methodologies used to detect objects. Next, the past work on object detection in low light environment was explored and a gap was identified in vehicle detection in low-light environment using state-of-the-art YOLOv10 model and its comparison with two stage detectors like Faster R-CNN. Thus, it was decided this project would further the existing comparison research by including a diverse dataset.

Literature Review 2

3.1 Introduction

Vehicle detection carries a lot of application in modern transportation management. However, there remain factors that compromises correct detection in nighttime, tunnels, or poorly lit urban areas like reduced visibility, sensor noise, and dynamic lighting variations (Loh & Chan, 2019). Despite advances in deep learning, particularly with models like YOLOv10 (Wang et al., 2024) and Faster R-CNN (Ren et al., 2015), the comparative performance of these models in low-light environments has yet to be thoroughly explored.

This section of the literature review explores the importance of vehicle detection, evaluates existing technologies, and identifies the challenges unique to low-light environments. By presenting the development in recent studies, it highlights the need for direct bench-marking of YOLOv10 and Faster R-CNN in real-world low-light scenarios.

3.2 Importance of vehicle detection

Vehicle detection systems are fundamental to several high-stakes applications in modern society.

- **Autonomous Vehicles**: Self-driving cars rely on real-time vehicle detection for safe navigation. Any kind of failure may result in catastrophic accidents. There have been studies conducted in the past of Tesla autopilot system struggling with glare from headlights at night in the past (Premaratne, et al., 2023).

- **Traffic Management**: Urban traffic systems use detection algorithms to monitor congestion and optimize signal timings. Inadequate detection during nighttime can skew traffic analytics, leading to inefficient routing (Anandhalli et al., 2022).

- **Law Enforcement**: Automated number plate recognition (ANPR) systems depend on vehicle detection to identify potential suspects to crime and that can be hampered by poor performance at night, reducing the reliability of crime prevention systems (Tang et al., 2022).

- **Accident Prevention**: Over 40% of fatal road accidents occur at night, despite lower traffic volumes, highlighting the critical need for robust low-light detection systems according to a study by the World Health Organization (Stephensons, 2019).

3.3 Existing Technologies for Vehicle Detection

Just like the other fields of computer vision, vehicle detection methodologies have also progressed from traditional image processing techniques such as histogram equalization, Retinex theory and HDR fusion to sensor-based approach and then deep learning (Wang et al., 2022). Thermal cameras work by detecting heat signatures and as a result perform well in darkness, but struggled in reflective surfaces, although there are not many vehicle-specific applications to it apart from pedestrian detection (Waters, 2024).

LiDAR on the other hand provides precise three-dimensional data that can be used to calculate all the surroundings at a rapid time and is highly accurate. Hence, there are many advocates for it to be used as the main source of automation, However, the cost for large-scale deployment is higher due to the price and production of each individual device (Chen et al., 2021).

In recent years, deep learning models have significantly advanced vehicle detection, leveraging powerful architectures like convolution neural networks (CNNs) to improve performance in both speed and accuracy (Alif, 2024). These models have largely replaced traditional techniques in many real-time applications due to their reliability in different types of environments. Among these, the YOLO series has gained widespread adoption as a highly efficient, single-stage detector. YOLOv10, released in 2024, introduces "Consistent Dual Assignments" to enhance detection speed while maintaining accuracy, making it particularly suitable for real-time applications in low-light conditions (Wang et al., 2024). Earlier iterations of YOLO, like YOLOv4, demonstrated impressive performance, achieving a 95% mean average precision (mAP) on daytime vehicle detection tasks (Bochkovskiy et al., 2020). The strength of YOLO lies in its ability to predict bounding boxes and class probabilities in a single pass, thus minimizing inference time, which is crucial for real-time vehicle detection in dynamic environments (Buhl, 2024).

On the other hand, Faster R-CNN, a two-stage detector, offers higher precision due to its use of Region Proposal Networks (RPNs), which propose potential regions of interest before the detection process. This method enables the model to focus more accurately on the most relevant parts of an image, leading to better detection results, especially in cluttered or occluded scenes. Faster R-CNN achieves a high mAP of 98% on the COCO dataset, showcasing its capability for precise object localization (Ren et al., 2015). However, this precision comes at the cost of slower processing speeds compared to single-stage detectors like YOLO, which is a key consideration for real-time vehicle detection in autonomous systems or traffic management. While Faster R-CNN's two-stage nature provides an advantage in terms of detection accuracy, its slower inference time makes it less suitable for environments that require rapid decision-making, such as autonomous driving or real-time traffic monitoring.

In the context of low-light environments, deep learning models like YOLOv10 and Faster R-CNN have been adapted and enhanced with techniques such as low-light image enhancement. These enhancements, often involving the use of synthetic low-light data or specialized training methods like domain adaptation, aim to improve the robustness of the models in challenging lighting conditions. For instance, methods such as the use of CycleGANs to generate low-light images help train Faster R-CNN to perform better under real-world nighttime conditions (Shovo et al., 2024). Despite these advancements, challenges remain in balancing real-time performance with high detection accuracy, especially when considering the impact of extreme lighting conditions on model effectiveness.

3.4 Challenges in Vehicle Detection

Detecting vehicles in low-light environments presents several unique challenges that significantly impact the performance and reliability of detection systems. One of the primary issues is poor illumination, which directly affects the quality of the images captured by cameras. Low-light conditions, such as those encountered at nighttime, drastically reduce the signal-to-noise ratio (SNR), leading to grainy or blurred images. For example, under moonlight, the available illumination is 0.1 lux, compared to 10,000 lux in broad daylight (Loh & Chan, 2019). This difference results in sensor difficulties, especially in cameras, which suffer from photon starvation and produce noise that hinders vehicle detection. The presence of low contrast in the images further complicates the task, making it difficult for models to distinguish vehicles from the background.

In addition to poor illumination, dynamic lighting conditions pose a significant challenge. Glare from headlights or streetlights can overwhelm camera sensors, creating regions of high and low brightness that obscure details in the image, leading to false positives or missed detection. This glare effect is more problematic in nighttime vehicle detection, as the bright spots in the image may be incorrectly classified as vehicles or region/object of interest for the model. There have even been studies that show that reflections on wet roads or surfaces further highlights this issue, often causing misclassifications where the reflections are mistaken for actual vehicles (Tahir et al., 2024). These dynamic lighting variations highlight the challenges and importance in development of detection models capable of differentiating real vehicles from such false signals.

Another challenge faced in low-light vehicle detection is occlusion and the detection of small objects. In real-life environments and particularly in cities, vehicles are often partially obstructed by other cars, people, or general street objects, making it difficult for detection models to accurately localize them (He et al., 2024). This is also more problematic in nighttime scenarios where limited lighting makes occlusions harder to detect. Faster R-CNN, with its Region Proposal Network (RPN), partially addresses this issue by proposing potential regions of interest where vehicles might be located, but heavily obscured targets or overlapping vehicles still pose significant difficulties (Ren et al., 2015). Furthermore, smaller vehicles like motorcycles, which occupy a smaller area in an image, are often missed due to the low resolution of low-light images, that further complicates detection (Wang et al., 2023).

Real-time processing is another critical challenge. Autonomous vehicles and intelligent transportation systems must have inference times as close to real time as possible to make decisions that are useful and trustworthy. YOLOv10, with its fast processing speeds, can achieve 50 frames per second (FPS) on GPUs, making it suitable for real-time applications (Wang et al., 2024). However, Faster R-CNN operates at a slower pace, achieving only 7 FPS, which makes it unsuitable for real-time applications that require quick decisions based on continuous input from sensors (Ren et al., 2015). This speed difference highlights the need for further optimization in low-light detection models to meet the stringent requirements of real-time systems.

In addition to these technical challenges, there is also the issue of dataset limitations. Many of the available low-light datasets that have been used in most of the research studied for this project, such as ExDark (Loh & Chan, 2019), suffer from a lack of diversity in both vehicle types and lighting conditions. This narrow focus makes it difficult for models trained on such datasets to generalize well to real-world low-light scenarios. Furthermore, the reliance on synthetic data for training, such as artificially darkened images from the COCO dataset, leads to poor generalization in real-world environments where lighting conditions are far more complex (Chen et al., 2023). As a result, there is a need for more diverse, real-world datasets that accurately represent the variety of low-light conditions encountered in different environments.

Lastly, the computational costs associated with vehicle detection models in low-light environments can be also a factor for real world scenarios (Liang, et al., 2024). While this project will be performed on low quantity images on Kaggle's free resources, real time deployment of Faster R-CNN, with its two-stage architecture, requires higher computational resources, particularly GPU memory, making it challenging to deploy on edge devices like drones or traffic cameras that have limited processing power. This high computational cost is a significant barrier to the widespread adoption of deep learning models in practical, real-world applications, where resource constraints pose significant challenges and must be considered alongside performance requirements.

3.5 Literature Gaps

The recent advancements in deep learning have improved detection accuracy under normal lighting, research addressing low-light environments, especially for critical applications, is still limited. First, although several studies have evaluated individual models like YOLOv8 and Faster R-CNN for vehicle detection, comparative analyses focusing explicitly on low-light conditions remain relatively on the lower side. The recent release of YOLO models (at the time of writing this report there already are YOLOv11 and YOLOv12 model), has further widened this gap as there are very few existing comprehensive evaluations of these models in normal conditions, let alone in challenging environments. Thus, there is a lack of consolidated benchmarks assessing detection accuracy, precision, and recall among these models specifically for low-light vehicle detection scenarios.

The increasing use of synthetic data to augment training datasets has shown promise in improving model robustness. However, few studies have systematically compared model performance when trained on synthetic low-light data versus real-world low-light datasets (Wang et al., 2024). This gap is critical, as reliance on synthetic data might introduce issues impacting real-world application performance. These identified gaps underline the need for comprehensive, comparative studies that not only benchmark model performance but also contextualize findings within real-world low-light applications.

3.6 Summary

Vehicle detection in low-light environments remains a crucial yet challenging task for safety-critical systems such as autonomous driving and intelligent surveillance. Existing object detection frameworks, notably YOLOv10 and Faster R-CNN, offer contrasting strengths—real-time inference speed versus high detection precision. However, their comparative analysis under real-world low-light conditions is not explored to its potential, with much of the literature relying on controlled datasets that fail to capture the complexity of night-time urban environments, dynamic lighting variations, and sensor noise.

Reviews of prior work, including Geetha et al. (2024), reveal an underlying assumption that model performance in standard benchmark datasets generalizes well to conditions with bad lighting as well. This assumption overlooks significant methodological gaps, such as the limited diversity and scale of low-light datasets, and the lack of rigorous cross-domain validation. Geetha et al.'s comparative analysis of YOLOv8 and YOLOv10, while informative, is limited by the dataset size of just around 1300 images and a focus on static object categories, which may not fully represent real-world traffic scenarios characterized by occlusions, motion blur, and different lighting sources.

Furthermore, the evaluation metrics mAP, do not adequately reflect the practical demands of real-time safety applications, where detection latency and robustness to environmental noise are equally critical. This chapter seeks to bridge these gaps by emphasizing the need for comprehensive performance benchmarks that balance accuracy, computational efficiency, and environmental adaptability.

In analyzing the findings, it becomes evident that no single architecture offers a universally acceptable solution. Instead, hybrid approaches that combine multi-sensor data with advanced CNN architectures hold promise for overcoming the limitations of vision-only detection systems. Future research directions should thus prioritize the creation of large-scale, diverse low-light datasets, the development of lightweight yet robust detection models, and the integration of cross-modal sensor inputs (e.g., LiDAR, thermal imaging) to enhance detection reliability in complex night-time scenarios. In conclusion, while this chapter establishes a solid technical foundation, it underscores the necessity for more context-aware evaluations of vehicle detection models.

Research Methodology

4.1 Research Design

The research designs this project adopts is descriptive research to comparatively evaluate the performance of YOLOv10 and Faster R-CNN models for vehicle detection in low-light environments. A descriptive approach is a type of design that aims to observe, describe and analyze the existing findings; hence it is appropriate to be used in this project that aims to analyze the behavior and accuracy of both models under predefined conditions with the same dataset (McCombes, 2019).

To ensure a structured comparison, the study will measure key performance metrics such as precision, recall, F1-score, mean average precision (mAP), and inference speed across the vehicular dataset consisting of low-light vehicle images. Additionally, the research will examine how factors like image noise, motion blur, and varying illumination levels influence detection reliability.

By analyzing and interpreting these results, the study aims to provide insights into the strengths and limitations of each model, particularly in challenging visibility conditions. The findings will not only contribute to academic discourse on deep learning-based object detection but also offer practical guidance for autonomous driving systems, traffic surveillance, and security applications where low-light performance is critical. Furthermore, the study's structured methodology ensures reproducibility, allowing future researchers to validate or extend the findings using alternative datasets or emerging detection models.

4.2 Research Methodology and Justification

The method of research for this project is quantitative research method, as it involves the systematic collection and analysis of numerical data to compare the performance of YOLOv10 and Faster R-CNN models. While visual observation plays a role in verifying detection accuracy which suggests some portion of qualitative research, the structured and numeric treatment of results aligns with quantitative research (Sreekumar, 2023). The research is grounded in objective measurement, using performance metrics such as precision, recall, F1-score, and inference time to evaluate each model's effectiveness in detecting vehicles under low-light conditions. Image data will be collected and curated specifically for this study, ensuring relevance and consistency. The quantitative approach enables a rigorous, data-driven comparison that highlights the relative strengths and weaknesses of the two models, providing a solid basis for conclusions regarding their suitability for real-world low-light vehicle detection tasks.

4.3 Research Instruments

The primary instrument used for data collection in this study is a smartphone camera. This choice is based on the practicality, accessibility, and quality offered by modern smartphone devices, which are equipped with high-resolution camera sensors capable of capturing clear images even in low-light conditions. Using a smartphone allows for flexible and efficient data collection across various urban and semi-urban environments where vehicles are commonly found at night or in dim lighting. Additionally, the portability of the device enables real-time data capture from diverse angles and distances, simulating realistic scenarios for vehicle detection tasks. The consistency and convenience of smartphone cameras make them a suitable tool for building a robust and representative dataset required for training and testing the model.

