Investigating CNN-based YOLO model architectures to improve vehicle detection accuracy in intelligent transportation systems

MSc Research Project
MSc in Data Analytics

Gowtham Sudharsan
Student ID: 23294418

National College of Ireland

Supervisor: John Kelly

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Gowtham Sudharsan
Student ID:	23294418
Programme:	MSc in Data Analytics
Year:	2024-25
Module:	MSc Research Project
Supervisor:	John Kelly
Submission Due Date:	15/09/2025
Project Title:	Investigating CNN-based YOLO model architectures to improve vehicle detection accuracy in intelligent transportation systems
Word Count:	7182
Page Count:	20

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Gowtham Sudharsan
Date:	15th September 2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Investigating CNN-based YOLO model architectures to improve vehicle detection accuracy in intelligent transportation systems

Gowtham Sudharsan
23294418

Abstract

Growing vehicle density in urban areas establishes major challenges to modern traffic management, particularly during harsh weather conditions. This paper investigates the potential of CNN-based YOLO models, i.e., YOLOv7, in achieving enhanced accuracy for detecting vehicles under intelligent transportation systems. Even though YOLOv7 has been most renowned for its real-time detection speed and detection accuracy, research has excluded it from its vulnerabilities in poor weather conditions. To connect this failing, the paper integrates YOLOv7 object detection with a ResNet50 model trained on the DAWN dataset to identify the state of weather by training YOLOv7 on the BDD10K dataset for the detection of eight traffic object classes where the models were coded in PyTorch and validated using the standard measurements of mAP, precision, recall, and classification accuracy. Results demonstrate the end-to-end pipeline's high accuracy across various environmental conditions with mAP@0.5 of 60.5% for object detection overall and over 85% for weather classification. A real-time dashboard simulator was developed with Gradio, illustrating the way in which the system can potentially be integrated into smart city systems. This two-model solution offers an end-to-end, scalable, modular, and context-aware approach to enhancing situational awareness, congestion control, and autonomous vehicle decision-making in high-density urban scenarios.

1 Introduction

Urbanization has led to a massive increase in the number of vehicles on the road, thereby establishing serious threats like traffic congestion, emergency response delays, and a rise in road accidents. The clear and static signal timing-based traffic management systems are hardly suitable to tackle these complicated problems, especially under dynamically evolving and critical conditions. Here comes the need for intelligent systems that can analyze real-time traffic and can adapt with weather abnormalities.

1.1 Research Background and Motivation

Over the past years, deep learning has become a strong challenger in the field of real-time object detection and scene understanding of which Convolutional Neural Networks

(CNN), particularly some versions of YOLO (You Only Look Once), has shown remarkable speed and precision in achieving tasks related to vehicle detection. The version of YOLO, i.e. YOLOv7, is introducing major architectural innovation with the initiation of the Extended Efficient Layer Aggregation Network (EELAN), leading to improved feature representation and multi-scale object detection.

However, it must be noted that most of these applications of YOLOv7 performance were studied under very conscious optimal conditions, and hence there are few studies regarding its performance under harsh weather conditions, including fog, rain, or low-light conditions. As such limitations put forward a great void in terms of usability of object detection models in real-world application scenarios, similarly, weather-resilient scene classification models like ResNet50 can identify weather from given images accurately. However, this potential has rarely been integrated with any object detection framework.

The study aims to create a reliable vision system in real-time that detects vehicles and understands environmental context. Using both YOLOv7 for vehicle detection and ResNet50 for weather classification, to build a dependable, real-time deployable system, that continues to perform well through all variability having an end-to-end pipeline that is based on the BDD10K dataset for detection and DAWN dataset for classification, with everything developed and tested in PyTorch and deployed via a web application using Gradio. Project containing the full pipeline ensures that our work will hold real practical relevance in addition to scientific significance.

1.2 Problem Statement and Research Context

Modern traffic monitoring systems must surely perform with high accuracy in different weather and lighting conditions to preserve public safety and operational efficiency. Nevertheless, the prevailing detection models often fail in real-world conditions due to limited adaptability and lack of contextual awareness. This project responds to the challenge by integrating together the state-of-the-art detection and classification models into a real-time solution for intelligent transportation systems.

1.3 Research Question

This study is guided by the following research question: **"To what extent are CNN-based YOLO (v7) architectures vulnerable to systematic optimization for enhancing the accuracy of vehicle detection in adverse environmental conditions, and how would these benefits be leveraged for the establishment of real-time congestion relief and violation sensing through resilient individual vehicle tracking in smart transportation systems."**

1.4 Research Objectives

- To investigate the adaptability of YOLOv7 under challenging environmental conditions.
- To examine the role of contextual scene understanding, such as weather classification, in supporting object detection.

- To explore the integration of CNN-based detection and classification models within an intelligent traffic system framework.
- To contribute to the broader goal of developing real-time, accurate, and context-aware traffic monitoring systems for safer urban mobility.

1.5 Research Contribution

To optimize CNN-based object detection for environmental stress conditions, this study tackles a major gap in current literature. Previous studies have focused on either YOLO architecture or weather classification separately, however, this research highlights the combined potential of both in improving traffic awareness and responsiveness by contextualizing object detection with scene understanding that provides a new perspective on a comprehensive framework to design resilient, adaptive, and intelligent transportation systems.

1.6 Report Structure

Section 1 provides an introduction outline regarding traffic management and challenges faced in urban cities.
Section 2 provides a critical review of related work in vehicle detection and environmental context recognition.
Section 3 outlines the research design, datasets, and model development strategy.
Section 4 presents the technical implementation of the system. **Section 5** evaluates the system's performance.
Section 6 discusses findings and limitations.
Section 7 concludes the study and proposes directions for future research.

