Department of Information Technology

Faculty of Computing

The Islamia University of Bahawalpur

Software Requirement Specifications

Real-Time Vehicle Detection, Classification, and Counting System Using Computer Vision and Deep Learning

Submitted by

Abdul Majid Ashraf

F22BINFT1E02058

Submitted to

Sir Faisal Shehzad

Meeting Details

Sr No	Details	Date	Supervisor Signature

Summary

This project develops an ML-based real-time vehicle detection and classification system capable of identifying and counting vehicles from live video feeds. Using computer vision techniques and deep learning models such as YOLO, the system detects moving vehicles and classifies them into categories like cars, trucks, and buses. It also keeps a real-time count of each category, which can be used for traffic analysis and intelligent transport management. The system can operate on live CCTV footage or pre-recorded videos, enabling authorities to monitor traffic flow, detect congestion, and gather data for infrastructure planning.

Table of Contents

1. Introduction

This Software Requirements Specification (SRS) document outlines the functional and non-functional requirements, along with the use cases, for the real-time vehicle detection and classification system. The objective of this project is to develop an intelligent computer vision solution capable of detecting, identifying, and counting vehicles from live video streams. Using deep learning techniques and object detection models such as YOLO, the system will recognize multiple vehicle categories including cars, trucks, and buses while continuously updating real-time counts. The system processes both live CCTV footage and pre-recorded videos, providing accurate and efficient traffic monitoring capabilities. Once deployed, this model will assist users, including traffic authorities and analysts, by offering automated insights into traffic flow, congestion patterns, and vehicle distribution to support smarter transportation management and planning.

1.1. Purpose

The purpose of this project is to develop a real-time vehicle detection and classification system using advanced computer vision and deep learning techniques. The system will utilize a YOLO-based model to accurately detect vehicles from live video streams and categorize them into predefined classes such as cars, trucks, and buses. Once deployed, the system will continuously monitor traffic, count vehicles in real time, and provide actionable insights for traffic management and analysis. This solution aims to support authorities and organizations in enhancing road safety, improving traffic flow, and making data-driven decisions for transportation planning.

1.2. Scope

The scope of this project involves developing a real-time vehicle detection and classification system using live video feeds from CCTV cameras or recorded footage. The system will be built using a YOLO-based deep learning model to detect vehicles and categorize them into classes such as cars, trucks, and buses. Instead of training a model entirely from scratch, a pre-trained YOLO architecture will be fine-tuned on a relevant vehicle dataset to improve accuracy and reduce development time. The final system will be integrated with a computer vision pipeline using OpenCV, enabling real-time detection, classification, and counting of vehicles. The solution will provide continuous monitoring capabilities to support intelligent traffic management and data-driven decision-making.

1.3. Product Perspective

The developed real-time vehicle detection and classification module will function as a core component of a larger intelligent traffic monitoring and analysis system. It will serve as a computer vision–driven backend service that processes live CCTV or video feed to detect vehicles, classify them into categories such as cars, trucks, and buses, and maintain real-time counts. The module will integrate seamlessly with a web-based or dashboard-style front-end interface, where traffic authorities or system operators can view live detection overlays, traffic statistics, and analytical insights. Using a fine-tuned YOLO model and an OpenCV video processing pipeline, the system will operate continuously and reliably, providing accurate vehicle detection for traffic management. This module, once deployed, enhances automation, supports decision-making for infrastructure planning, and enables intelligent transportation monitoring.

1.4. User Characteristics

End-users, such as traffic authorities, city planners, or transportation analysts, will interact with the vehicle detection and classification system to monitor and analyze real-time traffic data. The system is designed to be user-friendly, allowing users with minimal technical expertise in computer vision or deep learning to operate it effectively. Users will be able to view live video feeds, vehicle detection overlays, and traffic counts through a simple dashboard or interface. The goal is to provide an intuitive tool that enables users to monitor traffic flow, detect congestion, and gather actionable insights for transportation management without requiring specialized programming or AI knowledge. Users are assumed to have access to devices capable of running the application and viewing video streams.