4.4 Sampling

This study adopts a purposive sampling strategy aimed at constructing a diverse and representative dataset of vehicle images under varying environmental and lighting conditions. Purposive sampling is a non-probability sampling technique that allows the researcher to intentionally select samples based on specific characteristics based on their relevancy to the study (Bullard, 2024). Vehicles will be photographed at random times and locations, yet sampling will be intentionally done in a way to ensure coverage across multiple contexts, including highways, urban streets, residential areas, and street-parked vehicles, under both daylight and low-light scenarios. This ensures that the dataset captures the critical variables influencing detection performance, such as vehicle type, orientation, distance, occlusion, and lighting variability.

To ensure a statistically reliable evaluation of the object detection model's accuracy, I am collecting a sample of 400 images. The sample size is calculated using the formula for estimating a proportion with 1.96 certainty factor for 95% desired certainty as shown below. The output is just above 384, which was rounded up to 400 for proper representa-tion.

```
 Certainty factor: 1.960
 Acceptable error: 5%

sample_size = 0.25 * (certainty_factor/acceptable_error) **2

            = 0.25 * (1.96/0.05) **2
            = 0.25 * (39.2)**2
            = 0.25 * 1536.64
            = 384.16
```

Figure 4: Statistical Sample Calculation

The proposed sample size of approximately 400 images is typical in small to medium-scale empirical studies in computer vision, while also aligning with precedent in literature. For instance, Geetha et al. (2024) conducted their comparative study using a dataset of 1321 images across five vehicle classes, yet their findings were limited by the limited nature of the dataset. Given the targeted scope of this study, focused on comparative analysis instead of general accuracy, a smaller but carefully curated sample is justified.

Studies such as Zhao et al. (2024) highlight that small datasets comprising 300–1000 images can yield effective results when leveraging transfer learning with pre-trained models. Additionally, research by Hestness et al. (2017) shows that model performance in deep learning tasks tends to follow a power-law scaling with dataset size, indicating diminishing returns beyond a certain sample threshold.

In the context of object detection, prior benchmarks and empirical studies suggest that key performance metrics stabilize beyond 150–200 well-annotated images per class, especially for comparative evaluations (Redmon et al., 2016; Han et al., 2024). Therefore, a dataset comprising 400 carefully curated and balanced images offer a statistically reasonable basis for achieving meaningful model performance while supporting comparative analyses across key scenarios.

Although the sampling approach does not aim for statistical randomness, it prioritizes diversity, ensuring that the evaluation of YOLOv10 and Faster R-CNN reflects realistic deployment environments. This choice balances practical feasibility with the scientific rigor needed to support the study's empirical objectives.

4.5 Data Collection

For this study, primary data collection serves as a foundational component, providing the essential dataset required to evaluate object detection model performance under real-world conditions. Data will be collected using smartphone cameras from various devices, specifically targeting vehicle images across diverse lighting conditions and environments. By capturing images from multiple locations, this approach ensures diversity in backgrounds, viewing angles, and vehicle types, which is critical for building a representative dataset. Furthermore, conducting primary data collection allows for direct control over image quality, resolution, and environmental factors, enabling a focused investigation into the challenges of low-light object detection. This level of control is essential for ensuring the dataset aligns with the study's objectives and supports robust model evaluation.

In addition to primary data, secondary data sources will be incorporated to train and test the models in a later phase to enhance the accuracy of the findings. This mixed data collection strategy will help with hypothesis testing with real-world samples and validation across diverse image sources. All primary data will be initially annotated manually with the help of annotation tools available online.

4.6 Data Annotation

Before moving on to use the data to move ahead with the technical portion of research, the data had to be annotated. For this purpose, Roboflow, an online tool, was used. It is a premier computer vision tool in the market and provides a smooth workflow to annotate images in bulk with little to no hassle.

For the primary section of the research, the following classes were chosen to annotate with: 2-wheeler, bus, car and heavy. All bicycles and motorcycles are labeled with 2-wheeler. Bus and car are self-explanatory. Heavy class includes vehicles that don't fall in the car or bus category such as trams, trucks, mini vans, etc. All of these were included in a group category instead of separating them because of the scope of the project, which is to detect vehicle but not necessarily the accuracy regarding the type of vehicle.

Along with the above-mentioned classes, an additional class, pedestrian, is used in the secondary dataset, which is prepared in anticipation that the model trained with just the primary dataset might not produce a good enough model. This is because most of the public dataset for vehicle detection come with pedestrians as well and I want to include them in the model to increase the difficulty of separating pedestrians from the vehicles as well as identifying the vehicles.

After the below-mentioned steps to prepare the dataset, both the primary and secondary dataset are exported in respective formats for the YOLO models and Faster R-CNN models.

4.6.1 Primary Data

The primary dataset, compiled during different times of day, throughout Berlin using different phone cameras were all manually annotated using Roboflow. All 400 images were loaded and annotated on the platform.

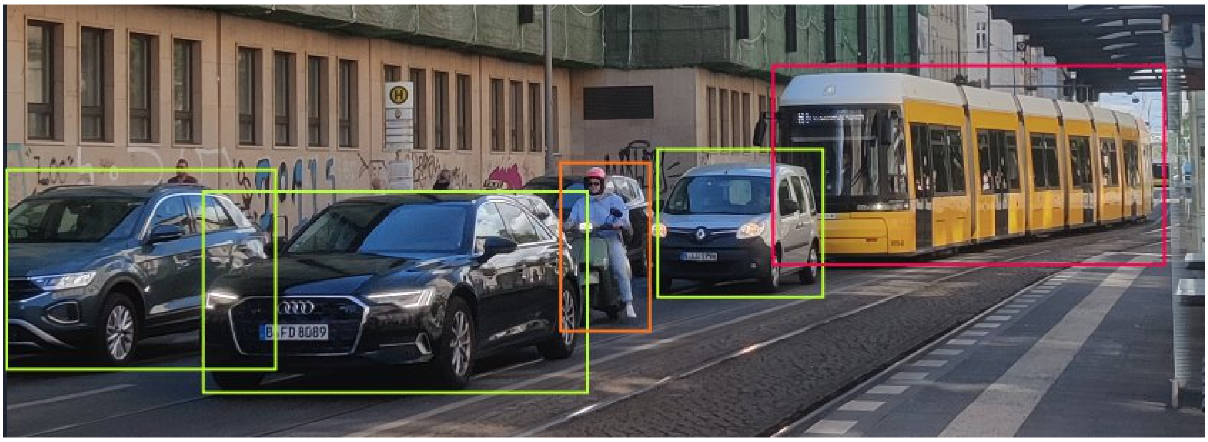


Figure 5: Sample of Primary Dataset

The annotation process involved the following steps:

- Drawing bounding boxes around each object of interest i.e. car, bike, trams, etc.

- Assigning the appropriate class label to each bounding box from the list of classes mentioned above.

4.6.2 Secondary Data

The secondary dataset was compiled using three existing datasets of various sizes. One data set had 901 images, all taken from a single place, where the environment looks like it was after a rainfall and the lighting is a bit dimmer than regular images. Another factor that made it ideal is the images have a lot of occlusions, which should test the capability of the different models. Another small dataset, just 200 images were added to the dataset from a source that shows images taken from inside a vehicle which mostly shows the back of vehicles in nighttime. This was chosen for its visibility and for different perspective images to include diversity in the dataset.

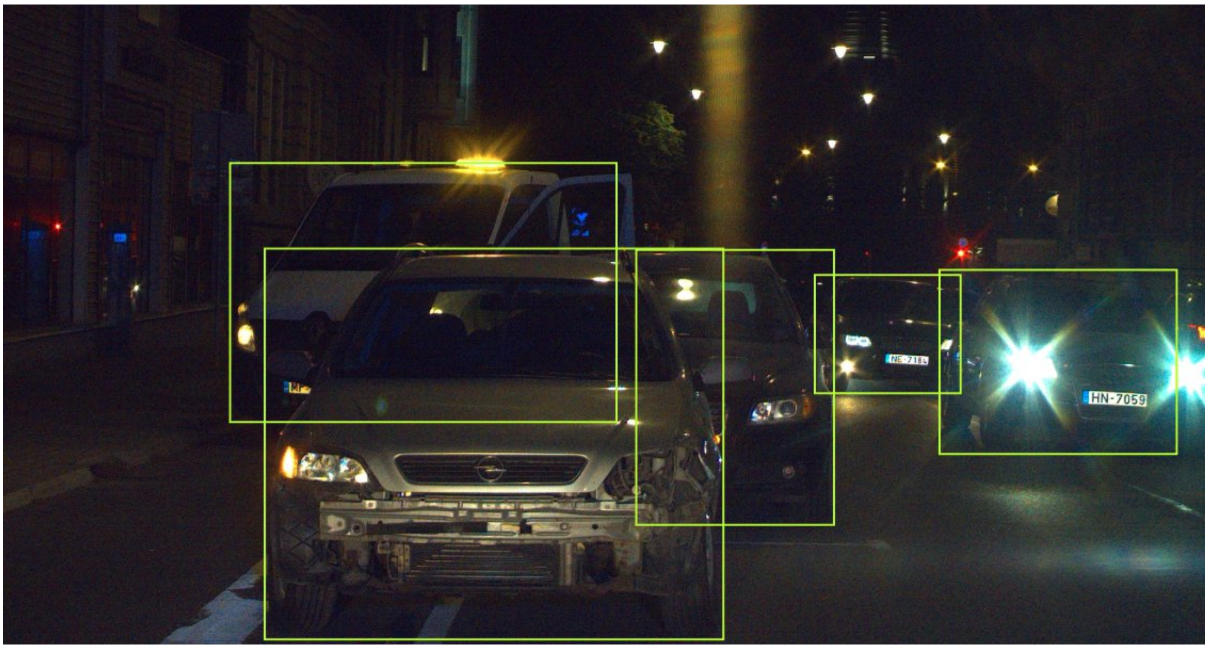


Figure 6: Sample of Secondary Dataset

These two relatively smaller datasets were then merged with a larger dataset, consisting of 5146 images, making it a comprehensive primary source for the secondary analysis portion of this research. The dataset mostly includes images of vehicles captured in low-light conditions, taken mostly from inside of another moving vehicle. As a result, many of the images contain vehicles with partial views, occlusions and complex backgrounds, all of which present significant challenges in object detection. All of these make the dataset well-suited to train models and evaluate their capabilities.

All three datasets came with distinct class definitions, with each following their own naming convention and scope. Hence, it was necessary to merge them into relevant classes to this research before further processing. Irrelevant classes to vehicle detection such as number plate related labels were discarded, and similar classes were merged (e.g., “car” vs. “auto”). All of the classes prior to their merging on specific dataset are presented in the table below.

Dataset	Images	Classes	Action Taken
detector_v2 (SmartScan, 2024)	5146	NaN, front, heavy_front, heavy_rear, moto_front, moto_rear, np, np_square, np_tractor, np_tractor_square, rear	- Removed number plate related classes NaN, np, np_square, np_tractor, np_tractor_square. - Merged heavy_front, heavy_rear into ‘heavy’ class. - Merged front and rear into ‘car’.
Vehicle_Detection (ProjectTraffic,	901	2-wheelers, auto, bus, car, null, pedes-	- Merged auto into the ‘car’ label.

2025)		trian, truck	- Removed null. - Merged truck into ‘heavy’.
G7_object_detection (supagonova, 2022)	200	Biker, car, truck, null	- Removed null. - Merged biker to ‘2-wheelers’ - Merged truck into ‘heavy’

After merging them into a single dataset, all the images were manually verified and classes were adjusted and added as required, before the dataset was finally ready.

4.7 Models Selection and Training Setup

For YOLO models training, Ultralytics SDK will be used. They provide the most actively developed implementation of the YOLO models and provide a streamlined pipeline that is easy to work with. The framework is optimized for high performance and comes out-of-the-box with automatic mixed precision, efficient data loading, and comprehensive visualizations of metrics by default. For Faster R-CNN models, PyTorch will be used. It’s a popular framework built on top of PyTorch and provides implementation of pre-trained models, making it an easy choice.

For training the models, a consistent computing environment will be used in Kaggle for reproducible outcome and fair comparison. The GPU that will be used is the NVIDIA Tesla T4 GPU with 16 GB VRAM.

4.8 Data Analysis

Once the dataset has been collected and annotated, the analysis will focus on evaluating and comparing the performance of three object detection models: Faster R-CNN, YOLOv8, and YOLOv10. The evaluation will be conducted using standard object detection metrics, including:

i. Mean Average Precision (mAP@0.5 and mAP@0.5-0.95): These metrics assess the overall detection accuracy by computing the average precision across multiple Intersection over Union (IoU) thresholds.
ii. Precision and Recall: Precision measures the proportion of correctly detected vehicles among all detection, while recall measures the proportion of actual vehicles correctly detected.

iii. F1-Score: The harmonic mean of precision and recall will be calculated to assess the balance between these two metrics.
iv. Inference Time (FPS / latency): The real-time performance of each model will be evaluated by measuring frames per second (FPS) or latency per image, which is critical for deployment in resource-constrained environments.

4.9 Research Limitations

There are few limitations to the research that might have an impact on the project that will be acknowledged in this section. First is the size of the dataset for the primary research phase, which will be relatively small for training deep learning model. As a result, the outcome could be potentially not conclusive enough to determine any conclusion. Secondly, the use of a smartphone camera as the primary tool for data collection could create situations where the images are not diverse enough to be taken as a realistic scenario at all. They could also introduce inconsistencies in image quality, operating system-based image manipulation and augmentations, which could impact the findings.

Additionally, the dataset may lack extensive diversity in vehicle types, models, and environments, which could affect model robustness. Another significant constraint is the limited computational resources, particularly the absence of high-performance GPU support for training the models. This may restrict the scale of model training and experimentation, potentially impacting performance optimization.