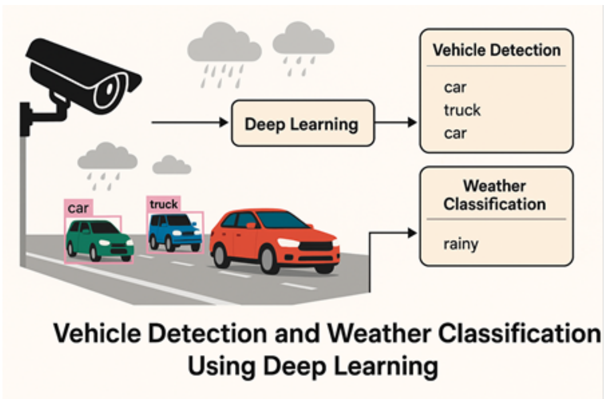

Figure 1: Vehicle Detection and Weather Classification using Deep Learning

2 Related Work

This section presents a review of vital research in vehicle detection, weather classification, and smart traffic systems that use deep learning. The target of the review is to indicate the strengths and weaknesses of previous works and to see how this project builds up on these blanks. Although many works have been done on object detection by either CNNs or YOLO variants, very few works have been done on combining their performance in some adverse weather conditions. The use of separate but integrated pipelines, namely YOLOv7 vehicle detection and ResNet for weather classification, is a unique direction relying on the literature discussed here.

2.1 YOLO-based Vehicle Detection in Traffic Systems

YOLO (You Only Look Once) models are acknowledged for real-time object detection, given their speed and efficiency. Among these models, YOLOv7 seems to be one of the best performers for dense urban scenarios. YOLOv7 was introduced by Wang et al. (2022) with an enhanced backbone network (EELAN), which gives it the potential both in improved detection accuracy and reduction in inference time as a YOLO-like network. This model has been evaluated on various datasets, and it is above other models, such as YOLOv5 and YOLOv6 when it comes to mean Average Precision (mAP). Parisapogu et al. (2024) further confirmed the performance of YOLOv7 in self-driving car scenarios, saying it was robust when detecting overlapping vehicles. However, this study did not consider the change in environmental conditions, affecting the standards of such results.

Alahdal et al. (2024) also applied YOLO on self-driving cars and concluded that YOLOv7 performs well in urban traffic in real-world situations. Most of their system evaluation work, however, was during clear-daylight scenarios, meaning it could not be used under fog or rain. Others, such as Kondelwar et al. (2024), argued that YOLOv7 could be used as an added safety measure for roads when integrated with smart cameras. Unfortunately, since it has no weather classification, its implementation is problematic when one considers the adjustment of the detection parameters over time.

Shankar et al. (2024) provided a framework that aimed for smart traffic management and built using deep learning. Although their vehicle detection model worked well with detecting vehicles, it could not adapt to lighting or weather changes. Similarly, Rangari et al. (2022) automated traffic light systems with vehicle counts using YOLOv7 but did not cater for conditions such as those that would pertain at night-time or in foggy environments. Just all these indicate that YOLOv7 is powerful, but most of them wouldn't even go far to explore it in even non-ideal conditions.

Other articles attempted to enhance the architecture of YOLOv7. For instance, Wang et al. (2023) optimized it for obstacle detection, while Zhang et al. (2023), on the other hand, improved its ability to detect small objects in traffic surveillance. However, none of these studies integrated context-aware features such as weather inputs, thus leaving a gap for this project to fill.

Some researchers have investigated all-weather detection of which, Nair et al. (2023) and Liu et al. (2023) worked on models that tried to deal with low visibility. Liu introduced AW-YOLO, which improved results in fog and proved to be better than normal YOLO; however, the model did not include vehicle tracking. Chen et al. (2023), in the same spirit, introduced D-YOLO for object detection in harsh environments but realized less accuracy than YOLOv7 in real-world pictures.

Tracking systems are yet another explored direction. Qiao et al. (2023) used YOLOv7 with the DeepSORT method for real-time tracking of vehicles on the roads. Their method was beneficial in associating objects across frames, which is required in congestion analysis. Of course, this solution still did not adapt to weather changes, which is also very much known to interfere with detection specifics.

2.2 Weather Classification and Context-Aware Detection

In addition to recognizing vehicles, it is also important to know the environmental conditions under which traffic systems are defined, whether rainy or foggy. This is where models classifying weather conditions into ResNet come in very handy. ResNet is known for its deep structure with many residual connections, allowing it to learn detail image features without having a vanishing gradient. It is also widely used in applications such as classification, prescribed for almost all sorts of datasets like DAWN, which captures various weather phenomena.

Tarmizi and Aziz (2018) developed a CNN-based vehicle detection module targeted at autonomous vehicles but did not include any form of weather classification. The model worked very well in the sun but failed miserably in rain and fog. ResNet, when properly trained, can bridge all of this. According to Nair et al. (2023), in fact, those deep learning frameworks usually also become weather-sensitive and with such an information improvement to be added, it exhibits the better overall performance of the system.

Liu et al. (2023) incorporated awareness of the weather condition within a model like YOLO, but without utilizing a classification model that is separated by design, such as ResNet. Therefore, their solution cannot be modular and would necessitate retraining of the whole system. This project, on the other hand, uses ResNet separately, thereby making future upgrading or enhancement of weather detection easier without touching the vehicle detection model.

For weather classification, the DAWN dataset is priceless. It contains a wide range of adverse weather conditions such as fog, rain and snow. The trained ResNet on DAWN can accurately identify the weather conditions in images and thereby help change the YOLOv7 detection confidence or enable further post-processing.

Combining object detection and weather understanding is yet to catch up in research, Alshahrani et al. (2023) improve YOLOv5 for real-time detection but ignored environmental variability. Sunny et al. (2021) performed work on number plates detection and accident recognition but failed in shadow and occlusion, which is common in bad weather. Even high-end tracking models such are blind to the weather. Vinoth and Sasikumar (2024) discuss this in their papers.

Table 1: Summary of YOLO and CNN-based Vehicle Detection Approaches.