1.5. Proposed Technologies

The proposed project will leverage the following technologies:

1. ***Deep Learning (YOLO)**: YOLO (You Only Look Once) object detection model, using pre-trained weights (e.g., YOLOv8) fine-tuned for vehicle detection and classification.*
2. ***Opencv (CV2)**: For real-time video capture, frame processing, and integration of the detection model with live or recorded video feeds.*
3. ***Python**: Primary programming language, utilizing libraries such as PyTorch or TensorFlow for model implementation and OpenCV for computer vision pipelines.*
4. ***Web Interface**: Front-end interface for displaying live video streams, detection results, and vehicle counts to users.*

2. Requirements

The system will allow users to view live video streams from CCTV cameras or uploaded video files through a web interface, after which the deep learning model will detect and classify vehicles in real time. The system will identify vehicle types such as cars, trucks, and buses, and maintain a live count for each category. Users will be able to see detection results overlaid on video frames, including bounding boxes and labels for each detected vehicle. The system will handle frame extraction and preprocessing to ensure accurate model predictions. Additionally, the web interface will provide an interactive and user-friendly experience, allowing users to monitor multiple video feeds and receive real-time analytics without technical expertise. The model will be trained or fine-tuned on a comprehensive vehicle dataset to ensure high detection accuracy and generalization across different traffic scenarios.

2.1. Function Requirements

The functional requirements of the project define the core features and operations that the system must perform to achieve its objectives. These requirements focus on the key functions that will be implemented in the real-time vehicle detection system. The system will enable users to view live video streams or uploaded video files, which will then be analyzed by a YOLO-based deep learning model to detect and classify vehicles such as cars, trucks, and buses. Detection results, including bounding boxes, labels, and real-time counts for each category, will be presented to users through an intuitive and interactive interface. The system will handle video frame extraction and preprocessing to ensure accurate predictions. Additionally, the web interface will allow users to monitor multiple feeds, review analytics, and continuously receive updates without technical expertise. The model will be trained or fine-tuned on a comprehensive vehicle dataset to ensure robust performance and generalization across different traffic scenarios. These functional requirements form the backbone of the project, guiding development and ensuring all critical tasks are efficiently performed.

2.1.1. Live Video Feed

- **Id:** *FR001*
- **Title:** *Live Video Feed*
- **Description:** *The system shall allow users to stream live video from CCTV cameras or upload recorded video files for analysis.*

2.1.2. Vehicle Detection

- **Id:** *FR002*
- **Title:** *Vehicle Detection*
- **Description:** *The system shall detect vehicles in each video frame using a trained YOLO-based model.*

2.1.3. Vehicle Classification

- **Id:** *FR003*
- **Title:** *Vehicle Classification*
- **Description:** *The system shall classify detected vehicles into predefined categories such as cars, trucks, and buses.*

2.1.4. Display Detection Results

- **Id:** *FR004*
- **Title:** *Display Detection Results*
- **Description:** *The system shall overlay bounding boxes and labels on detected vehicles and display real-time counts per category.*

2.1.5. Frame Preprocessing

- **Id:** *FR005*
- **Title:** *Frame Preprocessing*
- **Description:** *The system shall preprocess video frames, including resizing, normalization, and format adjustments, to ensure accurate detection.*

2.1.6. User Interaction

- **Id:** *FR006*
- **Title:** *User Interaction*
- **Description:** *The system shall provide a user-friendly interface for viewing live detection results, vehicle counts, and analytics, with continuous updates.*

2.1.7. Model Training / Fine-Tuning

- **Id:** *FR007*
- **Title:** *Model Training / Fine-Tuning*
- **Description:** *The system shall allow training or fine-tuning of the YOLO-based model on labeled vehicle datasets to ensure accurate and generalized detection.*

2.2. Non-Functional Requirements

Non-functional requirements define the overall qualities and constraints of the system, focusing on how the system performs rather than what it performs. These requirements ensure that the real-time vehicle detection system is reliable, efficient, and user-friendly.

