Project Title

Vehicle Classification and Counting Using AI

Submitted By

Khushboo Qureshi	**Roll No 26**
Areeba Gull	**Roll No 32**

BS Software Engineering (2021-2025)

Submitted to

Sir Sehrab Bashir

Date of Submission: June 30,2025

Department of Computer Sciences & Information Technology

Faculty: Basic & Applied Sciences

The University of Poonch Rawalakot

Contents

1. Introduction

The toll tax collection system in Rawalakot currently relies on entirely manual operations, where toll booth staff record vehicle counts and calculate taxes using pen and paper. This traditional method is prone to human error, inefficiency, revenue leakage, and fraud. Additionally, the lack of real-time monitoring and digital reporting makes it difficult to ensure transparency and effective oversight. This project proposes the development of an AI-powered automated toll tax system that leverages cameras. The system will process real-time data to detect, count, and classify vehicles accurately. Based on this classification, it will automatically compute the daily toll tax and generate digital reports, eliminating the need for manual record-keeping. By integrating modern AI techniques and automation, the proposed solution aims to enhance revenue collection, reduce fraud, and provide reliable traffic data for better decision-making and urban planning.

2. Objectives

The objectives define the clear and measurable goals that this project aims to achieve:

- **To develop** an accurate vehicle detection and classification model using advanced computer vision techniques that distinguishes between cars, trucks, motorcycles, and other vehicle types.
- **To automate** the toll tax calculation process based on classified vehicle types and predefined toll rates.
- **To create** a dynamic reporting dashboard that displays daily vehicle counts, toll tax collection data, and system performance metrics.
- **To integrate** the system seamlessly with the cameras for real-time and scalable deployment.
- **To ensure** transparency and fraud prevention through secure, tamper-proof digital records and AI-based evasion detection.

3. Problem Description

In Rawalakot, the toll tax collection process is currently managed through manual methods, where toll booth staff visually classify vehicles and record toll amounts using pen and paper. This traditional system lacks automation, relies heavily on human judgment.

This outdated process leads to several serious issues. Manual classification is prone to human error and manipulation, which results in inaccurate vehicle counts, revenue losses, and potential fraud. The absence of real-time monitoring means there is no immediate verification of toll transactions, and the lack of automated reporting makes data analysis and auditing extremely difficult. These inefficiencies reduce the reliability and transparency of the system and prevent city authorities from obtaining accurate traffic and financial data for planning and decision-making.

To address these challenges, there is a strong need for an AI-powered automated toll collection system. Such a system can utilize cameras to detect and classify vehicles in real time, calculate tolls automatically, and generate secure, tamper-proof digital reports. Implementing this solution will not only eliminate human error and fraud but also ensure transparency in revenue tracking and provide valuable traffic data for better urban management.

4. Proposed Solution

The current toll tax collection system in Rawalakot is inefficient, error-prone, and lacks transparency due to its reliance on manual vehicle classification and handwritten records. The absence of real-time monitoring and automated reporting further limits its effectiveness and opens the door to fraud and revenue leakage.

To address these issues, this project proposes the development of an AI-powered automated toll tax calculation system that integrates with the camera infrastructure. The core idea is to automate the entire toll collection process using real-time video analysis and machine learning techniques to ensure accurate and transparent operations.