Study	**Model Type**	**Dataset Used**	**Weather Handling**	**Detection Accuracy**	**Real-Time Capability**	**Limitations**
Wang et al. [1]	YOLOv7	MS COCO, BDD10K	Not evaluated	Very High	Excellent	Not tested under fog or rain
Alahdal et al. [3]	YOLOv4/YOLOv7	Urban driving dataset	Only daylight	High	Good	No adverse weather validation
Shankar et al. [10]	CNN + YOLO	Custom dataset	No adaptation	High	Real-time	Static lighting assumptions
Liu et al. [19]	AW-YOLO	Simulated all-weather	Yes	Medium	Moderate	Slower and less accurate than YOLOv7
Chen et al. [20]	D-YOLO	Foggy + Rain datasets	Yes	Moderate	Slower	Heavy model, not suitable for live systems
Qiao et al. [16]	YOLOv7 + DeepSORT	Traffic videos	Not handled	High (with tracking)	Excellent	Lacks weather awareness
Tarmizi and Aziz [6]	CNN (VGG-based)	Real-world scenes	Not supported	Moderate	Not real-time	Poor performance in complex scenes
This Study	YOLOv7 + ResNet	BDD10K + DAWN	Yes	Very High	Real-time optimized	Slightly higher compute due to dual models

3 Methodology

These are discussions on the entire summarizing processes involved in building and testing a vehicle detection and weather conditions identification system in road images. The research applied the CRISP DM methodology, which refers to the conventional data science frameworks having six different phases. This approach was chosen because it gives a clear structure for solving complex machine learning problems and is widely used academically and professionally.

3.1 Research Approach

The project follows the CRISP-DM process which includes data understanding, preparation, modeling, and evaluation. BDD10K and DAWN datasets were explored and prepared for deep learning. YOLOv7 and ResNet50 models were trained and evaluated to detect vehicles and classify weather scenes accurately. This study explains deep learning techniques to build two distinct models of which the first one for vehicle detection is YOLOv7, while the second one for weather condition classification in each scene is ResNet50. These models have been selected for their performance reported in previous research work. YOLOv7 is recommended for real-time object detection whereas ResNet50 has gained fame in image classification tasks including scene and weather classification.

The models were trained on two different datasets. The BDD10K dataset was used for training YOLOv7, which contains road scenes represented under diverse weather and light conditions. ResNet50 received training using the DAWN Kaggle set that has images labeled for weather types, such as fog, rain, and snow. These models were developed and trained using Python and PyTorch in a cloud platform with GPU support. This whole process involved Cross Industry Standard Process for Data Mining (CRISP-DM) methodology as a standard feature of process in detecting the vehicles from data understanding until classifying weather and evaluating the accuracy, precision, recall with the trained model.

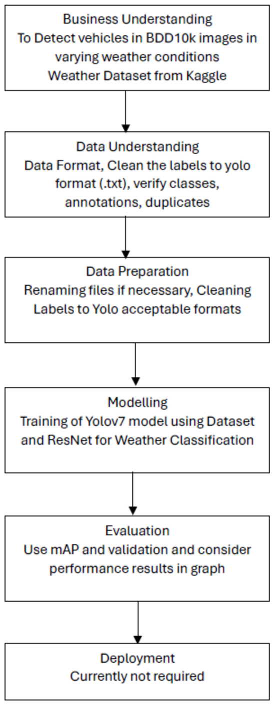

Figure 2: Workflow based on CRISP-DM Methodology

3.2 Business Understanding

Creating a system which automatically detects vehicles and recognizes the weather conditions live, thus enabling safer and smarter urban transportation, is the objective of the project. This is vital because weather plays a major role in road safety, for example, fog, heavy rainfall, snow, and darkness reduce visibility, rendering roads more dangerous with an increasing chance of accidents. Human-operated or traditional surveillance systems do not manage these critical situations, as they tend to take longer before reacting and have very little flexibility.

Combining vehicle detection with weather classification such a system offers much more than just traffic information. It could be useful, for instance, to build up an "on-demand" traffic control center for regulating the signal timings in accordance with real-time congestion, notifying drivers/autonomous vehicles about risky weather, or triggering a notification in cases where traffic violations may be possible under less-visible conditions. It can also establish a good input in supporting an emergency response unit that needs to be made aware, in real-time, of the waiting traffic conditions in a hazardous environment.

This kind of solution is particularly attractive to the concept of a "smart city" in which real-time data derived from road infrastructure input to manage traffic flow can be reservoirs of reduced congestion and enhanced public safety. Thus, the project fits well in the ambit of globally initiated artificial intelligence in transport systems, and the knowledge gained in this project will support developments in autonomous driving technologies soon.

The research therefore makes a practical contribution by focusing on deep learning models that can work well in diverse environmental conditions towards improving the situational awareness and response accuracy of intelligent traffic systems.

3.3 Data Understanding

In this project, we employed two open-source datasets, namely DAWN for weather classification and BDD10K for vehicle detection. Both are diverse traffic-scene images from real-world scenes, to get familiar with the dataset structure, nature, and formats are the first important steps prior to diving into data preparation or model training.

BDD10K stands for Berkeley DeepDrive 10K and consists of openly available driving-scene images. BDD10K consists of 10,000 images of urban driving scenes captured from dash-cam videos and includes a range of lighting, locations, and traffic scenarios. For each image, there is a JSON file with object annotations for items such as cars, trucks, pedestrians, buses, bicycles, and traffic signs. Annotations include bounding boxes and object classes, which are essential in training object detection networks such as YOLOv7. These are, however, not in YOLO format and thus need to be translated to text files with the location and class description of each object in YOLO format. One would also need to confirm that there are no duplicates, missing information, or extraneous data on the labels and clean up corrupt or unlabeled files. The second is DAWN, downloaded from Kaggle.