- ***Performance and Responsiveness:*** *The system shall process video streams and detect vehicles in real time with minimal latency, providing near-instantaneous detection and classification results.*

- ***Scalability:*** *The system shall handle multiple simultaneous video feeds without degradation in performance, ensuring reliable monitoring in larger traffic networks.*

- ***Accuracy and Reliability:*** *The detection and classification model shall maintain high accuracy across diverse traffic conditions, different lighting scenarios, and varying camera angles, producing consistent and trustworthy results.*

- ***Usability:*** *The system shall provide an intuitive interface that allows users, including traffic authorities or city planners, to easily monitor live traffic, view vehicle counts, and analyze detection results without requiring technical expertise.*

- ***Maintainability and Extensibility:*** *The system shall be designed to allow easy updates to the detection model, addition of new vehicle categories, or improvements to the interface with minimal disruption.*

- ***Security and Privacy:*** *The system shall ensure secure handling of video streams and any stored data, safeguarding user and organizational privacy.*

Overall, these non-functional requirements guide the design and deployment of the system to ensure a robust, efficient, and user-friendly experience suitable for real-time traffic monitoring and intelligent transportation management.

3. Use Cases and Flow of Processes

Use cases describe how users interact with a system to achieve specific goals, translating functional requirements into practical scenarios. In this project, the primary actors are end-users who monitor live video feeds or upload video files, and the system, which handles video processing, vehicle detection, classification, and result display. These use cases help the development team understand critical processes and allow stakeholders to visualize the system's operation in real-world traffic monitoring scenarios.

3.1. Live Video Feed

ID	UC001
Name	Live Video Feed
Requirement(s)	FR001
Actor(s)	User
Precondition	*The user has access to the web interface and a valid video source.*
Postcondition	*The video feed is ready for vehicle detection and classification.*
Basic Flow	1. *The user navigates to the video feed section of the web app.* 2. *The user either selects a recorded video file or connects a live CCTV feed.* 3. *The system validates the video format and accessibility.* 4. *The system confirms successful feed connection to the user.*

3.2. Vehicle Detection

ID	UC002
Name	Vehicle Detection
Requirement(s)	FR002,FR005
Actor(s)	User, System
Precondition	*The video feed is active and accessible.*
Postcondition	*Vehicles are detected in real-time in each video frame.*
Basic Flow	1. *The system preprocesses each video frame (resizing, normalization).* 2. *The YOLO-based deep learning model scans frames for vehicle objects.* 3. *The system identifies vehicle locations and applies bounding boxes.*

3.3. Vehicle Classification

ID	UC003
Name	Vehicle Classification
Requirement(s)	FR003, FR005
Actor(s)	User, System
Precondition	*Vehicles have been detected in the video frames.*
Postcondition	*Detected vehicles are classified into categories (car, truck, bus, etc.).*
Basic Flow	1. *The system analyzes detected objects using the trained YOLO model.* 2. *Each vehicle is classified into its respective category.* 3. *The system updates the classification results in real-time on the interface.*