4.10 Ethical Considerations

This study will follow and maintain the ethical standards for research throughout the duration of the whole project. During data collection phase, all images will be captured in public spaces where there is no reasonable expectation of privacy, and no personally identifiable information such as license plates or faces, will be kept, or used in the dataset. The data collected will be used strictly only for academic and research purposes. There is no human subject involvement requiring formal consent, as the focus is solely on vehicle detection.

Additionally, there are no conflicts of interest related to affiliations, sponsorship, or model bias. All tools and models used are open-source resources that are publicly available. And finally, the findings from the research will be presented with utmost transparency to maintain upheld academic integrity.

Findings, Analysis, and Discussion

5.1 Findings

This chapter presents the results of the experimentation done in training all three models before providing further analysis in subsequent sections. The experimentation was done at three different levels. First was to use a primary dataset and establish the baseline of performance across all three models. Secondly, the dataset was augmented, and each model was retrained to compare the improvements. Finally, a secondary data set was used to further improve the results.

Figure 7: Sample of Images for Prediction Visualization

Each model was fine-tuned in a pre-trained model available in the respective training infrastructure, i.e. Ultralytics and PyTorch. For consistency across the multiple models, before the training phase, three images were handpicked to test the prediction with. All three images were taken during nighttime with variable levels of objects, colors, glare, and occlusion.

5.1.1 EDA of Primary Dataset

The primary dataset collected aimed to represent the vehicle distribution in a regular environmental context for this research. The exploratory analysis highlighted that the total of 400 images resulted in a total annotation of 1299 bounding boxes. As shown in the bar chart below, the class imbalance is clear to see with cars accounting for most of the labels, hugely engulfing the rest of the classes with 2-wheeler, with bus and heavy vehicles even lower representation.

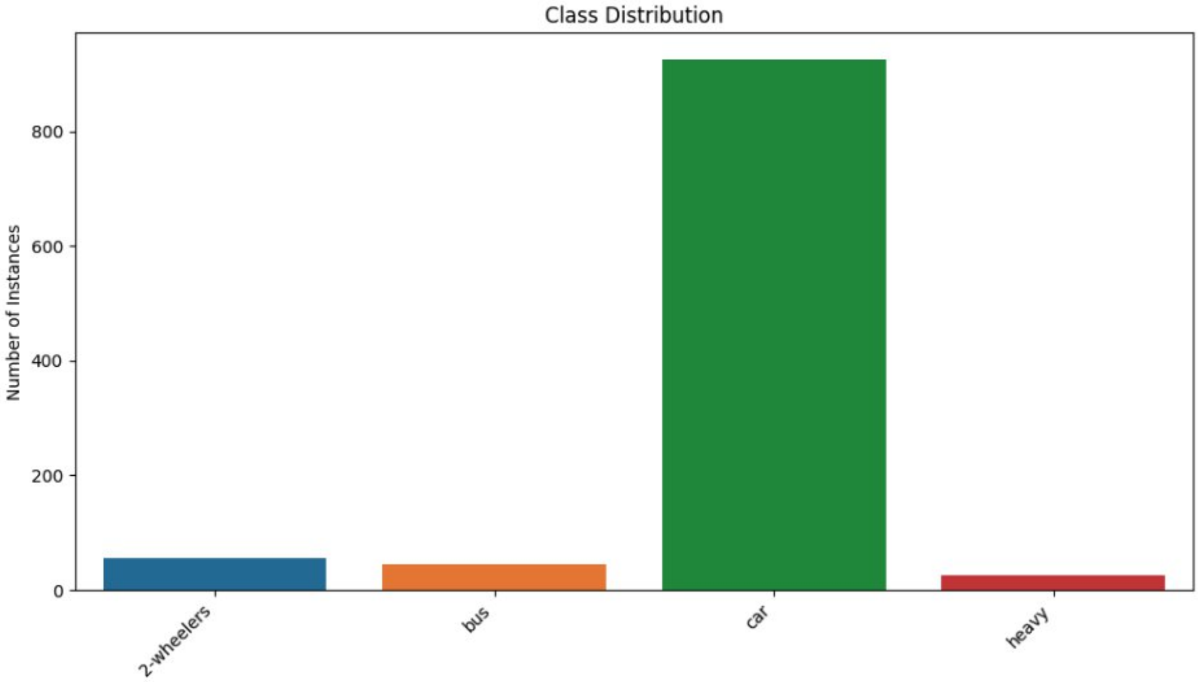


Figure 8: Primary Dataset - Class Distribution

This initial analysis highlighted limitations of the class, such as class imbalance, the car is much more prominent in the primary dataset than the other options, and lack of variability, which tend to pose significant challenges for models to generalize.

5.1.2 Baseline Models Performance

To establish a baseline for each of the models, all three were first trained on the 400 primary images with no augmentation applied to improve the existing model. The models were trained for 100 epochs and the metrics to evaluate the performances were calculated using 10% of the dedicated validation split.

5.1.2.1 Mean Average Precision

Classes		YOLOv8	YOLOv10	Faster R-CNN
Car	mAP@50	0.623	0.542	0.2946
	mAP@50-95	0.302	0.267	0.1768
2-wheelers	mAP@50	0.00702	0.0146	0.0000
	mAP@50-95	0.00224	0.00475	0.0000
bus	mAP@50	0.181	0.338	0.0839
	mAP@50-95	0.037	0.135	0.0503
heavy	mAP@50	0.199	0.0641	0.0000
	mAP@50-95	0.0693	0.0378	0.0000
Overall	mAP@50	0.252	0.239	0.0946
	mAP@50-95	0.103	0.111	0.0568

Table 1: Primary Models - mAP@50 & mAP@50-95 comparison

It can be observed that the overall mAP@50 and mAP@50-95 for all three models were relatively low, from 0.239 of YOLOv10 model to 0.252 of the YOLOv10 model and 0.0946 for the Faster RCNN model, indicating significant challenge in detecting vehicle by the models. These overall scores were heavily skewed by the score of cars – YOLOv8 had mAP@50 of 62.3 percent and even the low performing Faster R-CNN had 29 percent accuracy.
On the other hand, the detection performance of 2-wheelers across all three models was exceptionally poor and ranged from 0.007 to 0.00. YOLOv10 showed the best ability to detect buses at around 33 percent mAP@50.

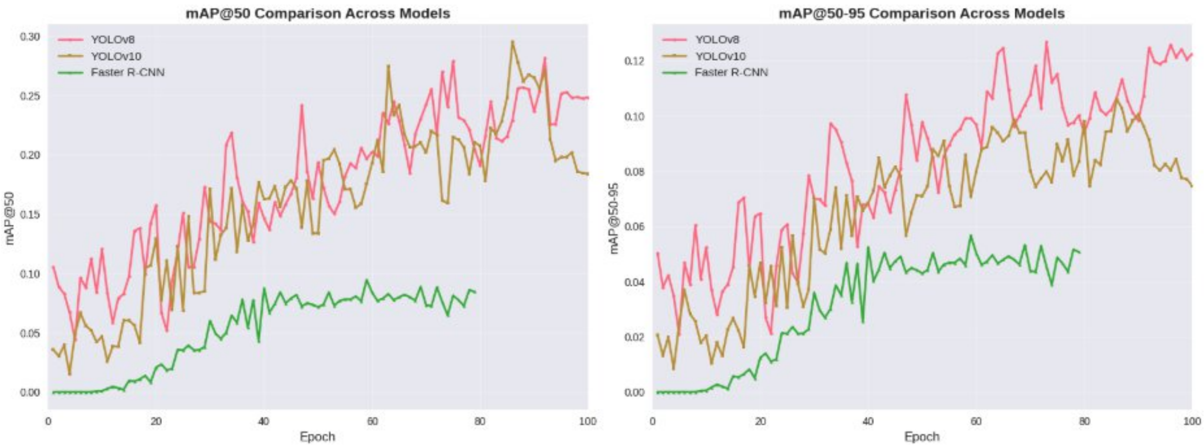


Figure 9: Base Models - mAP@50 and mAP@50-95 comparison

The table above can be further supported by the comparison of mAP over 100 epochs. The YOLOv8 model (pink line) consistently maintained a higher mAP@50, peaking at around 0.28 and settling around 0.25 at the end. YOLOv10 however demonstrated more volatility, starting lower than the v8 version but sometimes surpassed it, ultimately ending at around 0.20 mAP@50. The Faster R-CNN model on the other hand struggled to reach 100 epochs because it kept triggering the fail safe to end training if 15 epochs didn't show enough improvements. Both of its performance indicators never really came close to that of either of the other models.

5.1.2.2 Training Loss

The box loss, class loss, and dfl_loss components consistently decreased for all three models, indicating a continuous learning process, with YOLOv8's class loss showing a sharp initial decline and stabilizing after around 10 epochs, while YOLOv10's class loss also experienced a significant drop from a higher initial value. In contrast, the Faster R-CNN model exhibited a much more rapid convergence, with its box loss, class loss, and object loss dropping steeply within the first few epochs and stabilizing at very low values for the remainder of the train-ing.

Figure 10: Base Models - Loss Comparison

5.1.2.3 Precision and Recall Graph

YOLOv8 demonstrated highly fluctuating precision and recall values throughout its 100 epochs, with precision generally remaining higher than recall, though both show a slight up-ward trend towards the end. Similarly, YOLOv10 showed expected volatility in both precision and recall, particularly in the early stages, with precision often spiking and recall generally remaining at lower values, showing some increase in the later epochs. In contrast, Faster R-CNN shows a recall that fluctuates wildly, often reaching 1.0 in the early epochs before set-tling into a more consistent but still variable pattern, while its precision remains relatively low and stable, generally below 0.3, across the 80 epochs of training.

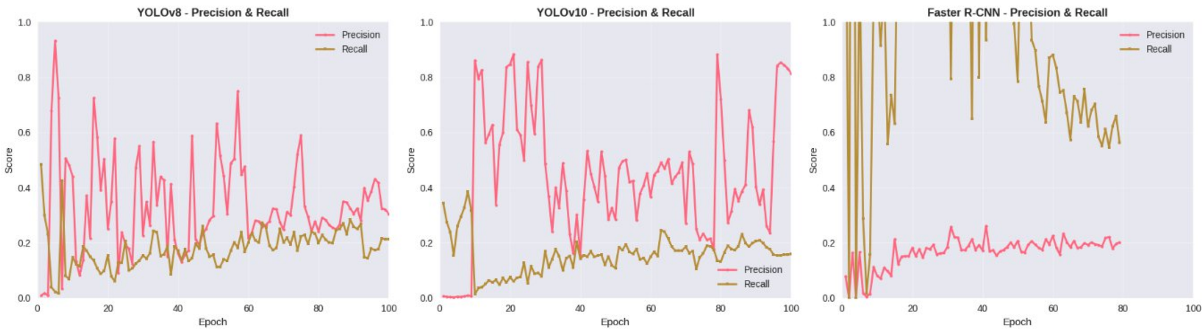


Figure 11: Base Models - Precision and Recall Curve

5.1.2.4 Baseline Models Summary

The training time for the base models showed a huge difference in training time between the two shot Faster R-CNN model and YOLO models. Overall summary showed that the mAP@50 and mAP@50-95 of YOLOv8 was the best along, whereas Faster R-CNN took the most time to train and produced the least satisfactory result.

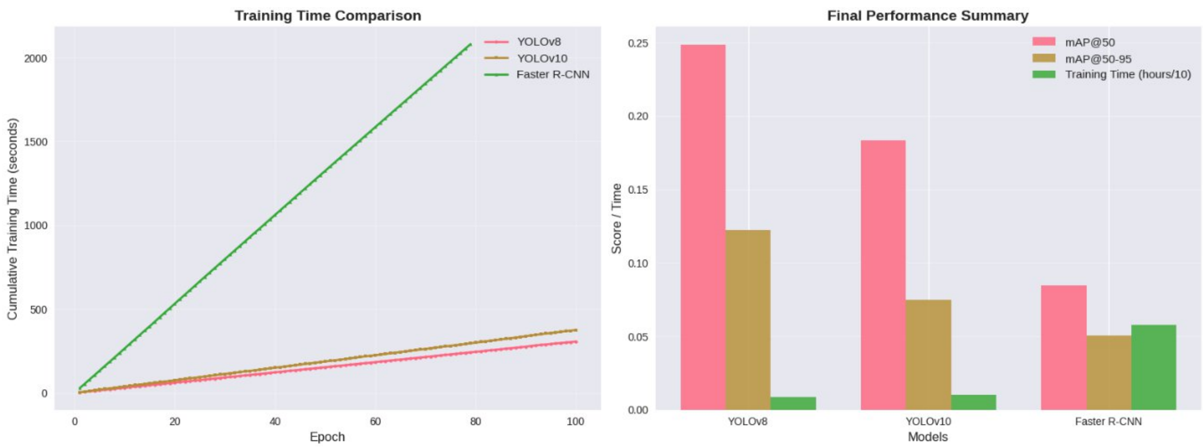


Figure 12: Base Models - Training Time and Summary

5.1.2.5 Inference

The figure below shows the prediction made by the base model of YOLOv8 on the images curated above. It can be observed that in the first image it fails to detect the tram and another instance of a car. In the middle and last image, it identifies all of the vehi-cles.

Figure 13: Base Models - YOLOv8 – Sample Prediction

Similarly, the figure below is the prediction of the base YOLOv10 model. Unlike the base v8 model, this model doesn't detect all the cars or the tram in the first image and also falsely detects heavy vehicle category in the second image. In the third one, it identifies cars but with a lower confidence level.

Figure 14: Base Models - YOLOv10 - Sample Predictions

Lastly, the image below shows the prediction of the base Faster R-CNN model on the same image samples. In the first sample, it identifies the car properly but misses the heavy category of tram. It accurately identifies both cars in the third image; however, there are a few false positives in the second image.