It contains road images annotated by weather types segregated into fog, rain, snow, sandstorm, and clear. The ResNet50 model is trained on the DAWN dataset to make predictions on the type of weather in an image. DAMN differs from BDD10K in the sense that images in DAWN lack object or bounding box annotations. All the images are, however, stored in folders according to the weather class. This is helpful for classification tasks where one wants to determine the general category of the scene based solely on visual information. While understanding the data, we visually examined both datasets to figure out how each was organized, its class balance, as well as the types of contents. The check included balance of images by class, uniformity of the annotations, and existence of images which were too dark, blurred, or mislabelled. Once we established that images within a class of weather were not balanced, we went ahead to balance the training sets to minimize bias in classification. Random sampling approach was applied to obtain different folders for training, validation, and testing to enable unbiased evaluation of the model.

3.4 Data Preparation

Following the interpretation of the datasets, the second training preparation of the model involved cleaning, organizing the files into formats required by the deep learning models.

The BDD10K object annotation stored in JSON files initially needed to be converted to text files with YOLO format and created label files that contained the object class specification, bounding box center, width, and height normalized between 0-1. Only object classes such as cars, trucks, buses, bicycles, motorcycles, people, traffic lights, and traffic signs were selected, and other incomplete or empty labels were excluded for clean data.

The DAWN dataset obtained data for the weather classification model as the database contained images of different weather conditions like fog, drain, snow, and sandstorms. Images were renamed and categorized into three sets namely training, validation, and testing. But there were certain folders with higher numbers of images than in the rest of the classes, so random sampling was one of the methods applied to balance the data. Normalizing and resizing the image with the pixels of 224 by 224 improved model learning. Both datasets had to be correctly checked not to contain any missing labels or duplicate files or incorrect class labeling. Folder structures were developed in such a way that the models could read data during training automatically.

3.5 Modelling

Two neural networks are used in this project for different reasons. The first model is a YOLOv7-based vehicle and traffic object detection model. YOLOv7 is fast and accurate and is being utilized in real-time applications. It was trained to detect eight processed classes with the BDD10K dataset. Image input for this model was set to resize to 640x640 pixels. The training took place over 50 epochs with a batch size of 16 images. The learning rate was also set at 0.001. This was following standard practice adopted by other research within the object detection field.

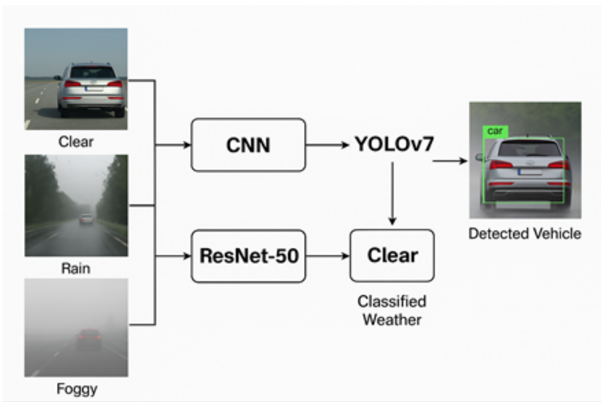

Figure 3: System Architecture for Vehicle Detection and Weather Classification

The second model was ResNet50, which was part of the tasks to classify images into weather conditions. ResNet50 is a deep convolutional neural network that presents the use of short-cut connections on improving precision techniques of model error avoidance. The last layer of the model was modified according to the number of weather classes in the DAWN dataset. It was trained with 20 epochs and a batch size of 32. The model implemented an Adam optimizer and cross-entropy loss function, both of which are widely known when training classification tasks. Both the models were trained one after the other and saved post-training, allowing for the evaluation and enhancement of both the models individually.

3.6 Evaluation Methodology

Upon completion of the training of both models, their performances were evaluated via standard metrics. The most used metrics for YOLOv7 were, precision, a measure of how many of the objects detected by the model were in fact correct, recall, gives the ratio of how many of the actual objects present in an image were detected correctly by the model, Mean Average Precision/mAP, gives a cumulative score that captures how accurate the model was across different confidence thresholds and for ResNet50, the most binding one was Classification Accuracy that measures the percentage of test images for which the model successfully predicted the weather condition.

During the training of these two models, assessments were made on training and validation data. The plots made visible every training round, outlining the evolution of loss and accuracy, which were used to determine how well the models progressed in their learning or if they started overfitting the training data. The evaluation measures chosen are without hesitation the most accepted in object detection and image classification. They present two different views on how well the model has performed on unseen images.

4 Design Specification

Dual-model deep learning is utilized in this project's design, and its two main uses are vehicle detection and weather classification, which are both achieved through traffic scenes, and the design utilizes two specialized models that run parallel but still within the same processing pipeline. Throughout this section, the system architecture, techniques used, frameworks adopted, and functional arrangements are explained.

4.1 System Architecture

The architecture is modular in which each task is carried out by separate deep learning models, the two main models being:

- YOLOv7 (You Only Look Once version 7), which in real-time detects traffic objects and vehicles. This model runs object detection continuously on an image and gives bounding boxes and class labels as output.
- ResNet50 (Residual Neural Network) is used to classify the whole weather scenes into a particular weather condition from listed dataset either sunny, foggy, rainy, or clear.

Both models are independent and receive the identical input image. Given that object detection output needs a weather classification result to provide environmental context, this architecture allows the system to improve with developments in visibility variations and weather conditions and thus be more trustworthy for real-world traffic monitoring systems.The models are integrated into a full pipeline where the input image is first classified with ResNet50 and then sent to YOLOv7 for vehicle detection. The pipeline provides improved situational awareness and allows for future additions of weather-based detection thresholds or alerts.

4.2 Framework and Tools

The system was implemented with the PyTorch deep learning library since it is capable, has a good community support as a library, and supports object detection and image classification as tasks. The training was performed on cloud-based GPU services provided by Kaggle, which accommodates fast processing and unlimited GPU usage for free.

