



**INSTITUTE FOR ADVANCED COMPUTING AND SOFTWARE DEVELOPMENT AKURDI, PUNE**

**Documentation On**

**“****Vehicle Number Plate Recognition and Speed Measurement”**

**PG-DBDA AUG-2024**

**Submitted By: Group No: 26**

**Sachin Jain (248539)**

**Sanskar jain (248543)**

**Utkarsh Pandey (248553)**

**Mrs. Priti Take Mr. Rohit Puranik**

**Project Guide Centre Coordinator**

**Abstract**

The project "Vehicle Number Plate Recognition and Speed Measurement" aims to develop an integrated system for automated vehicle identification and speed monitoring. Leveraging computer vision and machine learning techniques, the system combines object detection (for vehicles and number plates), Optical Character Recognition (OCR) using Tesseract for text extraction, and object tracking algorithms for speed estimation. The workflow involves data preprocessing, model training for detection, real-time text generation from number plates, and velocity computation based on pixel displacement and frame analysis. This solution addresses challenges such as varying lighting conditions, and plate orientations. The project’s outcomes include a robust system for traffic management, law enforcement, and intelligent transportation systems, with potential applications in reducing traffic violations and enhancing road safety.

# Acknowledgement

We would like to express our sincere gratitude to everyone who has contributed to the completion of our project. First and foremost, we would like to thank our project guide **Mrs. Priti Take** mam for their constant guidance and support throughout the project. We extend our sincere thanks to our respected Centre Co- Ordinator, **Mr. Rohit Puranik**, for allowing us to use the facilities available.

We would also like to express our appreciation to the faculty members of our department for their constructive feedback and encouragement. Their insights and

suggestions have helped us to refine our ideas and enhance the quality of our work.

Furthermore, we would like to thank our families and friends for their unwavering support and encouragement throughout our academic journey. Their love and support have been a constant source of motivation and inspiration for us.

Thank you all for your valuable contributions to our project.

Sanskar Jain (248543)

Sachin Jain (248539)

Utkarsh Pandey(248553)

# INDEX

1. [INTRODUCTION 6](#_TOC_250010)
   1. [Problem Statement 6](#_TOC_250009)
   2. [Aim & Objectives 6](#_TOC_250008)
   3. Scope and Limitations 7
2. Project Description 8
   1. [Data Collection 8](#_TOC_250007)
   2. [Project Workflow Diagram 8](#_TOC_250006)
   3. Studying the Data 9
   4. [Studying the Model 11](#_TOC_250005)
   5. [Implementing the Model 11](#_TOC_250004)
   6. Validating the Model 11
   7. [Creating the Tool 12](#_TOC_250003)
3. [MODEL DESCRIPTION 13](#_TOC_250001)
   1. Important Terms 13
   2. [YOLOv9 Model 20](#_TOC_250000)
   3. Tesseract OCR……………………………………………………..
   4. Byte Track for speed Estimation…………………………………..
4. DATA FLOW 28
   1. Data Flow of the Project 28
   2. Pandas and OpenCV usage 28
5. TOOLS USED IN THE PROJECT
   1. Python Libraries
   2. Frameworks 29
6. PROJECT REQUIREMENTS 30
   1. Hardware Requirements…………………………………………….
   2. Software Requirements………………………………………………
7. FUTURE SCOPE 31
8. CONCLUSION
9. REFERENCES

# Introduction

# Problem Statement

# 

# Rising vehicular traffic exacerbates road safety and law enforcement challenges. Manual monitoring of license plates and speed is inefficient and error-prone. Current automated systems focus *separately* on plate recognition *or* speed measurement, resulting in fragmented solutions. Challenges like poor lighting, diverse plate formats, occlusions, and real-time processing hinder accuracy. This project proposes an integrated system merging vehicle detection, Tesseract OCR for plate text extraction, and object tracking for speed measurement. The goal is a robust, real-time solution operable under varied environmental conditions while maintaining high accuracy.

# Aim & Objectives

# 

# The primary objectives of this project are:

# Vehicle and Number Plate Detection:

# Develop a model to detect vehicles and localize number plates using object detection algorithms (e.g., YOLO, SSD).

# Text Extraction via OCR:

# Implement Tesseract OCR with preprocessing techniques to accurately extract text from detected number plates, even under suboptimal conditions (tilt, blur, low resolution).