Figure 15: Base Models - FRCNN - Sample Prediction

The inference speed for the YOLO models was consistently between 7 to 15 milliseconds per image whereas for Faster R-CNN was above 40 milliseconds.

5.1.3 Model Performance with Augmentation

The result of the primary dataset was tried to be improved upon by applying multiple augmentation techniques. The augmentations applied were horizontal flipping, hue between –12and 12 degrees, brightness between –12 and 12 percent, and noise up to 0.1% of pixels. Because the dataset was split into 80%, 10% and 10% for training, validation and testing respectively, because of augmentation, the training size was tripled to 1040 images.

5.1.3.1 Mean Average Precision

The experiment was first carried out with the YOLOv8 pretrained model and if there were improvements seen with it, the same was applied to the other models. On a few iterations, some improvements were observed in the YOLOv8 with the mAP@50 reaching 30 percent and mAP@50-95 reaching almost 15%, which is a 5% increase compared to the base model for YOLOv8. The YOLOv10 model, however, did not show as many improvements as expected. With the same level of augmentation as the v8 model, the mean average precision at both levels slightly decreased instead.

Classes		YOLOv8	YOLOv10	Faster R-CNN
Car	mAP@50	0.535	0.467	0.2190
	mAP@50-95	0.264	0.228	0.1314
2-wheelers	mAP@50	0.099	0.0183	0.0000
	mAP@50-95	0.0199	0.0056	0.0000
bus	mAP@50	0.568	0.405	0.2277
	mAP@50-95	0.031	0.188	0.1366
heavy	mAP@50	0.000	0.000	0.0000
	mAP@50-95	0.000	0.000	0.0000
Overall	mAP@50	0.301	0.222	0.1117
	mAP@50-95	0.148	0.106	0.0670

Table 2: Base Models - mAP@50 and mAP@50-95 Comparison

Figure 16: Augmented Models - mAP@50 & mAP@50-95 Comparision

Faster R-CNN, which had the lowest base mAP@50 of around 8% in base model, showed only marginal improvement with data augmentation. Its mAP@50 on the augmented dataset stabilized between 8% and 10%, with mAP@50-95 remaining low as well.

5.1.3.2 Training Loss

As shown in the figure below, YOLOv8 on the augmented dataset showed lower initial and final loss values across all components (box, classification, and dfl loss) compared to its base model, indicating more efficient learning.

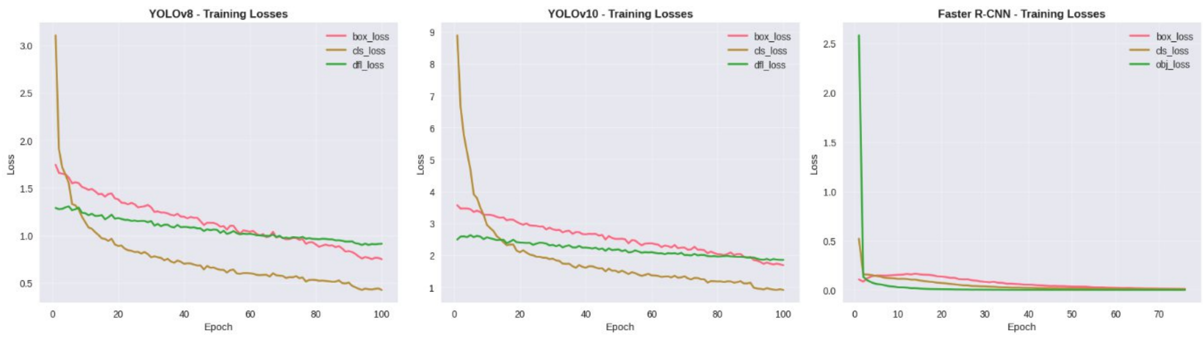


Figure 17: Augmented Models - Loss Comparison

YOLOv10's loss curves for the augmented dataset remained similar to its base model and Faster R-CNN demonstrated the most significant change, with its augmented dataset training beginning with drastically lower initial loss values across all components compared to its base model, while retaining rapid convergence.

5.1.3.3 Precision and Recall curves

For YOLOv8 compared to its base model, both precision and recall generally achieved higher peak values and maintained improved average scores throughout training, indicating a more robust performance.

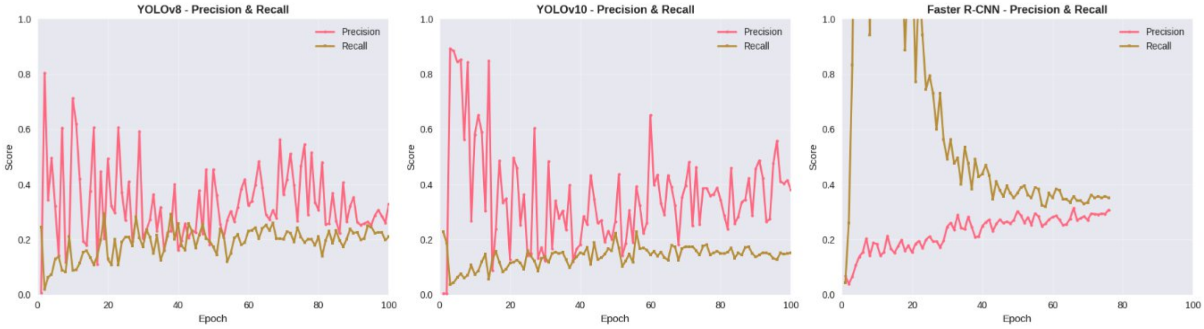


Figure 18: Augmented Models - Precision & Recall Curve

In contrast, the precision and recall for YOLOv10 with the augmented dataset closely resembled those of its base model, showing high volatility and overall low scores without clear signs of improvement, confirming the mAP score from above. Faster R-CNN, when trained with the augmented data, showed a slight upward trend in its precision compared to its consistently low base performance, and its recall, though remaining volatile, appeared to stabilize at marginally higher levels in later training epochs than that of its base model.

5.1.3.4 Augmented Models Summary

Like the base model, the training time for the augmented dataset for FRCNN was significantly larger than those of the two YOLO models without any substantial growth in precision and accuracy to go with it.

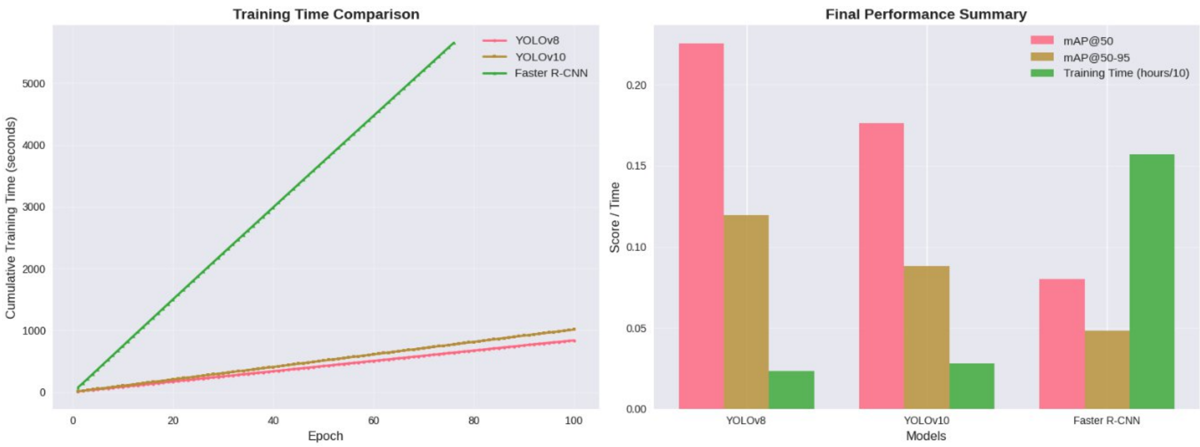


Figure 19: Augmented Models - Training Time and Summary

5.1.3.5 Inference

The YOLOv88 model accurately detected most of the objects better than the base model. In the first image, it correctly identified the different class objects and the confidence score stayed at a high level.

Figure 20: Augmented Models – YOLOv8 - Prediction

YOLOv10 on the other hand deteriorated as shown in the figure below with the detection accuracy as well as the confidence score went down considerably.

Figure 21: Augmented Models – YOLOv10 – Sample Prediction

The figure below shows the prediction made by the FRCNN model trained on the augmented dataset. Compared to the base model, the augmented version showed some improvements with the first image.

Figure 22: Augmented Models - FRCNN - Sample Prediction

5.1.4 Model Performance with Integrated Secondary Dataset

To overcome the limitations of the primary set due to its size and to further support the analysis, the pre-processed secondary dataset mentioned in section 4.6.2 was used to train the same three models. During exploratory data analysis, it was found that the secondary dataset of 6247 images has a total of 43213 bounding boxes, which were distributed as shown be-low.

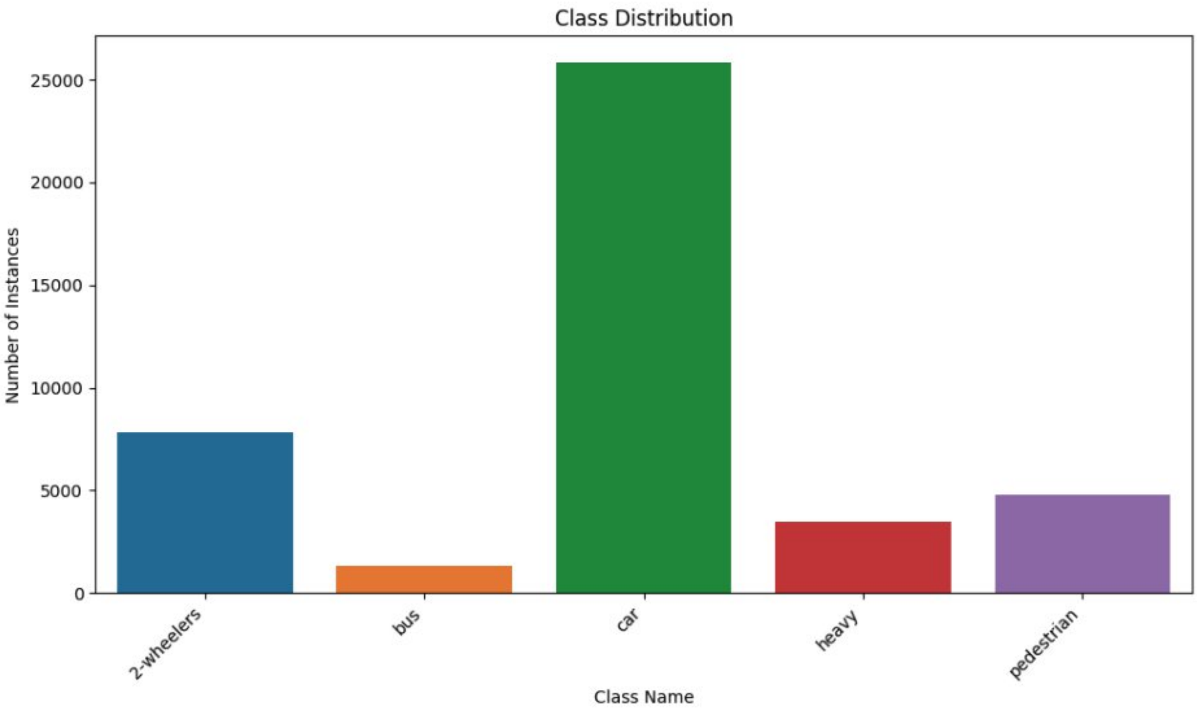


Figure 23: Secondary Dataset - Class Distribution

The YOLOv8 model instance from the previous training was used as a pre-trained model to further train with the secondary dataset. The dataset was trained without any augmentation for 50 epochs. At the end of it, the model showed huge improvements compared to both the base models.

5.1.4.1 Mean Average Precision

The mAP performance of all three models – YOLOv8, YOLOv10, and Faster R-CNN – showed a significant improvement when trained with the secondary dataset, far surpassing results from both the base and augmented primary datasets. YOLOv8 achieved exceptional accuracy, with its mAP@50 rapidly rising to stabilize consistently above 0.90, reaching close to 0.95, and its mAP@50-95 similarly increasing to over 0.80.

Metric	YOLOv8	YOLOv10	Faster R-CNN
mAP@50	0.952	0.94	0.78
mAP@50-95	0.845	0.837	0.47
Precision	0.922	0.887	0.78
Recall	0.879	0.871	0.63

Table 3: Overall mAP@50, mAP@50-95, Precision and Recall

YOLOv10 showed a similar, near-identical performance trajectory to YOLOv8, with its mAP@50 and mAP@50-95 also stabilizing above 0.90 and 0.80 respectively. Faster R-CNN, while not reaching the peak performance of the YOLO models, showed a substantial increase in its mAP@50, stabilizing around 0.75-0.80, and its mAP@50-95 reached approximately 0.45-0.50. From the Figure 24 it can also be observed that models displayed rapid initial increases in mAP, suggesting quick learning from the new dataset.