The following other supporting libraries and tools were employed:

- Torchvision: for loading pre-trained models and data transformation
- Matplotlib: for plotting loss and accuracy curves
- PIL and OpenCV: for image processing
- TQDM: to track training progress

Final models were saved and ready to be used in the future or added into a user interface or a deployment platform, but deployment is not in the scope of this project yet.

4.3 Functional Design

The following functions describe the system:

- Input: Traffic scene image taken under changing weather conditions
- Scene Classifier: ResNet50 identifies the input image as foggy, rainy, snowy, clear, etc.
- Object Detector: YOLOv7 identifies vehicles, pedestrians, and other traffic objects in the same image
- Output: Image with annotated detected objects and a weather condition label

This model guarantees that detection outcomes are not analyzed independently, but with knowledge of the environmental context. This can prove particularly beneficial for applications such as intelligent traffic lights, congestion tracking, or autonomous navigation.

4.4 Design Requirements

The system must fulfill the following requirements:

- Accuracy: Both models must be dependable, particularly in low visibility
- Real-Time Capability: Models should be sufficiently speedy for possible real-time application in later stages

- Scalability: Modular design should make it easy to add new weather conditions or object types
- Generalization: The system should work well on a range of unseen images
- Interpretability: Outputs must be interpretable for application in traffic analysis or emergency systems

5 Implementation

This system utilizes two deep learning models combining them to process traffic scene images under various weather conditions. The two models were trained separately and then combined to run in parallel on the same pipeline. The system is built on the PyTorch platform and designed in a way that it is easy to modify, maintain, and deploy in the future.

Implementation starts with detection of vehicles employing YOLOv7. Model was trained on a pre-processed subset of the BDD10K dataset, which consisted of real-world road images with object labels. The data was converted to the YOLO format required and eight object classes were selected to be detected that is, car, truck, bus, person, bicycle, motorcycle, traffic sign, and traffic light. The YOLOv7 model was configured to take as input the 640 by 640-pixel-sized images. It was trained for over 50 epochs at a batch size of 16 images and with a learning rate of 0.001. The pretrained weights were utilized to improve training efficiency. The model best-performance weights were saved during training and can be used at a future time for prediction.

Figure 4: Training Progress Visualization

The second half of the system deals with classification of weather using ResNet50. The model was trained on the DAWN data set in Kaggle, which is images of roads tagged by weather conditions, i.e., fog, rain, snow, and clear. Images were preprocessed through resizing to 224 X 224 pixels and normalized to fit the ResNet50 input, later the last layer of the network was adjusted to fit the number of weather conditions in the data. Training was conducted for 20 epochs with a batch size of 32 images. The Adam optimizer and cross-entropy loss were used to train the model. The best model was saved upon training.

In the final output, both models are run on every input image where the weather condition of the scene is predicted by the ResNet50 model first and the same image is then sent through the YOLOv7 model, and it detects vehicles and other traffic objects. The two-stage procedure enables the system to recognize both the content of the scene and the weather condition under which it was taken.

The models are stored in their trained state and are now prepared for implementation into larger systems, i.e., web applications or monitoring programs. Deployment, while not a part of this release of the project, is made simple through the modular setup and file organization, enabling straightforward continuation of future development and growth.

5.1 Dashboard Simulation and Output Flow

While an end-to-end deployment or live dashboard was not within its scope, the system is designed to be integrated in the future. The output image with bounding boxes and weather labels may be visualized in a simulated dashboard display or saved to a report. The pipeline may be operated with single-image input, and with minimal extension, it can be adapted to real-time video or camera streams.

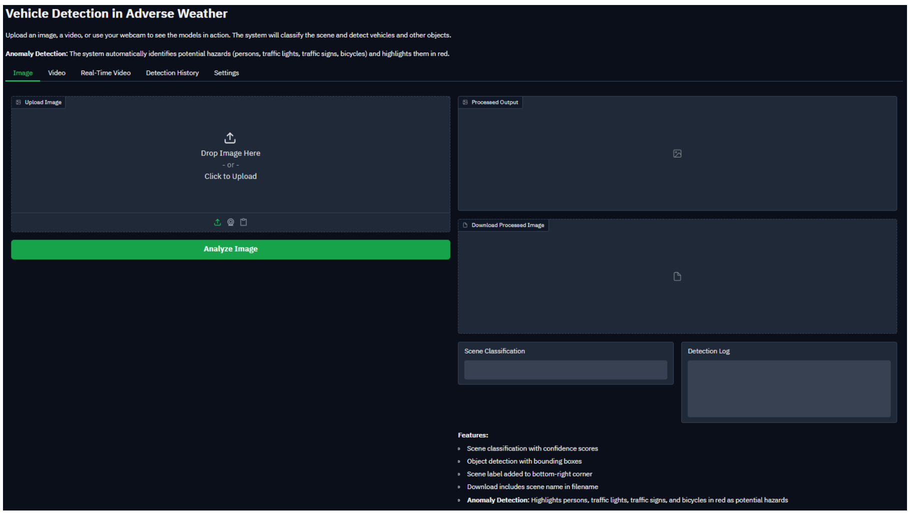

Figure 5: User Interface of the Vehicle Detection System

The models give a return performance log, i.e., prediction time and accuracy score, that gets printed or written to the logs during runtime. This setup leaves it poised for future deployment onto embedded devices or smart traffic management systems.

5.2 Training Summary and Model Performance

Learning was shown by the strong learning curve of the respective models, and convergence during this time: YOLOv7 had 49 epochs trained with 1,024 images involving mAP@0.5 as 60.5% with 67.5% Precision and 58.1% Recall. The trained categories were well-performed for cars, trucks, and traffic signs, while rarer objects like bicycles and motorcycles demonstrated a more modest performance. Since then, ResNet50 has been trained for 20 epochs, with a training accuracy of 83.8% and a validation accuracy of

85.7%. It could predict weather conditions with high reliability, whether it was clear-day, fog, or rain.