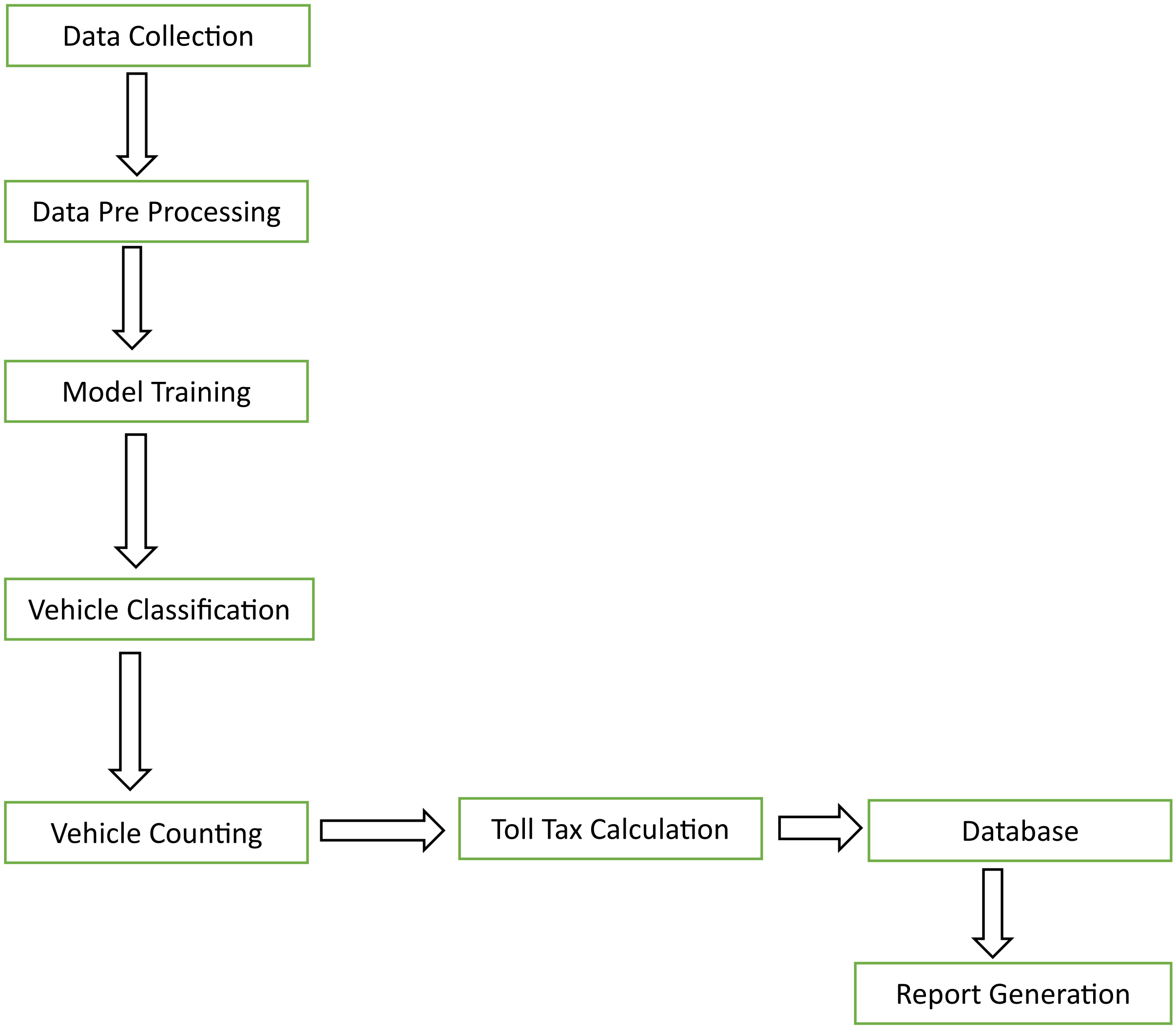
The key features of the proposed system include:

- Live video data collection from cameras.
- Data preprocessing to handle challenges such as poor lighting, camera obstructions, motion blur, and missing frames.
- AI-based vehicle detection and classification to differentiate between vehicle types (e.g., cars, trucks, motorcycles).
- Automated toll tax calculation based on vehicle type and predefined tax rates.
- Digital report generation showing daily vehicle counts, tolls collected, and system performance metrics.
- Fraud detection mechanisms using AI to flag toll evasion attempts.
- Secure, tamper-proof records for revenue tracking and auditing.
- Web-based dashboard for real-time monitoring and reporting.

The system will be developed using technologies such as:

- Python and OpenCV for video processing,
- TensorFlow or YOLO (You Only Look Once) for real-time object detection,
- Flask/Django for backend development,
- MySQL/PostgreSQL for database management,
- HTML/CSS/JavaScript for the user dashboard.

Flowchart:


Data Collection
Data Pre Processing
Model Training
Vehicle Classification
Vehicle Counting
Toll Tax Calculation
Database
Report Generation

5. Methodology

This project will follow the Agile development methodology, which is ideal for systems where requirements may evolve during development. Agile supports iterative progress, continuous feedback, and flexibility—making it suitable for real-time AI-based systems like automated toll tax collection. This approach allows for continuous improvement in vehicle detection accuracy, user interface design, and system integration as the project is tested in real-world conditions.

5.1. System Architecture

The proposed system will follow a modular and scalable architecture, consisting of the following key components:

- **Live Video Stream Handler**
  Collects real-time video footage from cameras.
- **Data Preprocessing Module**
  Cleans video frames by addressing issues such as poor lighting, motion blur, and obstructions.
- **AI Detection & Classification Engine**
  Uses machine learning algorithms to detect and classify different vehicle types (cars, trucks, motorcycles, etc.).
- **Toll Calculation Module**
  Calculates toll tax automatically based on the classified vehicle type and predefined tax rates.
- **Database Module**
  Stores vehicle logs, toll data, and daily reports securely.
- **Web Dashboard**
  Displays daily traffic statistics, toll amounts collected, and overall system performance for administrators.