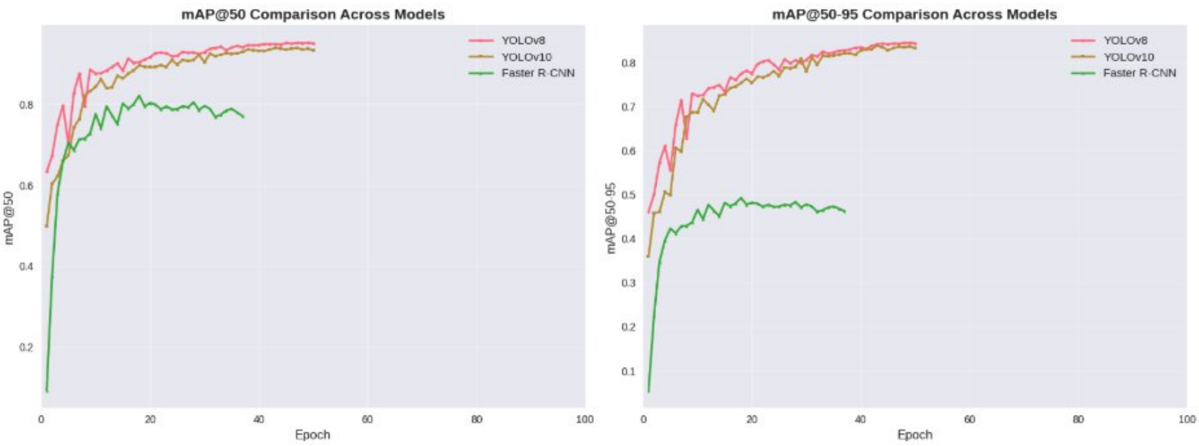


Figure 24: Secondary Dataset trained Model - mAP@50 & mAP@50-95 comparison

5.1.4.2 Training Loss

YOLOv8's loss components (box, classification, and dfl loss) decreased over 50 iterations, with final values reaching around 1.0 for box loss, 0.6 for classification loss, and 1.1 for dfl loss. YOLOv10 losses also showed a clear downward trend over, with classification loss experiencing a significant initial drop, stabilizing around 1.4, while box and dfl losses ended near 2.1.

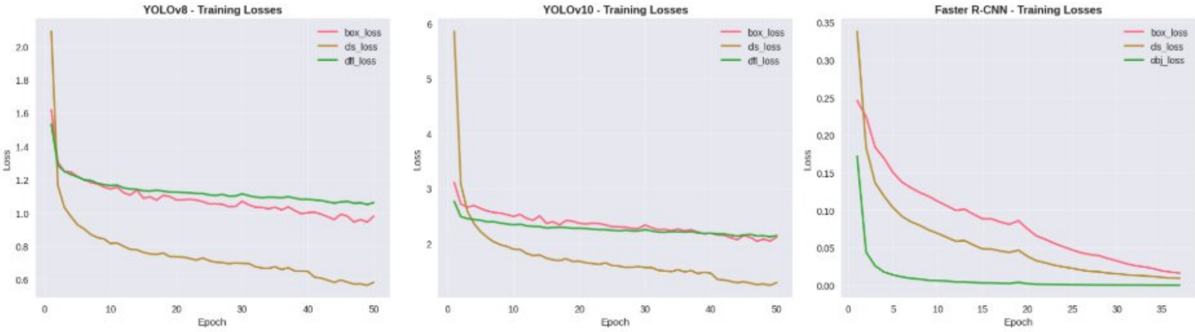


Figure 25: Secondary Dataset trained Model – Loss Comparison

Faster R-CNN also showed rapid convergence within the first few epochs; its box, classification, losses, starting near 0.35, quickly plummeted to near-zero values, indicating very efficient optimization on this dataset.

5.1.4.3 Confusion Matrix

The confusion matrices also revealed significantly improved classification accuracy across all models compared to their performance on the primary dataset. For YOLOv8, "2-wheelers" and "car" classes were correctly classified at 95.0%, and "bus" at 94.8%, with "heavy" vehicles showing 78.5% accuracy. YOLOv10 achieved 95.0% accuracy for "car", but lower rates for "2-wheelers" (75.0%) and "bus" (73.3%), with "heavy" at 74.4%. Faster R-CNN also demonstrated high classification rates, correctly identifying "2-wheelers" and "bus" at 95.0% and "heavy" at 84.5%, with "car" at 84.0%.

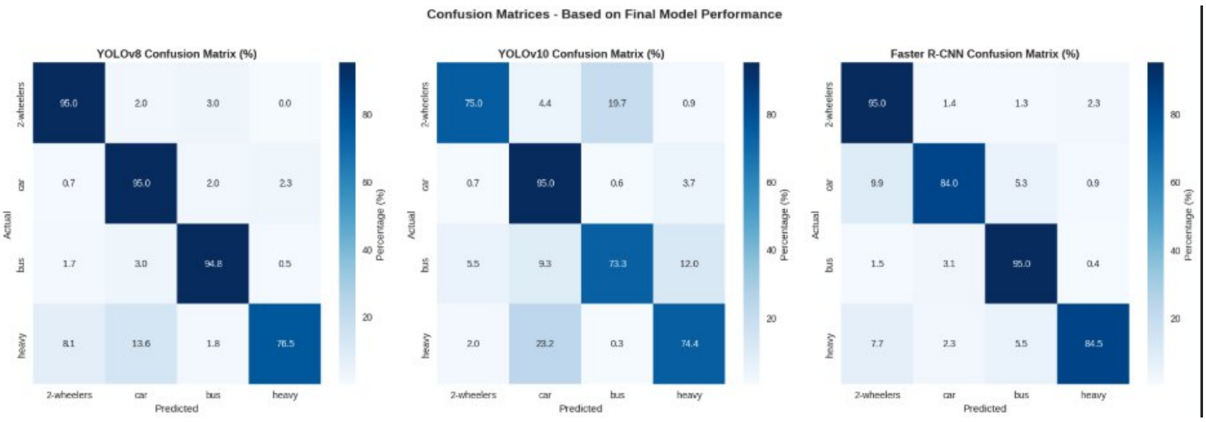


Figure 26: Secondary Dataset trained Model – Confusion Matrix

5.1.4.4 Secondary dataset summary

Faster R-CNN required more computational resources, with its cumulative training time reaching close to nearly 4 hours over approximately 50 epochs. In contrast, both YOLOv8 and YOLOv10 completed their training within 3,000 seconds over the same epoch count, showing significantly higher training efficiency. To summarize, YOLOv8 and YOLOv10 achieved remarkably high mAP@50 and mAP@50-95 scores, while consuming minimal training time in comparison to FRCNN.

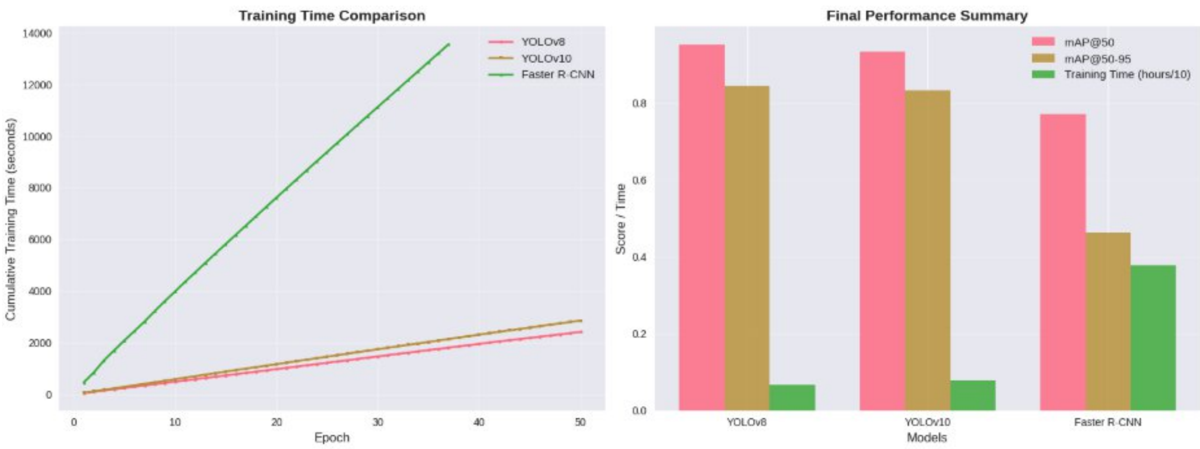


Figure 27: Secondary Dataset trained Model – Training Time & Summary

5.1.4.5 Inference

The figure below of the YOLOv8 model's predictions shows its highly accurate in detecting all the objects with high accuracy and prediction. Compared to the previous version with just primary data, the model is more confident detecting lower occurrence classes such as 'heavy' and 'bus'. It still struggled to correctly differentiate the occlusion scenario in the middle sample.

Figure 28: Secondary Dataset trained Model – YOLOv8 – Sample Prediction

YOLOv10, shown below, also showed substantial improvements. Compared to YOLOv8, the confidence scores were slightly below and in the first sample it failed to identify the tram. Likewise, the previous model also produced false positives of two different vehicles in the middle sample.

Figure 29: Secondary Dataset trained Model – YOLOv10 – Sample Prediction

Despite lower accuracy scores than the previous models, this model was highly confident in its detection with all the predictions made with full certainty. It failed to correctly detect the occlusion in the sample from the model and also failed to identify all the vehicles in the first sample image.

Figure 30: Secondary Dataset trained Model – FRCNN – Sample Prediction

5.1.5 Summary

The use of secondary datasets elevated the performance across all evaluated models, effectively overcoming the limitations observed with the primary dataset. Both YOLOv8 and YOLOv10 achieved very high detection accuracy, with mAP@50 consistently exceeding 90% and mAP@50-95 over 80%, coupled with highly efficient training times significantly shorter than Faster R-CNN.

Their precision and recall scores also demonstrated very high classification accuracy across most vehicle categories. Faster R-CNN, while also showing substantial improvement in mAP (mAP@50 around 78%) and achieving rapid loss convergence, required considerably more training resources and did not match the YOLO models' peak mAP, suggesting a trade-off in efficiency and overall accuracy.

5.2 Analysis

This chapter presents a comprehensive analysis of the experimental outcomes from the findings chapter above by interpreting the observed metrics to determine the trends of YOLOv10, YOLOv8, and Faster R-CNN models under varying dataset conditions. By critically evaluating performance across base, augmented primary, and secondary datasets, this analysis explains the critical factors influencing robust vehicle detection in low-light environments. Lastly, it summarizes how the findings helped achieve the objectives outlined during the initial phase of the research.

5.2.1 Baseline models analysis

The mean average precision of the base models was very low for each individual model. As seen from table 1, the overall mAP@50 percent for YOLOv8 was just over 25%, which surprisingly went down to 23% for the YOLOv10 baseline for the exact same training set. This was unexpected because of the assumption that the more modern approach of the v10 model should have outperformed the YOLOv8 models based on the literature studied above but the nature of the samples meant it is not conclusive enough to say whether it's the result of the small dataset or because the dataset, primarily including only images taken in nighttime meant the model was not optimized (Liu et al., 2019).

The precision and recall graphs further explained these mAP values; YOLOv8 and YOLOv10 displayed highly volatile precision and recall curves, indicating very unstable learning process. Faster R-CNN, while showing rapid recall fluctuations in early epochs, maintained a consistently low precision throughout training, suggesting a struggle to accurately classify detected objects without generating numerous false positives, which contributed to its lower overall mAP, which failed to reach 10 percent.

This was also likely because of the small number of images in the primary dataset and relatively low number of classes per image (Wang et al., 2023). The ability of the model to detect cars at an acceptable rate showed that despite low precision, because of the good nature of pre-trained YOLO models, it was able to detect images properly.

The training loss plot in Figure 10 showed that the loss was converging as expected to lower rate in the YOLO models, but it dropped steeply for the FRCNN model and smoothed out the rest of the way. The FRCNN model also failed to extract enough features from the images and even failed to improve consistently past 60 iterations. This could be attributed to the two-shot approach of the Faster R-CNN approach, where it first must identify potential areas before processing (Redmon and Farhadi, 2018). This showed up in the inference time as well, where it was considerably slower than the other models to predict the objects in the image.

As shown in Figure 12, the training time comparison reveals significant differences between the models. Faster R-CNN required more training time, requiring over 2000 seconds over 80 epochs. In contrast, both YOLOv8 and YOLOv10 trained considerably faster, completing 100 epochs in under 500 seconds. Faster R-CNN's training time, even when scaled, is very high relative to its mAP scores. This showed that Faster R-CNN is more resource consuming at the same time failed to provide baseline accuracy in this specific scenario (Wang et al., 2024). On the other hand, YOLOv8 and YOLOv10 offer a much more efficient training method, as well as providing a quick inference speed although with moderate baseline accuracy, suggesting that any potential trade-off in accuracy is worth the lower computational cost at the base level.

5.2.2 Augmentation Performance Analysis

The augmentation applied to the YOLOv8 model with multiple iterations provided successful results for the v8 model, as demonstrated by an increase of around 5% in table 2. Augmentation mostly helps model training by diversifying the dataset and increases the set itself, which probably mitigated some of the shortcomings of the primary dataset by tripling the training images with slightly different profiles, which makes augmentation a common step to train models in, especially for small dataset (Shorten and Khoshgoftaar, 2019).

However, application of exact same augmentation to the v10 model resulted in a worse outcome than the already poor performing base model, which was unexpected for a newer model. This suggested that the augmentation tailored to a model trained on the same exact dataset and of the same family of the model, did not necessarily translate to the benefits of the v10 model. It is possible that the newer v10 model has a more refined architecture and as a result the augmentation introduced noise to the images instead of extracting its features. It can be interpreted as data augmentation not having a fit-for-all solution and needs to be tailored to each model individually (Perez and Wang, 2017).

The observed improvements in YOLOv8's precision and recall with augmentation strongly suggested that the additional data enabled the model to reduce false positives and enhance its ability to correctly identify true objects, directly supporting its previously noted increase in mAP (Syahrudin et al., 2024). On the other hand, YOLOv10's unchanged and volatile precision and recall profiles despite augmentation suggests that the specific augmentation strategy employed was not optimally suited for its architecture, failing to provide the necessary improvements for accurate object detection.

For the FRCNN model on the other hand, the metrics indicated that augmentation did not significantly degrade its performance like YOLOv10, but it also did not provide a substantial boost seen with YOLOv8. However, the prediction on the sample images showed clear improvements as shown in Figure 22 compared to Figure 15 with the model correctly detecting all the different objects and showed less false positives in the second image. There is a possibility that the model is better than the mean average precision suggests. This could happen because the metrics are calculated with the valid dataset at the end of each iteration and it might not represent the model state properly.