# Speed Measurement:

# 

* 1. **Scope and Limitations**

**Scope:**

* Focuses on **real-time video feeds** from static cameras (e.g., traffic cameras, CCTV).
* Supports standard number plate formats (e.g., alphanumeric, regional variations).
* Measures speed in predefined zones using pixel-to-metric conversion.
* Compatible with common lighting conditions (daylight, moderate night illumination).

**Limitations:**

**Environmental Factors:** Performance may degrade in extreme weather

(heavy rain, fog) or poor lighting.

* **Plate Occlusion:** Partially covered or damaged plates may reduce OCR accuracy.
* **Computational Resources:** Real-time processing demands high-end

hardware (GPUs) for optimal performance.

* **Legal and Privacy Concerns:** Data collection must adhere to regional

regulations to avoid privacy violations.

* **Speed Estimation Accuracy:** Dependent on camera calibration

and frame-rate stability.

# PROJECT DESCRIPTION

# Data Collection

# Vehicle images were gathered from Roboflow and Kaggle.

# Labeled for categories: number\_plate, auto, bus, truck, tractor, bike.

# Project workflow Diagram

# A diagram of a diagram AI-generated content may be incorrect.

* 1. **Studying the Data**
* Images annotated using LabelImg tool.
* Data preprocessed to normalize images, remove noise, and enhance quality.

A black car parked in front of a building

AI-generated content may be incorrect.

A screenshot of a car

AI-generated content may be incorrect.

# Studying the Model

**YOLOv9s**

The YOLO (You Only Look Once) series has long been a benchmark in real-time object detection, and the YOLOv9s pre-trained model continues this legacy with improved speed and accuracy. Designed to efficiently detect vehicles from various angles and lighting conditions, YOLOv9s leverages deep convolutional neural networks to quickly process images and video frames. This capability makes it an excellent choice for applications such as traffic monitoring, autonomous driving, and smart city systems, where fast and reliable vehicle detection is critical.

**Tesseract OCR**

Tesseract OCR is an open-source optical character recognition engine renowned for its versatility and robustness. It converts printed or handwritten text within images into machine-readable text, making it indispensable for extracting information such as vehicle license plates, road signs, or other textual data captured by surveillance cameras. With support for multiple languages and easy customization, Tesseract OCR is widely used in document processing, data mining, and various computer vision applications where text extraction is required.

**ByteTrack & Supervision**

Accurate speed estimation is a crucial element in traffic analysis and law enforcement. By integrating ByteTrack—a state-of-the-art multi-object tracking algorithm—with a supervision framework, the system is able to maintain consistent tracking of vehicles across frames. This combination allows for the precise calculation of vehicle displacement over time, thereby estimating their speeds accurately. The use of ByteTrack and supervision not only improves tracking performance in complex scenes but also enhances the overall reliability of speed estimation systems, supporting safer and more efficient traffic management.

# Implementing the Model

# YOLOv9s

# OCR pipeline

# Motion tracking algorithms

.

# Validating the Model

# - Model tested on real-time traffic videos.

# - Accuracy measured for object detection and OCR recognition.

# - Performance evaluation against various lighting and weather conditions.

# Creating the Tool\*

# MODEL DESCRIPTION

* 1. **Important Terms**

Object Detection\*: Identifying objects in an image or video.