5.2 Tools and Technologies

The following tools and technologies will be used:

- Python, OpenCV – For video processing and frame extraction
- YOLO / TensorFlow – For vehicle detection and classification
- Flask or Django – For backend API development
- MySQL / PostgreSQL – For managing data storage
- HTML, CSS, JavaScript – For developing the user dashboard interface

- Cameras – As the primary data source for real-time video streams.

5.3. Data Collection Methods

The system will source live video streams from cameras installed at toll collection points. These streams will be processed into image frames used to build and train the vehicle classification model. A combination of real-time footage and publicly available labeled datasets will be used to train the AI model for high accuracy and robustness.

5.4. Agile Sprint-Based Implementation Phases

The development will be carried out in eight Agile sprints, each focusing on specific deliverables:

- Sprint 1: Requirements Gathering & Planning
  - Identify core system requirements.
  - Define user stories and prioritize system features.
- Sprint 2: Initial System Design
  - Design the overall architecture and database schema.
  - Create wireframes/mockups for the dashboard UI.
- Sprint 3: Data Collection & Preprocessing
  - Capture and clean sample video frames.
  - Begin building and annotating the vehicle dataset.
- Sprint 4: Model Training & Testing
  - Train the vehicle classification model using YOLO or TensorFlow.
  - Test and refine model accuracy using collected data.
- Sprint 5: Backend & Toll Engine Development
  - Develop the toll tax calculation logic.
  - Connect model output to the backend and database.
- Sprint 6: Dashboard Integration
  - Build the frontend dashboard for traffic and tax visualization.
  - Implement user authentication and access roles (if needed).
- Sprint 7: System Testing & Feedback

    - Test the system using live video feeds.
    - Gather feedback and adjust model/UI based on results.
- Sprint 8: Final Deployment & Documentation
    - Deploy the complete system to a test environment.
    - Prepare user manuals and technical documentation.

6. Project Scope

The proposed **AI-Powered Toll Tax Automation System** is a real-time, web-based platform designed to automate the toll tax collection process using video feeds from the **camera network**. The system will detect and classify vehicles using artificial intelligence and calculate toll taxes based on predefined rates eliminating the need for manual vehicle counting and record-keeping.

The scope of this project includes:

- **Real-time video data capture** from cameras at toll points.
- **Automated vehicle detection and classification** using AI models (e.g., YOLO).
- **Toll tax calculation module** based on classified vehicle types and fixed tax rates.
- **Data preprocessing module** to handle video quality issues such as motion blur, poor lighting, and obstructions.
- **Secure database** to store vehicle logs, toll amounts, and daily traffic data.
- **Dynamic web-based dashboard** for administrators to view daily reports and monitor system status.
- **Tamper-proof digital reports** for transparency and audit purposes.
- **Scalable architecture** to support integration with camera infrastructure.

This system aims to enhance revenue tracking, reduce human error, and support better traffic analysis and urban planning through intelligent automation.

7. Brief Feasibility Study

The proposed AI-Powered Toll Tax Automation System is technically feasible by utilizing reliable technologies such as Python, OpenCV, and YOLOv5 for AI-based vehicle detection and classification, along with Flask/Django for backend development and MySQL/PostgreSQL for database management. The system will integrate with the camera network, providing a ready and stable source of real-time video data for analysis.

Operational feasibility is strong, as the system is designed to automate repetitive and error-prone tasks like manual vehicle counting and toll calculation. A user-friendly web dashboard will ensure easy monitoring, reporting, and minimal training requirements, promoting smooth adoption by toll and city administrators.

Economically, the project is cost-effective since it leverages open-source libraries and existing infrastructure, minimizing hardware and software expenses. In return, it offers high value through increased toll revenue accuracy, fraud prevention, and transparent reporting.

Legally, the system adheres to data privacy and surveillance laws, as it uses existing public video feeds without collecting personal information. All data will be securely stored and protected, ensuring ethical compliance.

The project timeline is realistic, with an estimated completion period of 3 to 4 months, supported by an Agile development approach and clearly defined milestones for smooth implementation and delivery.

8. Solution Application Area

The proposed AI-powered toll tax automation system is ideal for highways, urban toll plazas, and city entry points where accurate vehicle tracking and tax collection are needed. It helps reduce manual errors, prevent fraud, and improve revenue transparency. The system is especially useful for government agencies, transport departments, and Safe City projects looking to modernize toll operations. By automating vehicle detection and toll calculation, it ensures faster processing, better data insights, and efficient traffic management.