There is also a possibility that the sample or a variant of the sample data is present in the training data, and the model is trained on that. This performance characteristic, combined with its considerably longer training time compared to the YOLO models, which considerably increased compared to base models, reinforces the observation that Faster R-CNN, in this low-light context with limited data, did not present a compelling accuracy advantage despite its two-stage complexity (Ge et al., 2021).

5.2.3 Secondary Dataset Performance Analysis

While there was an argument for iterating over and over the primary dataset until it reached an optimal accuracy level, there was a higher risk and likelihood for the model to over-fit and get worse in the long run. The graph of loss for both the base models and augmented models shows instability, suggesting the models were struggling to converge and having to oscillate back and forth. Hence, the secondary dataset was used to further train the data.

First observation of the importance of the secondary dataset was observed in the mean average precision score right away. As shown in table 3 and figure 24, this dramatic increase highlights the critical role of dataset size and diversity to train a high performing detection model. For YOLOv8 and YOLOv10, the mAP scores indicated that with sufficient and varied data, these modern architectures are able to learn highly robust features, overcoming the low performances seen with the primary dataset. Faster R-CNN's significant improvement, despite not matching the YOLO models, demonstrates its capacity for robust learning when provided with adequate data, fundamentally addressing its previous struggles.

The loss observed in figure 25 when models were trained on the secondary dataset directly correlates with the improvements in mAP. YOLOv8 compared to base, and particularly the significantly reduced initial and final losses for Faster R-CNN compared to both base and augmented primary datasets, confirm that the models benefited immensely from the richer data. While the mAP for FRCNN still was below the YOLO models, its near-zero convergence showed the capability of the two-shot model's ability to learn when provided with enough higher quality data.

While Faster R-CNN also showed substantial improvement in accuracy, its disproportionately higher training time and comparatively lower mAP (especially mAP@50-95) suggest that despite its architectural complexity, it was less efficient and ultimately less effective than the YOLO models in fully leveraging this high-quality dataset for optimal performance in this specific task.

However, while all the metrics suggested that the YOLO based models achieved near-perfect detection capabilities and FRCNN failed to reach similar heights, the prediction on sample images suggested a slightly different outcome. As shown in figure 28 and figure 29, YOLO based models still made some mistakes correctly detecting the objects. All three models struggled to correctly detect the car that occluded in the middle sample, detecting first and second half as separate cars. The FRCNN model's confidence score of 1.00 for its prediction suggested that, while the training time was higher and the accuracy was lower, the two-stage detection algorithm converged nicely to detect the vehicles.

5.2.4 Summary

The analysis of the findings highlighted that model performance on low-light vehicle detection is influenced highly by data availability and quality and helped achieve the objectives set during the early part of the project. Initially, the evaluation of YOLOv10, YOLOv8, and Faster R-CNN on the primary dataset fulfilled the objective of implementing and fine-tuning models while assessing their baseline performance in low-light conditions using metrics like mAP and training losses.

The following analysis of augmented primary data showed mixed outcomes, with YOLOv8's enhanced mAP and stabilized performance, but YOLOv10's metrics remained largely unaffected, underscoring the model-specific nature of augmentation. Crucially, the objective of training models on a larger secondary dataset to evaluate for further accuracy and speed of evaluation was achieved which showed increment in mAP across all models, with YOLOv8 and YOLOv10 consistently exceeding 90% mAP@50 and demonstrating superior training efficiency compared to FRCNN. This comprehensive process, from baseline models performance to comparative analysis with secondary dataset-based performance metrics, thus fulfilled the objective of presenting and comparing findings of the main objective.

5.3 Discussions

5.3.1 Introduction

This chapter provides a comprehensive discussion of the findings in the above section within the context of low-light vehicle detection and aims to interpret the significance of the results, with the context of research questions posed in Chapter 1, and relate the study's contributions to the broader field. Furthermore, it will discuss the limitations encountered during the research.

5.3.2 Research Questions Answered

The first research question asked how the three models compared in terms of accuracy and precision while detecting vehicles in low-light conditions. Initial experiments with the limited primary dataset revealed very low mAP scores and volatile precision-recall curves across all models, underscoring the challenges of similar data in this sector, especially in the pretrained models. While data augmentation successfully improved YOLOv8's accuracy, it proved ineffective for YOLOv10, showing the model-specific nature of augmentation. The integration of a secondary dataset dramatically improved performance on all three models, leading to exceptionally high mAP values and near-perfect classification accuracies for all models.

The YOLOv8 model was the answer to the second research question for the model that provided the most practical suitability for real-world low-light applications. The application of it in autonomous driving remained inconclusive because it is such a high-stake application, where even minimal latency and inaccuracy can result in catastrophic outcomes. While the inference speed of both the YOLO models remained comparable and the final outputs suggests either of the YOLO models would be perfectly acceptable, the higher accuracy of the v8 model will be perfect in the field of traffic management and vehicle surveillance.

Lastly, the performance difference between YOLOv8 and YOLOv10 was surprisingly in favor of the older YOLOv8 model, which performed better for both the primary and secondary datasets. This can be interpreted as a positive outcome that despite the latest development in the detection field, YOLOv8 with more recent YOLOv11 and more in development, the v8 still performs at an exceptional level.

Regarding FRCNN, while YOLOv8 and YOLOv10 separated themselves as both highly accurate and computationally efficient detectors, surpassing FRCNN in overall performance metrics and training speed, the FRCNN model still showed its capability despite the lower score by confidently detecting objects in sample images. It showed that with enough data to support and training infrastructure, it can still outperform the single-shot models.

5.3.3 Research Beneficiaries

Firstly, this research contributes to the existing computer vision research community by highlighting the importance of a comprehensive dataset to achieve optimal outcomes, irrespective of the field of study. Secondly, building upon the research answers, this project offers significant benefits to several key stakeholders in the field of object detection, such as traffic management and surveillance systems and autonomous driving even during challenging low light environments. For traffic management, the findings provide quantitative evidence to support deployment of systems capable of traffic flow optimization and monitoring. Although not fully conclusive in high stake environments like autonomous driving, the findings from this research could still be used as a study point to gain insights from.

Lastly, even though the shortcomings of the FRCNN models regarding training time and inference speed which were observed here, makes it unusable in fields where real-time detection is of paramount importance, sectors where precision is more important than quickness and the resources and infrastructures are not any factors could still use this study to support and further their applications.

5.3.4 Limitations

Despite gaining valuable insights from the research, it had few limitations that may have influenced the outcome of the experiments and the research in general.

Scope of low light environment

The primary dataset for the experimentation was restricted to the specific definition of a low-light environment. This meant the data collection strictly focused on nighttime, thus potentially skewed the dataset towards only having representation of one type of low-light setup and not for other kinds of harsh environments such as overcast conditions, fog, heavy rain, etc. As a result, models trained on such skewed dataset will have difficulty in generalizing, possibly under-performing in such scenarios.

Variety of images

A significant limitation came as a result of the variety of images in the dataset. Not only were the images for the primary research only taken from phone cameras, but it was also difficult to collect data from secondary sources, that were of optimal nature and distribution from multiple sources. For example, most data were taken from the same angle, either from inside the vehicle or from a static position on the road. This lack of diversity meant that even with augmentation, the models struggled to develop strong capabilities between the objects and even with high accuracy with secondary dataset; some expected detections were not optimal.

Resources to train the model

The computational resources required to train the models provided another significant road-block. While due to the training speed, there wasn't much challenge in training the YOLO models. Once it came to Faster R-CNN, the constraint of restricted resources in Kaggle and Google Collab became a challenge. This was especially felt when it was time to iterate over the models with hyper-parameter tuning, and the training time made it impractical to keep tuning it to improve the results.

Another key issue as a result of resource constraint was further experimentation with different levels of pre-trained models. For this project, the pre-trained models chosen were smaller in size and subsequently, the parameters in them. With enough computational resources to train models, bigger models could be used for better results.

Differences in Models

One of the key variables while trying to perform comparative analysis was trying to keep everything equal for all three models. The experimental design, particularly when attempting to transfer successful strategies, introduced complexity stemming from the number of variables at play. The process of experimenting with augmentation, initially optimized for YOLOv8, highlighted this. While specific augmentation strategies proved beneficial for YOLOv8, leading to a notable mAP improvement, it did not transfer as expected to other models despite everything else being the same. Other variables include the pre-trained models. Due to resource constraints, the smallest possible pretrained model was chosen for YOLO models provided by Ultralytics. There was no way to evaluate the difference between those models with that of the Faster R-CNN pretrained model from PyTorch to compare things equally.

Presentation of Findings

A significant challenge encountered during the research was presenting findings, and the process of selecting the most appropriate charts and visualization methods proved difficult. Furthermore, the task of effectively explaining these complex visual representations and ensuring they told the complete story of the models' behavior and their performance in low-light conditions was an ongoing challenge and might provide misleading information or be misinterpreted.

Conclusion

The challenge of highly accurate vehicle detection in challenging low-light environments is of significant importance for the advancement of intelligent transportation systems, traffic monitoring, and autonomous driving. This research project aimed to compare three different object detection models, YOLOv8, YOLOv10 and Faster R-CNN, to evaluate their performance in key metrics such as mean average precision, precision, and recall scores in images with restricted visibility. Using that experimentation, the research aimed to answer research questions on which one provided the best metrics, their acceptance in real-world scenarios and their overall differences in performance, training time and inference speed.

First, a literature review was conducted in the broader field of object detection, which discussed the other applications of object detection in fields such as medical, facial recognition, and agriculture. The literature discussed the different object detection methods and architecture – CNN based methods and transformers-based methods. Since the models for this project are all CNN based, further analysis was done on different types of this approach – single-shot method, which is used by the YOLO models and two-stage method of the Faster RCNN model. Upon study of the existing literature, the gap between studies done on object detection with regular environment compared to low-light environment was highlighted. Studying previous work showed the challenges of low light detection was a result of noise introduced in the images by their environment.

A more constrained literature review was done in the next section, focusing primarily on the smaller scope of the past research, specifically on vehicle detection. It discussed the importance of object detection in low light in the field of vehicle to further autonomous driving, traffic management and decreasing death by road accidents. It was identified that the YOLO models offered advantage in speed whereas the two-stage method was more accurate. It was identified that the existing works, such as that of Geetha et al. (2024), did not cover enough ground, relying on the standard benchmark datasets, with assumption of generalization in low-light environment as well. A gap was identified in the field where not a lot of comprehensive, comparative research was conducted in this specific field of vehicle detection and as a result the importance of this project was highlighted.

For comparative analysis, a primarily quantitative research methodology was implemented and was grounded in measuring key performance metrics such as precision, recall and inference time. A total of 400 images, calculated using statistical formula to represent 96% desired certainty, were taken for primary dataset, using smartphone cameras, primarily during nighttime and over 6000 images were curated from different secondary sources to support the primary research. The dataset was manually annotated using Roboflow, with the FRCNN model trained in PyTorch and the YOLO models trained using the Ultralytics development kit.

First, the primary dataset was trained without any augmentation for all three models and then later with augmentations over multiple iterations. The findings showed a clear progression in model performance directly correlated to the characteristics of the training data. Initial evaluations from just primary dataset showed that all models struggled, exhibiting low mean Average Precision and volatile performance, which was interpreted as the difficulties of object detection with low number of images in dataset. The following phase, involving data augmentation on this primary dataset performed interesting results - while YOLOv8 notably improved its mAP, YOLOv10's performance remained largely unaffected, highlighting the model-specific nature of different augmentation methods. The introduction of the secondary dataset enabled all models to achieve significantly higher mAP scores and superior classification accuracy, showing that data quality and quantity were important in overcoming the challenges of low-light detection.

To conclude, while Faster R-CNN showed substantial improvement with the secondary dataset, its comparatively higher computational demands and lower peak mAP values, and YOLOv10 not improving as expected, the YOLOv8 model was decided as the more suitable for real-world low-light applications requiring both high precision and efficiency.

6.1 Support for existing literature

The findings from this research project largely supported the existing literature that present the importance of comprehensive and diverse datasets to achieve robust performance in deep learning setups. The speed and accuracy of both the YOLO models also supported the pre-existing literature that the single-shot models are vastly superior compared to two-stage detectors and that usage of the two-stage methods are not feasible for fields in real-time detection. Similarly, the training time for the two-shot Faster R-CNN model also followed the predictable outcome presented in previous studies.

6.2 Contradictions for existing literature

The findings of this project also presented certain conclusions that could be interpreted as a contradiction for the existing literature discussed in literature review 1 and literature review 2. While YOLOv8 demonstrated clear benefits from data augmentation with the primary dataset, YOLOv10's outcomes suggested that the same augmentation strategy best practices for data augmentation does not apply across all advanced model architectures. This highlighted a need for more model-specific augmentation techniques even with models within the same family.

6.3 Future Work

Building upon the insights gained and the limitations identified in this study, several future research projects can be done to advance this research. Primarily, future efforts should prioritize the expansion of the primary dataset as shown by the improvement with the secondary dataset. This will involve capturing images in an even wider array of low-light conditions, environment such as different weather condition, different seasonal data, etc. To further enhance the quality of data, the integration of multi-modal sensor data such as LiDAR and thermal cameras should also be considered.

Beyond the enhancement of the primary data, future work should extend to comparative analysis with newer models and architecture. For example, for YOLO models, there already are newer versions of the models that could possibly improve the results. Similarly, Transformers based approach could also be used for a more comprehensive analysis, which has become a modern option for vision-based tasks. Finally, to optimize the model performance, future studies can be done with more iterations of training with more diverse hyper-parameter tuning such as learning rates, batch sizes and regularization techniques.

Bibliography

Abbas, S. K., Khan, M. U. G., Zhu, J., Sarwar, R., Aljohani, N. R., Hameed, I. A. and Hassan, M. U. (2023) 'Vision based intelligent traffic light management system using Faster R-CNN' In: *CAAI Transactions on Intelligence Technology* 9 (4) pp.932–947.

Al Sobbahi, R. and Tekli, J. (2022) 'Comparing deep learning models for low-light natural scene image enhancement and their impact on object detection and classification: Overview, empirical evaluation, and challenges' In: *Signal Processing: Image Communication* 109 p.116848.