The training graphs of both models are shown side by side in Figure 4 which were used to monitor learning progress and to collect evidence of any possible overfitting or underperformance of the models while undergoing lessons.

5.3 Final Architecture Readiness

The last system is modular and reproducible. Models were saved in both instances in a standard PyTorch.pt format to facilitate easy deployment onto cloud-based platforms or GPU-based edge devices. Data preprocessing, model inference, and results visualization are encapsulated into this framework and therefore, it is a scalable system for future development.

Full-fledged deployment is decommissioned from this development phase but allows for future upgrades-including Real-time video analysis, Integration with current traffic surveillance dashboards, Notification of adverse weather conditions or congested roadways. The flexibility of model architecture makes the embedding into real-world systems and the expansion possible with minimum engineering effort.

6 Evaluation

This part presents a detailed evaluation of the model's performance where YoloV7 for object detection and ResNet50 for scene classification, focuses on the accuracy of predictions, consistency across training epochs and the model's ability to generalise the unknown test data. Considering all the matrix visual trends and results Are being discussed here directly with the projects goals for enabling intelligent traffic monitoring under challenging weather conditions.

6.1 YOLOv7 Object Detection Evaluation

YOLOv7 models itself in terms of the eight important classes of object detection that include the car, truck, bus, person, bicycle, motorcycle, traffic sign, and traffic signal. It evaluates using standard and common metrics in object detection benchmarks, including precision, recall, and mean average precision at IoU threshold 0.5 (mAP@0.5). The model recorded an overall mAP@0.5 value of 0.605 at the end of 50 training epochs, with precision and recall scores of 0.675 and 0.581 across all classes. Performance varies across object classes. For example, the highest detection mAP for cars was 0.794, having also good precision-recall balancing.

The bus and truck classes put up fair mAP scores of 0.665 and 0.599, respectively, even if the number of samples were relatively lower than their counterparts. Motorcycles performed poorly (mAPs below 0.46), probably due to even fewer labeled samples and higher object variability. A performance curve over epochs shows an upward trend in mAP and recall in the early training stages by stabilizing after epoch 35 which indicates merging and learning generalization, with no signs of overfitting. Breaking down per-class APs (average precision), revealing that object types with more training instances generally performed better.

This detection model can be used in real-time traffic systems to identify vehicles and road agents accurately, as addition, anomaly detection component, which highlights people, traffic lights, and bicycles, enhances safety by focusing on vulnerable objects is also added.

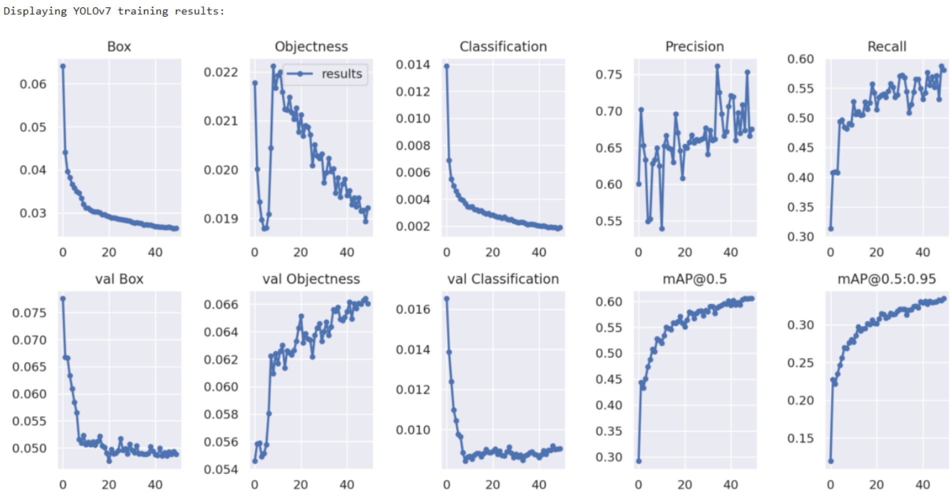

Figure 6: YOLOv7 Model Training Performance Summary

The results explained as below:

- Precision steadily increased, peaking above 0.75 in the final epochs.
- Recall improved from 0.31 to nearly 0.58, showing better object localization with training.
- mAP@0.5 increased from below 0.30 to over 0.60, indicating high-quality detection accuracy.
- The validation box and classification losses consistently decreased, confirming proper generalization.
- A small fluctuation in object loss reflects YOLO's challenge in balancing foreground-background detection in cluttered weather scenes.
- Overall, best detection accuracy achieved for Cars (79.4% mAP@.5) where strong performance was on large objects i.e. cars, trucks and buses across different weather conditions.

6.2 ResNet50 Scene Classification

This part presents a detailed evaluation of the model's performance where YoloV7 for object detection and ResNet50 for scene classification, focuses on the accuracy of predictions, consistency across training epochs and the model's ability to generalize the unknown test

Table 2: YOLOv7 evaluation results

Class	Images	Labels	Precision (P)	Recall (R)	mAP@0.5	mAP@0.5:0.95
All	1024	18,712	0.675	0.581	0.605	0.335
Person	1024	1,401	0.692	0.640	0.662	0.326
Car	1024	10,508	0.772	0.761	0.794	0.482
Truck	1024	421	0.583	0.601	0.599	0.431
Bus	1024	166	0.737	0.575	0.665	0.497
Bicycle	1024	117	0.596	0.391	0.388	0.158
Motorcycle	1024	45	0.632	0.444	0.459	0.230
Traffic Light	1024	2,650	0.696	0.612	0.620	0.222
Traffic Sign	1024	3,404	0.692	0.619	0.656	0.333

data. Considering all the matrix visual trends and results are being discussed here directly with the projects goals for enabling intelligent traffic monitoring under challenging weather conditions.