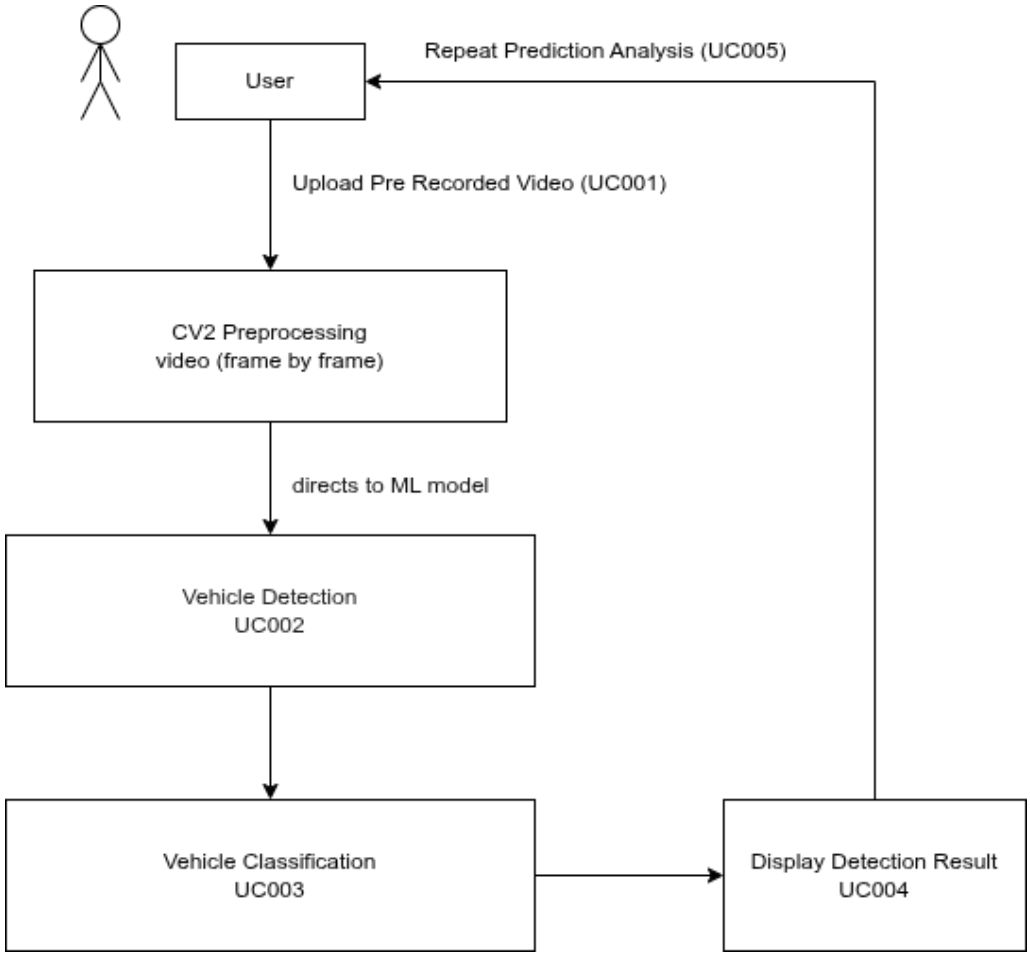
3.4. Display Detection Results and Analytics

ID	UC004
Name	Display Detection Results and Analytics
Requirement(s)	FR004, FR005
Actor(s)	User, System
Precondition	*Vehicles have been detected and classified.*
Postcondition	*Users can view live bounding boxes, labels, and vehicle counts.*
Basic Flow	1. *The system overlays bounding boxes and labels on the detected vehicles.* 2. *Vehicle counts per category are updated in real-time.* 3. *Users can monitor traffic flow and review analytics through the web interface.*

3.5. Repeat Monitoring

ID	UC005
Name	Repeat Monitoring
Requirement(s)	FR001 - FR006
Actor(s)	User, System
Precondition	*A video feed has been connected and processed.*
Postcondition	*Continuous detection and classification are maintained without restarting the system.*
Basic Flow	1. *The system continues to process each frame of the video feed in real-time.* 2. *Detection and classification results are continuously updated.* 3. *Users can switch between feeds or upload new videos while monitoring continues.*

4. UML diagram of workflow



5. References

1. Ultralytics YOLO models : *https://docs.ultralytics.com/modes/*
2. Majid Ashraf, ***"Real-Time Vehicle Detection, Classification, and Counting System Using Computer Vision and Deep Learning,"*** *GitHub Repository*. *https://github.com/abdulmajidashraf0/Real-Time-Vehicle-Detection-Classification-and-Counting-System-Using-Computer-Vision.git*
3. TensorFlow, ***"Keras Guide,"*** *TensorFlow Documentation*. https://www.tensorflow.org/guide/keras.
4. Streamlit, ***"Streamlit Documentation,"*** *Streamlit Web Tool* : https://docs.streamlit.io/.
5. Google, ***"Colaboratory,"*** *Google Colab* : https://colab.google/.