Alif, M. A. R. (2024) *YOLOv11 for Vehicle Detection: Advancements, Performance, and Applications in Intelligent Transportation Systems*. Directed by Alif, M.A.R. (s.l.): arXiv. At: http://arxiv.org/abs/2410.22898 (Accessed 02/04/2025).

Anandhalli, M., Baligar, P., Saraf, S. S. and Deepsir, P. (2022) 'Image projection method for vehicle speed estimation model in video system' In: *Machine Vision and Applications* 33 (1) p.7.

Bai, T., Luo, J., Zhou, S., Lu, Y. and Wang, Y. (2024) 'Vehicle-Type Recognition Method for Images Based on Improved Faster R-CNN Model' In: *Sensors* 24 (8) p.2650.

Balaji, A., Venkateswarlu, J., Addeti Gopi, Gottipati Surya, and Bailadugu Ajith (2025) 'Illuminating Low-Light Scenes: YOLOv5 and Federated Learning for Autonomous Vision' In: *ResearchGate* At: https://www.researchgate.net/publication/390948198_Illuminating_Low-Light_Scenes_YOLOv5_and_Federated_Learning_for_Autonomous_Vision (Accessed 20/04/2025).

Bochkovskiy, A., Wang, C.-Y. and Liao, H.-Y. M. (2020) *YOLOv4: Optimal Speed and Accuracy of Object Detection*. Directed by Bochkovskiy, A., Wang, C.-Y. and Liao, H.-Y.M. (s.l.): arXiv. At: http://arxiv.org/abs/2004.10934 (Accessed 05/04/2025).

Boesch, G. (2024) *Object Detection: The Definitive Guide*. At: https://viso.ai/deep-learning/object-detection/ (Accessed 12/03/2025).

Bučko, B., Lieskovská, E., Zábovská, K. and Zábovský, M. (2022) 'Computer Vision Based Pothole Detection under Challenging Conditions' In: *Sensors* 22 (22) p.8878.

Buhl, N. (2024) *YOLO Object Detection Explained: Evolution, Algorithm, and Applications*. At: https://encord.com/blog/yolo-object-detection-guide/ (Accessed 14/04/2025).

Bullard, E. (2024) *Purposive sampling | EBSCO Research Starters*. At: https://www.ebsco.com/research-starters/social-sciences-and-humanities/purposive-sampling (Accessed 24/04/2025).

Chen, W., Li, Y., Tian, Z. and Zhang, F. (2023a) '2D and 3D object detection algorithms from images: A Survey' In: *Array* 19 p.100305.

Chen, W. and Shah, T. (2021) *Exploring Low-light Object Detection Techniques*. Directed by Chen, W. and Shah, T. (s.l.): arXiv. At: http://arxiv.org/abs/2107.14382 (Accessed 17/03/2025).

Chen, Y., Huang, M., Liu, H., Shao, K. and Zhang, J. (2023b) 'Real-world low-light image enhancement via domain-gap aware framework and reverse domain-distance guided strategy' In: *Frontiers in Physics* 11 p.1147031.

detector_v2 Dataset (s.d.) At: https://universe.roboflow.com/smartscan/detector_v2 (Accessed 22/05/2025).

Ding, J., Hu, J., Lin, J. and Zhang, X. (2024) 'Lightweight enhanced YOLOv8n underwater object detection network for low light environments' In: *Scientific Reports* 14 (1) p.27922.

Diwan, T., Anirudh, G. and Tembhurne, J. V. (2022) 'Object detection using YOLO: challenges, architectural successors, datasets and applications' In: *Multimedia Tools and Applications* 82 (6) pp.9243–9275.

Frnda, J., Bandyopadhyay, S., Pavlicko, M., Durica, M., Savrasovs, M. and Banerjee, S. (2024) 'Analysis of Pothole Detection Accuracy of Selected Object Detection Models Under Adverse Conditions' In: *Transport and Telecommunication Journal* 25 (2) pp.209–217.

Ge, Z., Liu, S., Wang, F., Li, Z. and Sun, J. (2021) *YOLOX: Exceeding YOLO Series in 2021*. Directed by Ge, Z., Liu, S., Wang, F., Li, Z. and Sun, J. (s.l.): arXiv. At: http://arxiv.org/abs/2107.08430 (Accessed 24/05/2025).

Girshick, R., Donahue, J., Darrell, T. and Malik, J. (2014) *Rich feature hierarchies for accurate object detection and semantic segmentation*. Directed by Girshick, R., Donahue, J., Darrell, T. and Malik, J. (s.l.): arXiv. At: http://arxiv.org/abs/1311.2524 (Accessed 28/05/2025).

Hadi, R. A., Sulong, G. and George, L. E. (2014) *Vehicle Detection and Tracking Techniques: A Concise Review*. At: https://arxiv.org/abs/1410.5894v1 (Accessed 01/06/2025).

Han, Z., Yue, Z. and Liu, L. (2024) '3L-YOLO: A Lightweight Low-Light Object Detection Algorithm' In: *Applied Sciences* 15 (1) p.90.

He, J., Chen, H., Liu, B., Luo, S. and Liu, J. (2024) 'Enhancing YOLO for occluded vehicle detection with grouped orthogonal attention and dense object repulsion' In: *Scientific Reports* 14 (1) p.19650.

Hestness, J., Narang, S., Ardalani, N., Diamos, G., Jun, H., Kianinejad, H., Patwary, M. M. A., Yang, Y. and Zhou, Y. (2017) *Deep Learning Scaling is Predictable, Empirically*. Directed by Hestness, J., Narang, S., Ardalani, N., Diamos, G., Jun, H., Kianinejad, H., Patwary, M.M.A., Yang, Y. and Zhou, Y. (s.l.): arXiv. At: http://arxiv.org/abs/1712.00409 (Accessed 20/04/2025).

Keita, Z. (s.d.) *YOLO Object Detection Explained: A Beginner's Guide*. At: https://www.datacamp.com/blog/yolo-object-detection-explained (Accessed 10/03/2025).

Li, T., Wang, J. and Zhang, T. (2022) 'L-DETR: A Light-Weight Detector for End-to-End Object Detection With Transformers' In: *IEEE Access* 10 pp.105685–105692.

Li, Z., Dong, M., Wen, S., Hu, X., Zhou, P. and Zeng, Z. (2019) 'CLU-CNNs: Object detection for medical images' In: *Neurocomputing* 350 pp.53–59.

Liang, L., Ma, H., Zhao, L., Xie, X., Hua, C., Zhang, M. and Zhang, Y. (2024) 'Vehicle Detection Algorithms for Autonomous Driving: A Review' In: *Sensors* 24 (10) p.3088.

Lin, Z., Wu, Y., Ma, Y., Chen, J., Zhang, R., Wu, J., Yin, G. and Lin, L. (2025) *YOLO-LLTS: Real-Time Low-Light Traffic Sign Detection via Prior-Guided Enhancement and Multi-Branch Feature Interaction*. Directed by Lin, Z., Wu, Y., Ma, Y., Chen, J., Zhang, R., Wu, J., Yin, G. and Lin, L. (s.l.): arXiv. At: http://arxiv.org/abs/2503.13883 (Accessed 28/05/2025).

Liu, L., Muelly, M., Deng, J., Pfister, T. and Li, L.-J. (2019) *Generative Modeling for Small-Data Object Detection*. Directed by Liu, L., Muelly, M., Deng, J., Pfister, T. and Li, L.-J. (s.l.): arXiv. At: http://arxiv.org/abs/1910.07169 (Accessed 28/05/2025).

Loh, Y. P. and Chan, C. S. (2019) 'Getting to know low-light images with the Exclusively Dark dataset' In: *Computer Vision and Image Understanding* 178 pp.30–42.

Ma, C. and Xue, F. (2024) 'A Review of Vehicle Detection Methods Based on Computer Vision' In: *Journal of Intelligent and Connected Vehicles* 7 (1) pp.1–18.

McCombes, S. (2019) *Descriptive Research | Definition, Types, Methods & Examples*. At: https://www.scribbr.com/methodology/descriptive-research/ (Accessed 21/04/2025).

Moussaoui, H., Akkad, N. E., Benslimane, M., El-Shafai, W., Baihan, A., Hewage, C. and Rathore, R. S. (2024) 'Enhancing automated vehicle identification by integrating YOLO v8 and OCR techniques for high-precision license plate detection and recognition' In: *Scientific Reports* 14 (1) p.14389.

Perez, L. and Wang, J. (2017) *The Effectiveness of Data Augmentation in Image Classification using Deep Learning*. Directed by Perez, L. and Wang, J. (s.l.): arXiv. At: http://arxiv.org/abs/1712.04621 (Accessed 28/05/2025).

Premaratne, P., Jawad Kadhim, I., Blacklidge, R. and Lee, M. (2023) 'Comprehensive review on vehicle Detection, classification and counting on highways' In: *Neurocomputing* 556 p.126627.

ProjectTraffic (2025) *Vehicle_Detection Dataset*. At: https://universe.roboflow.com/project-traffic-1cf6g/vehicle_detection-m5pa5 (Accessed 24/05/2025).

Quach, L.-D., Quoc, K. N., Quynh, A. N. and Ngoc, H. T. (2023) 'Evaluating the effectiveness of YOLO models in different sized object detection and feature-based classification of small objects' In: *Journal of Advances in Information Technology* 14 (5) pp.907–917.

Reddy, S., Pillay, N. and Singh, N. (2024) 'Comparative Evaluation of Convolutional Neural Network Object Detection Algorithms for Vehicle Detection' In: *Journal of Imaging* 10 (7) p.162.

Redmon, J., Divvala, S., Girshick, R. and Farhadi, A. (2016) *You Only Look Once: Unified, Real-Time Object Detection*. Directed by Redmon, J., Divvala, S., Girshick, R. and Farhadi, A. (s.l.): arXiv. At: http://arxiv.org/abs/1506.02640 (Accessed 28/05/2025).

Redmon, J. and Farhadi, A. (2018) *YOLOv3: An Incremental Improvement*. Directed by Redmon, J. and Farhadi, A. (s.l.): arXiv. At: http://arxiv.org/abs/1804.02767 (Accessed 24/05/2025).

Region-based CNNs (R-CNNs) — Dive into Deep Learning 1.0.3 documentation (s.d.) At: https://d2l.ai/chapter_computer-vision/rcnn.html (Accessed 28/05/2025).

Ren, S., He, K., Girshick, R. and Sun, J. (2016) *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. Directed by Ren, S., He, K., Girshick, R. and Sun, J. (s.l.): arXiv. At: http://arxiv.org/abs/1506.01497 (Accessed 28/05/2025).

Roboflow: Computer vision tools for developers and enterprises (s.d.) At: https://roboflow.com (Accessed 02/04/2025).

Shorten, C. and Khoshgoftaar, T. M. (2019) 'A survey on Image Data Augmentation for Deep Learning' In: *Journal of Big Data* 6 (1) p.60.

Shovo, S. U. A., Abir, Md. G. R., Kabir, Md. M. and Mridha, M. F. (2024) 'Advancing low-light object detection with you only look once models: An empirical study and performance evaluation' In: *Cognitive Computation and Systems* 6 (4) pp.119–134.

Sivasubramanian, A., Prakash, J. A., Kumar, K. S. D., Prashanth, V. R., Sowmya, V. and Variyar, V. V. S. (2023) 'Object Detection Under Low-Lighting Conditions Using Deep Learning Architectures: A Comparative Study' In: Chakraborty, B., Biswas, A. and Chakrabarti, A. (eds.) *Advances in Data Science and Computing Technologies*. Singapore: Springer Nature Singapore. pp.269–276. At: https://link.springer.com/10.1007/978-981-99-3656-4_27 (Accessed 22/04/2025).

Sreekumar, D. (2023) *What is Quantitative Research? Definition, Methods, Types, and Examples | Researcher.Life*. At: https://researcher.life/blog/article/what-is-quantitative-research-types-and-examples/ (Accessed 01/05/2025).

Stephensons (2019) *Driving In The Dark - How The Number Of Accidents Rise Towards The End Of The Year*. At: https://www.stephensons.co.uk/site/blog/personal-injury-blog/driving-in-the-dark-accidents-rise-towards-end-of-year (Accessed 02/04/2025).

Sun, Y., Sun, Z. and Chen, W. (2024) 'The evolution of object detection methods' In: *Engineering Applications of Artificial Intelligence* 133 p.108458.

Sundaresan Geetha, A., Alif, M. A. R., Hussain, M. and Allen, P. (2024) 'Comparative Analysis of YOLOv8 and YOLOv10 in Vehicle Detection: Performance Metrics and Model Efficacy' In: *Vehicles* 6 (3) pp.1364–1382.

supagonova@gmail.com (2022) *g7_object_detection Dataset*. At: https://universe.roboflow.com/supagonava-gmail-com/g7_object_detection (Accessed 21/05/2025).

Syahrudin, E., Utami, E. and Hartanto, A. (2024) 'Augmentation for Accuracy Improvement of YOLOv8 in Blind Navigation System' In: *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)* 8 pp.579–588.

Tahir, N. U. A., Zhang, Z., Asim, M., Chen, J. and ELAffendi, M. (2024) 'Object Detection in Autonomous Vehicles under Adverse Weather: A Review of Traditional and Deep Learning Approaches' In: *Algorithms* 17 (3) p.103.

Tang, J., Wan, L., Schooling, J., Zhao, P., Chen, J. and Wei, S. (2022) 'Automatic number plate recognition (ANPR) in smart cities: A systematic review on technological advancements and application cases' In: *Cities* 129 p.103833.

Ultralytics (s.d.) *Ultralytics*. At: https://docs.ultralytics.com/ (Accessed 02/04/2025).

Wang, A., Chen, H., Liu, L., Chen, K., Lin, Z., Han, J. and Ding, G. (2024) 'YOLOv10: Real-Time End-to-End Object Detection' At: https://proceedings.neurips.cc/paper_files/paper/2024/file/c34ddd05eb089991f06f3c5dc36836e0-Paper-Conference.pdf (Accessed 10/04/2025).