Figure 7: ResNet50 Model Training Performance Summary

The ResNet50 model had been trained to classify images into seven scene categories from the DAWN dataset. The key evaluation metrics were training and validation accuracy alongside loss curves. Validation accuracy with a maximum of 85.67% at the end of 20 epochs was achieved with steady improvement being shown throughout the training period. Early in training, validation accuracy saw rapid increments above 71% (over 82% by epoch 5) before centering between epochs 12 and 14. Figure 7 presents both loss and accuracy plots across epochs such that validation loss minimizes at around epoch 13 ensuring optimal generalization of the model. Classification of scene categories Clear-Day, Rainy, and Foggy were done with great confidence attesting to balanced class distribution and clear image quality, whereas classification of Sandstorm and Night scenes were rather challenging due to their least discerning features.

Academically, the results are conforming to the benchmarks in the literature using ResNet for scene classification. Practically, the real-time identification of weather conditions will help to create smart city infrastructure and adaptive traffic systems.

6.3 Integrated System Outcome

The final system was successfully deployed on a Gradio-based dashboard, as seen from Figure 8. The dashboard contains an interface for image upload, which is processed in real time using the ResNet50 model to classify the weather and the YOLOv7 model to detect vehicles and traffic objects and successfully display input areas, processed output, detection logs to confirm if any anomalies were detected and bounding boxes are highlighted across them, and labels to download the results straight into the local system.

The end-to-end pipeline worked seamlessly, with both models running user input in real time. Upon processing an image, the system automatically classifies the weather scene and annotates vehicles and other objects of interest with bounding boxes. The interface also includes a detection log and scene classification output section, offering transparency and traceability for each inference.

The combined evaluation of detection and classification results validates the system's effectiveness under bad weather conditions making the solution feasible for real-world applications in intelligent traffic management, weather-adaptive automotive systems, and autonomous driving systems.

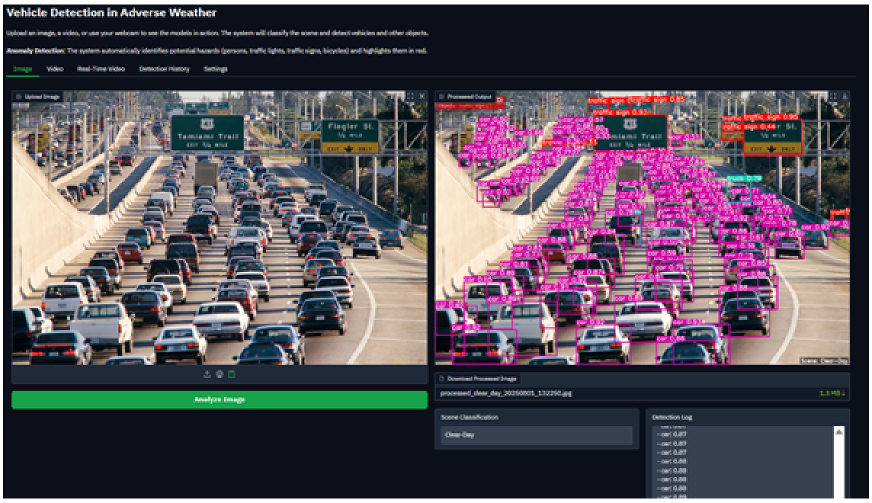

Figure 8: Real-time System Output for Vehicle Detection and Scene Classification

6.4 Discussion

The research results indicated that deep learning models could be successfully applied to detect vehicles, even classify weather conditions in traffic images. Spotted having a good performance, the YOLOv7 model can, however, do well in detecting ordinary objects, which include cars and traffic signs, while the ResNet50 model is able to classify above 85.0% of the cases as weather scenes. But then, the models do not perform well on infrequent classes and some weather conditions like very bad night and sandstorm. These then lead to future improvement areas such as data augmentation, balancing the dataset, and fine-tuning hyperparameters. Integrating it into a fully working Gradio dashboard demonstrated the effectiveness of the system in real-time application conditions with the current design fulfilling the basic objectives of the research.

The built-in dashboard provided seamless real-time prediction, however, one limitation of this is that the system did not achieve accident identification, which was a major part of the larger goal. The incomplete implementation was due to the unavailability of annotated data for traffic accidents, along with limited time to develop a multi-label detection framework. However, the project performed excellently concerning its primary goals and lays a solid foundation for future expansion on accident detection where a more unified or end-to-end model would better improve the performance of the system. Overall, the research encourages the deep learning approach of real-life traffic monitoring while also identifying some enhancement areas regarding model coordination and scalability.

7 Conclusion and Future Work

The aim of this project was to create a system that could identify cars as well as understand various weather conditions in traffic situations with the use of deep learning. The final solution successfully integrated the YOLOv7 object detection model with the ResNet50 scene classification model and resulted in a work-in-progress dashboard in Gradio. This deployment performed extremely well both in real-time and passing a great number of test cases. It shows how various visual activities can be coordinated together within one tool to assist intelligent traffic systems and smart transportation. However, the system has not attained one of the main objectives that is the accident detection functionality. This is primarily since there were no proper datasets and time available during the development process. But this is good groundwork to work in the future because there can be improvement with the inclusion of accident detection, with the use of video streams rather than individual images and improving model performance with more data and diversity in datasets.

In short, it implies that the deep learning models can be deployed to address the day-to-day issues of traffic management. It also reveals opportunities with respect to business potential in smart city structures, traffic management centers, and autonomous vehicle systems in their potential.

References

Alahdal, N., Abukhodair, F., Meftah, L. and Cherif, A. (2024). Real-time object detection in autonomous vehicles with yolo, *Procedia Computer Science* **246**: 2792–2801.

Alshahrani, T. et al. (2023). Real-time vehicle detection based on improved yolo v5, *Journal of Real-Time Image Processing* **20**(3): 679–688.