9. Functional Requirements

Function	Description
Admin Login	Allows authorized users to securely access the system.
Live Video Feed Access	Connects to cameras and fetches real-time video streams.
Video Frame Processing	Cleans and processes frames to handle issues like blur, lighting, and noise.
Vehicle Detection	Detects vehicles in video frames using AI-based models.
Vehicle Classification	Classifies detected vehicles (e.g., car, truck, motorcycle etc).
Toll Calculation	Automatically calculates toll tax based on vehicle type and predefined rates.
Data Storage	Saves vehicle logs, timestamps, and toll details in a secure database.
Report Generation	Creates daily, weekly, and monthly reports on traffic and toll collection.
Dashboard for Admins	Displays vehicle counts, toll revenue, and system status in a visual format.
Export Reports	Allows exporting reports in PDF or Excel formats.
System Monitoring	Tracks system health and shows alerts for missing feeds or errors.
Manual Review Support	Flags unclear or missed detections for manual verification.
Vehicle & Rate Management	Admins update toll rates or manage vehicle categories.
Admin Logout	Allows the admin to securely end their session and return to the login page.

10. Non-Functional Requirements

Requirement	Description
Performance	The system should process and classify vehicles in real time with minimal delay.
Accuracy	The vehicle detection and classification accuracy should be at least 80%.
Scalability	The system should be able to handle multiple camera feeds as the project expands.
Security	All toll data and user access should be securely protected using encryption and authentication.
Usability	The dashboard should be simple, intuitive, and easy to navigate for non-technical users.
Reliability	The system should run continuously with minimal downtime or errors.
Maintainability	The code should be modular and well-documented for easy maintenance and future updates.
Compatibility	The system should work with cameras infrastructure without needing major hardware changes.
Data Privacy	The system should not collect personal data and must comply with surveillance and privacy laws.
Availability	The system should be available 24/7 to ensure uninterrupted toll monitoring and reporting.

11. Tools and Technologies

The following tools and technologies will be used:

- Python, OpenCV – For video processing and frame extraction
- YOLO / TensorFlow – For vehicle detection and classification
- Flask or Django – For backend API development
- MySQL / PostgreSQL – For managing data storage
- HTML, CSS, JavaScript – For developing the user dashboard interface
- Safe City Rawalakot Cameras – As the primary data source for real-time video streams.

12. Expertise of the Team

Our team has a solid background in core areas of computer science, especially in Artificial Intelligence, Computer Vision, and Software Engineering. This academic and practical experience has equipped us with the knowledge to design and build intelligent, automated systems.

We are skilled in Python and AI libraries like YOLOv5, TensorFlow, and OpenCV, which are essential for vehicle detection and classification in real-time video streams. We also have experience in data preprocessing to handle issues such as motion blur, poor lighting, and frame noise.

On the web development side, we are proficient in front-end technologies like HTML, CSS, JavaScript, and Bootstrap, allowing us to build responsive and user-friendly dashboards. Our knowledge of Flask and Django helps us develop secure and efficient backend systems.

We are comfortable working with MySQL for designing structured databases and managing large amounts of toll and vehicle data. Additionally, we've gained hands-on experience with real-time applications, which prepares us to manage continuous video feeds and live data processing effectively.

Together, our technical skills and teamwork make us capable of developing a reliable, scalable, and AI-driven toll automation system.

13. Milestones

Milestone	Expected Completion Date
Requirement Analysis	Week 1–2
System Architecture & Design	Week 3–4
Data Collection & Preprocessing	Week 5–6
AI Model Development & Training	Week 7–8
Backend Development (Toll Engine & APIs)	Week 9–10
Frontend Dashboard Development	Week 11–12
System Integration & Testing	Week 13–14
Final Deployment & Documentation	Week 15–16

15. References

1. Bochkovskiy, Alexey, Chien-Yao Wang, and Hong-Yuan Mark Liao. *YOLOv4: Optimal Speed and Accuracy of Object Detection*. arXiv, 2020. https://arxiv.org/abs/2004.10934
2. Redmon, Joseph, and Ali Farhadi. *YOLOv3: An Incremental Improvement*. arXiv, 2018. https://pjreddie.com/media/files/papers/YOLOv3.pdf
3. *TensorFlow Documentation*. TensorFlow, https://www.tensorflow.org/
4. *OpenCV Documentation*. OpenCV, https://docs.opencv.org/
5. *Safe City Projects*. National Database and Registration Authority (NADRA), Government of Pakistan, https://www.nadra.gov.pk/safe-city-projects/
6. *Flask Documentation*. Flask, https://flask.palletsprojects.com/
7. *MySQL Documentation*. MySQL, Oracle Corporation, https://dev.mysql.com/doc/
8. *Toll Rates*. Pakistan National Highway Authority (NHA), http://nha.gov.pk/
9. Badrinarayanan, Vijay, Alex Kendall, and Roberto Cipolla. "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, 2017, pp. 2481–2495. https://doi.org/10.1109/TPAMI.2016.2644615
10. *YOLOv5 GitHub Repository*. Ultralytics, https://github.com/ultralytics/yolov5