Wang, Y., Xie, W. and Liu, H. (2022) 'Low-light image enhancement based on deep learning: a survey' In: *Optical Engineering* 61 (04) At: https://www.spiedigitallibrary.org/journals/optical-engineering/volume-61/issue-04/040901/Low-light-image-enhancement-based-on-deep-learning--a/10.1117/1.OE.61.4.040901.full (Accessed 02/04/2025).

Wang, Z., Zhan, J., Duan, C., Guan, X., Lu, P. and Yang, K. (2023) 'A Review of Vehicle Detection Techniques for Intelligent Vehicles' In: *IEEE Transactions on Neural Networks and Learning Systems* 34 (8) pp.3811–3831.

Waters, S. (2024) *Thermal vs Night Vision vs Infrared - Which One is Right for You?*. At: https://www.coram.ai/post/thermal-vs-night-vision-vs-infrared (Accessed 14/04/2025).

Watkins, D. C. (2023) *Using Secondary Data in Mixed Methods is More Straight-Forward Than You Think*. At: https://researchmethodscommunity.sagepub.com/blog/using-secondary-data-in-mixed-methods-is-more-straight-forward-than-you-think (Accessed 03/05/2025).

Zhang, Y., Carballo, A., Yang, H. and Takeda, K. (2023) 'Perception and sensing for autonomous vehicles under adverse weather conditions: A survey' In: *ISPRS Journal of Photogrammetry and Remote Sensing* 196 pp.146–177.

Zhao, Y., Lv, W., Xu, S., Wei, J., Wang, G., Dang, Q., Liu, Y. and Chen, J. (2024a) 'DETRs Beat YOLOs on Real-time Object Detection' In: *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Seattle, WA, USA: IEEE. pp.16965–16974. At: https://ieeexplore.ieee.org/document/10657220/ (Accessed 18/03/2025).

Zhao, Z., Alzubaidi, L., Zhang, J., Duan, Y. and Gu, Y. (2024b) 'A comparison review of transfer learning and self-supervised learning: Definitions, applications, advantages and limitations' In: *Expert Systems with Applications* 242 p.122807.

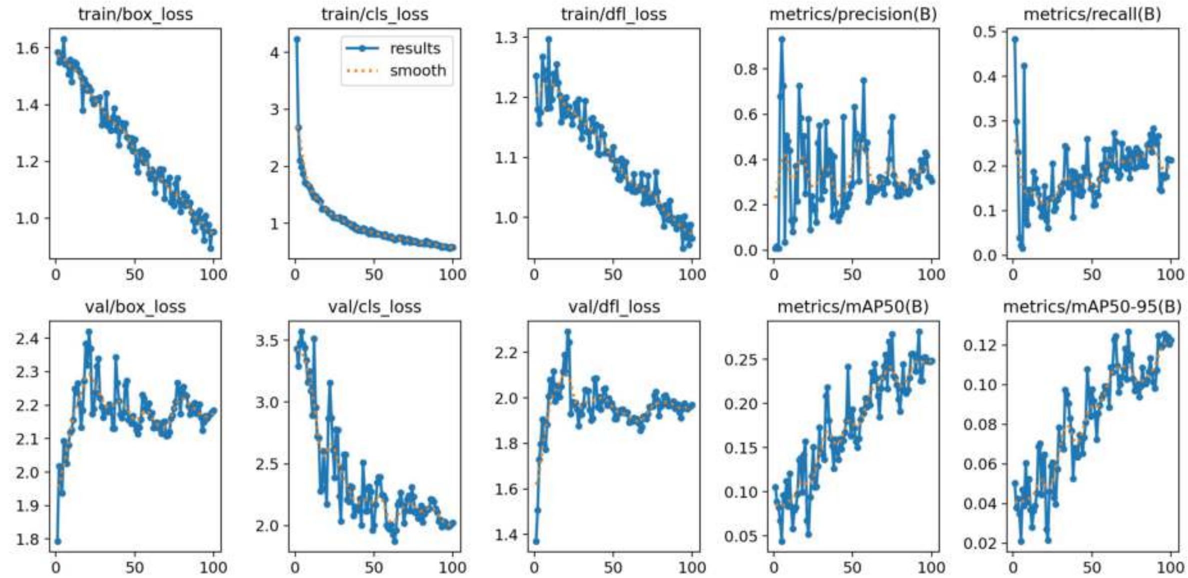
Table of Figures

Appendix

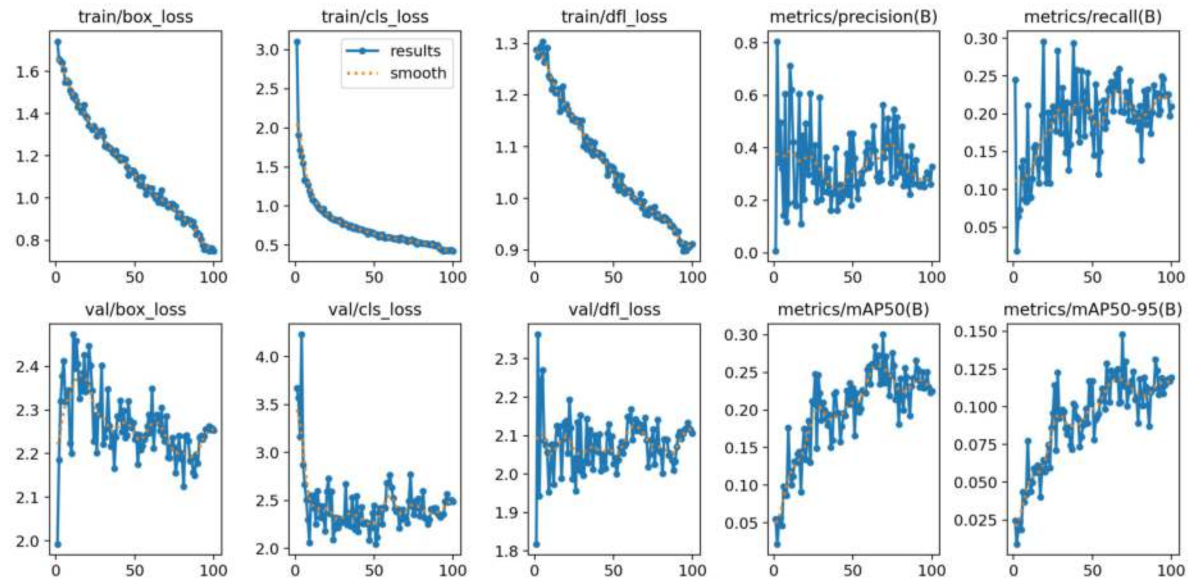
Findings, Analysis and Discussions

This part of the document consists of supporting visualization generated during the training process.

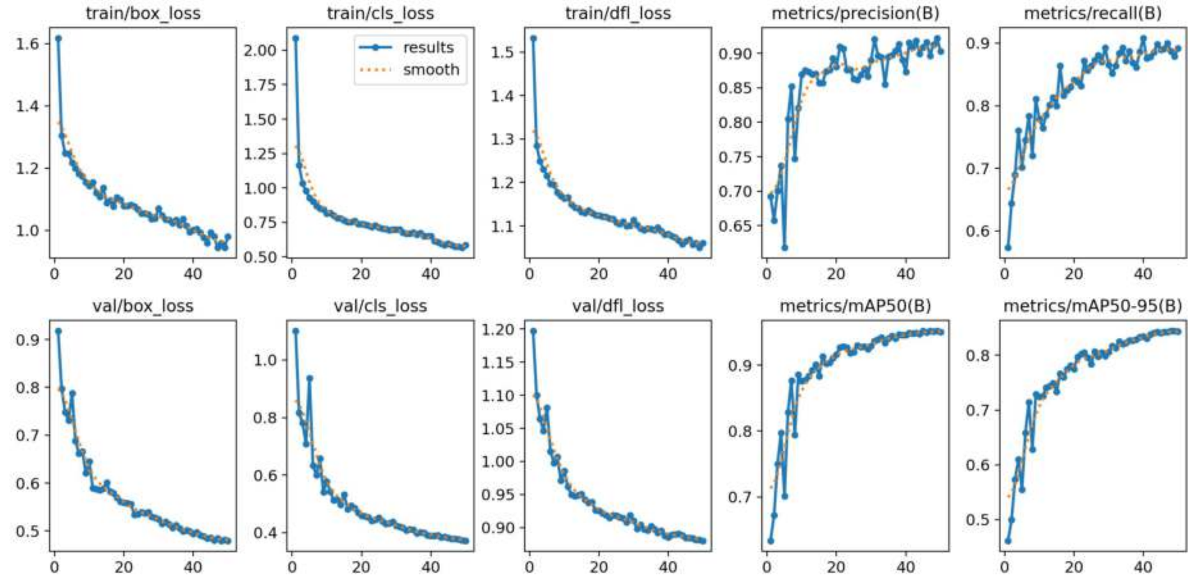
1. YOLOv8 Base Model complete findings:



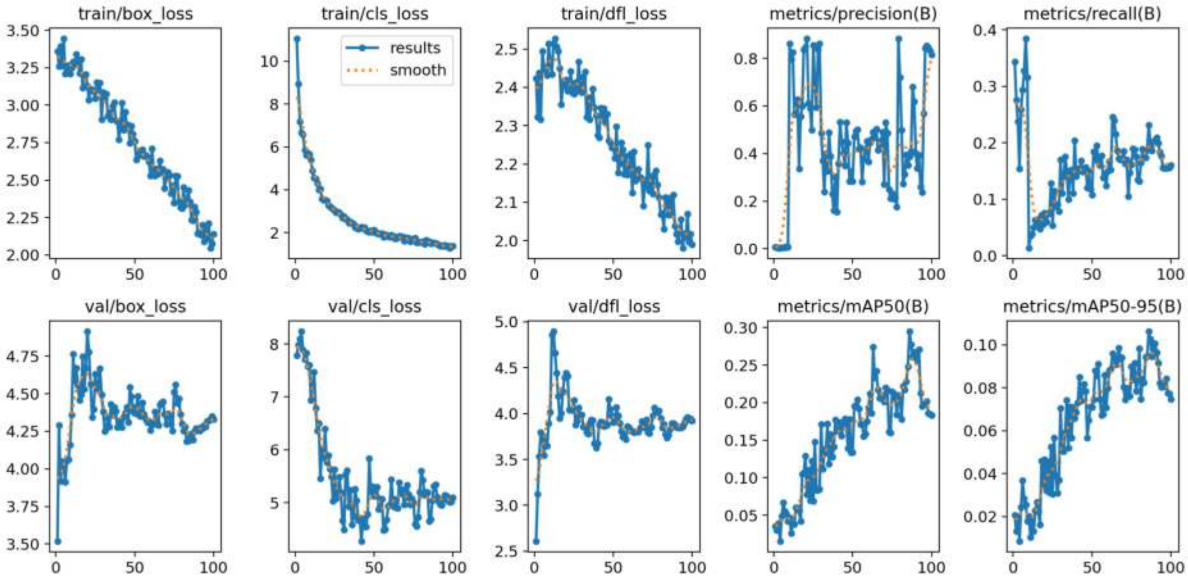
2. YOLOv8 Augmented Model complete findings:



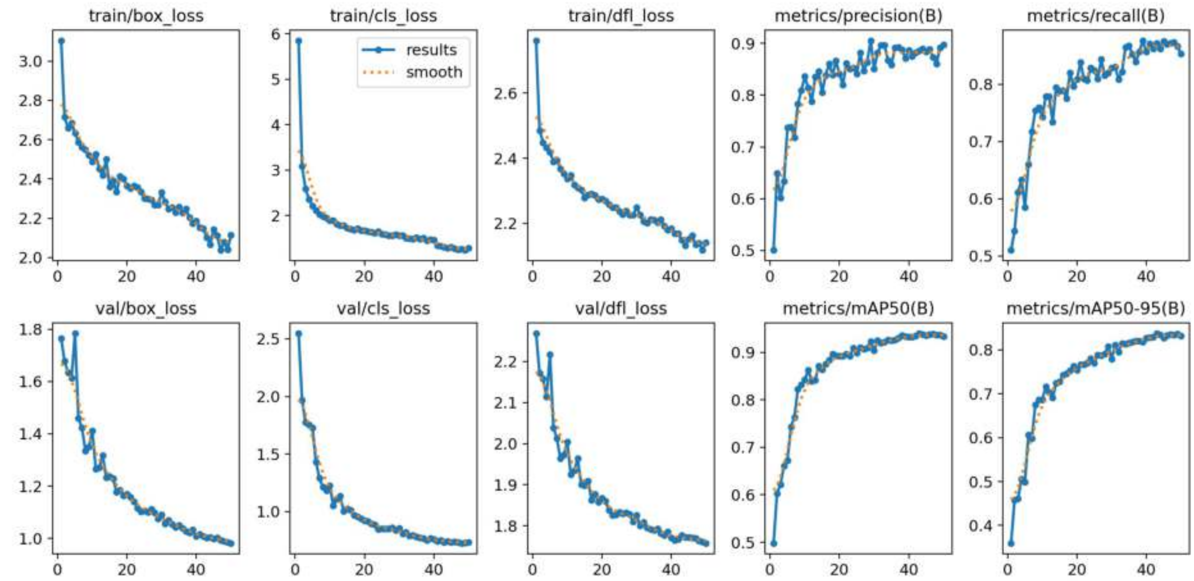
3. YOLOv8 Secondary Dataset Trained complete find-ings:



4. YOLOv10 Base Model Findings



5. YOLO10 Augmented Model Findings


train/box_loss
train/cls_loss
results
smooth
train/dfl_loss
metrics/precision(B)
metrics/recall(B)
val/box_loss
val/cls_loss
val/dfl_loss
metrics/mAP50(B)
metrics/mAP50-95(B)