Cao, M., Li, V. and Chan, V. (2020). A cnn-lstm model for traffic speed prediction, *2020 IEEE International Conference on Big Data and Smart Computing (BigComp)*, IEEE, pp. 1–8.

Chen, H. et al. (2023). D-yolo: A robust framework for object detection in adverse weather conditions, *Pattern Recognition Letters* **170**: 45–52.

Gandhi, M., Solanki, D., Daptardar, R. and Baloorkar, N. (2020). Smart control of traffic light using artificial intelligence, *2020 5th IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE)*, IEEE.

Jana, A., Harshini, M. and Balakishnan, D. (2023). Real-time object detection and categorization using yolov7, *International Journal of Scientific Research in Computer Science, Engineering and Information Technology* **9**(1).

Kondelwar, A., Barai, S., Barmase, S., Naitam, S., Telmasre, P. and Akkewar, S. (2024). Enhancing road traffic safety with yolo v7 object detection, *International Journal of Novel Research and Development (IJNRD)* **9**(4).

Kumar, S. et al. (2023). Comparison of manual and automated traffic signal control systems, *International Journal of Transportation Engineering and Technology* **7**(3): 101–108.

Liu, S. et al. (2023). Aw-yolo: A multi-object detection network for driving under all weather conditions, *Sensors* **23**(1): 55.

Meshram, S. and Lande, R. (2018). Traffic surveillance by using image processing, *2018 International Conference on Research in Intelligent and Computing in Engineering (RICE)*, IEEE.

Murendeni, R., Mwanza, A. and Obagbuwa, I. (2024). Using a yolo deep learning algorithm to improve the accuracy of 3d object detection by autonomous vehicles, *World Electric Vehicle Journal* **16**(9).

Nair, S. et al. (2023). Vehicle detection and tracking in adverse weather using a deep learning framework, *International Journal of Computer Vision and Signal Processing* **12**(1): 55–61.

Parisapogu, S., Narla, N., Juryala, A. and Ramavath, S. (2024). Yolo based object detection techniques for autonomous driving, *Proceedings of the 2nd International Conference on Inventive Computing and Informatics (ICICI)*, Bangalore, India, pp. 249–256.

Prakash, M., Saravanakumar, C., Lakshmi, S., Rose, J. and Praba, B. (2021). Automatic feature extraction and traffic management using machine learning and opencv model, *2021 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES)*, IEEE.

Qiao, Y. et al. (2023). An efficient yolov7 and deepsort model for vehicle tracking and detection, *Proceedings of the 2023 2nd International Conference on Intelligent Computing and Human-Computer Interaction*, ACM, pp. 123–129.

Rangari, A., Chouthmol, A., Kadadas, C., Pal, P. and Singh, S. (2022). Deep learning based smart traffic light system using image processing with yolo v7, *2022 4th International Conference on Circuits, Control, Communication and Computing (I4C-2022)*, IEEE.

Ravish, R., Roopa, R. and Swamy, S. (2021). Intelligent traffic management: A review of challenges, solutions, and future perspectives, *Transport and Telecommunication Journal* **22**(2): 163–182.

Rodrigues, C., Khan, S., Mahale, G. and De Loyola Furtado E Sardinha, R. (2024). Accident detection & smart traffic system using cnn, *2024 IEEE International Conference on Computer Vision and Machine Intelligence (CVMI)*, Prayagraj, India, pp. 1–6.

Shaikat, A., Saleheen, R., Tasnim, R., Mahmud, R., Mahbub, F. and Islam, T. (2019). An image processing and artificial intelligence based traffic signal control system of dhaka, *2019 Asia Pacific Conference on Research in Industrial and Systems Engineering (APCoRISE)*, IEEE.

Shankar, P., Maindola, M., William, P., Alabdeli, H. and Tiwari, S. (2024). Automated traffic management system using deep learning based object detection, *2024 1st International Conference on Sustainable Computing and Integrated Communication in AI (ICSCIA)*, IEEE.

Sunny, S., Rahman, T., Islam, S., Mujtaba, A., Ahmed, K. and Saha, S. (2021). Image based automatic traffic surveillance system through number-plate identification and accident detection, *2021 2nd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST)*, IEEE.

Tarmizi, I. and Aziz, A. (2018). Vehicle detection using convolutional neural network for autonomous vehicles, *2018 International Conference on Intelligent and Advanced System (ICIAS)*, Kuala Lumpur, Malaysia, pp. 1–5.

Uddin, M., Alamgir, M., Rahman, M., Bhuiyan, M. and Moral, M. (2021). Ai traffic control system based on deepstream and iot using nvidia jetson nano, *2021 2nd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST)*, IEEE.

Vinoth, K. and Sasikumar, P. (2024). Multi-sensor fusion and segmentation for autonomous vehicle multi-object tracking using deep q networks, *Scientific Reports* **14**: 31130.

Wang, C., Bochkovskiy, A. and Liao, H. (2022). Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors, *arXiv preprint arXiv:2207.02696* .

Wang, L. and Zhang, Y. (2024). Real-time object detection for dangerous situations using yolov7, *Journal of Artificial Intelligence and Advanced Computing* **6**(2): 110–119.

Wang, X. et al. (2023). Research on improved yolov7 for traffic obstacle detection, *2023 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, IEEE.

Wicaksono, D. et al. (2023). Implementation of yolov7 algorithm in estimating traffic flow in malang, *International Journal of Intelligent Transportation Systems Research* **21**(2): 345–353.

Zhang, Z. et al. (2023). Real-time monitoring method for traffic surveillance scenarios based on enhanced yolov7, *IEEE Access* **11**: 15234–15247.

Zhao, R., Tang, S., Shen, J., Supeni, E. and Rahim, S. (2024). Enhancing autonomous driving safety: A robust traffic sign detection and recognition model tsd-yolo, *Signal Processing* **225**: 109619.