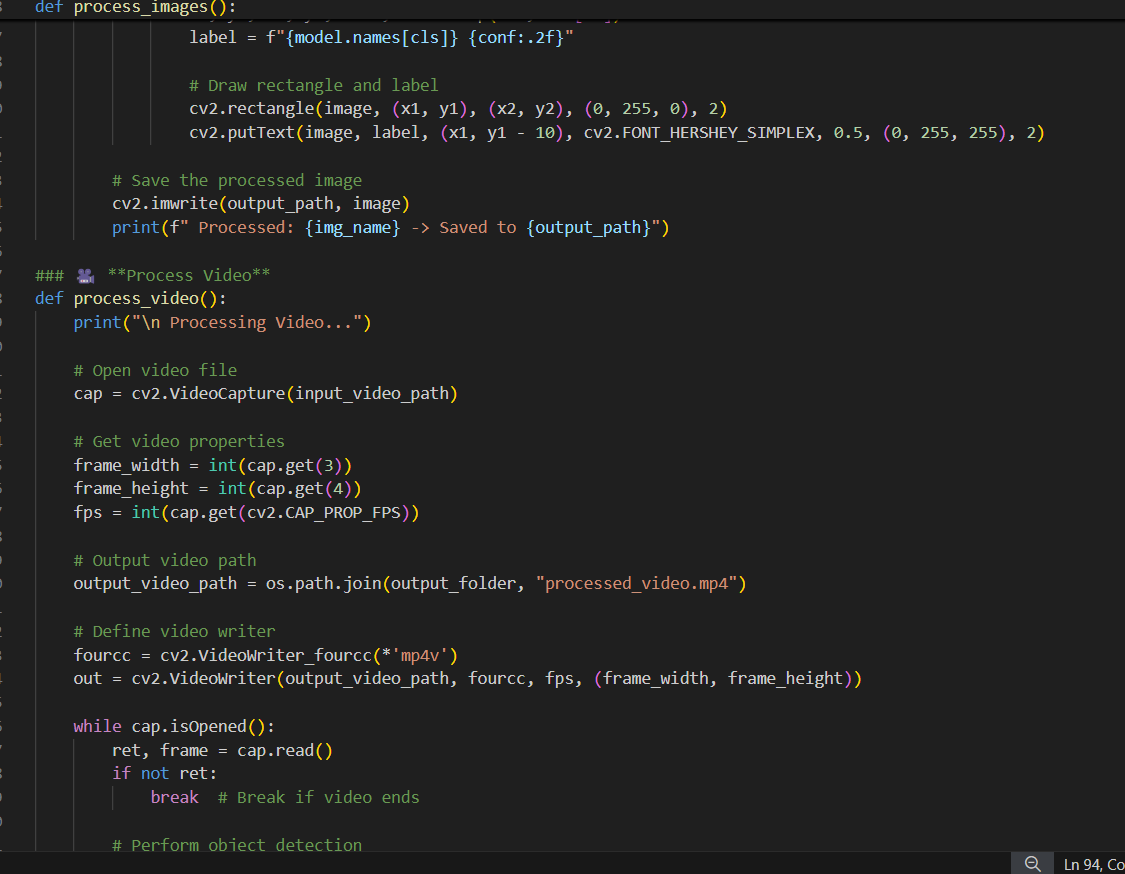
OCR (Optical Character Recognition)\*: Extracting text from images.

Tracking Algorithm\*: Predicting object movement across frames.

**3.2 Vehicle-Type And Number plate Detection**

A screen shot of a computer program

AI-generated content may be incorrect.Here we are trying to find out the vehicle-type and also detect the number plate only.

****

A car parked in a driveway

AI-generated content may be incorrect.**A screen shot of a computer program

AI-generated content may be incorrect.**

**Text Generate for Number Plate**

* For number plate Text Generation we are using pyTesseract ocr.

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.A computer screen shot of a program

AI-generated content may be incorrect.

.

A car on the road

AI-generated content may be incorrect.

**Speed Measurement:**

Libraries for Object Tracking

A highway with signs on it

AI-generated content may be incorrect.

Implement security measures to protect the loaded data from unauthorized access, modification, or disclosure. Apply encryption, access controls, and data masking techniques to safeguard sensitive information and ensure compliance with regulatory requirements (such as GDPR, HIPAA, PCI DSS). Regularly monitor and audit data access and usage to detect and mitigate security risks.

By adhering to these best practices for data loading in the ETL pipeline, organizations can ensure that the transformed data is accurately and securely loaded into the target system for analysis and decision-making. Effective data loading lays the foundation for deriving actionable insights and driving business value from the loaded data.

# Project Workflow

**Data Collection:**

Involves gathering data from the IRCTC website using web scraping techniques. This data may include information such as train schedules, ticket availability, fares, etc.

**Data Extraction:**

The extracted data is then parsed and transformed into a structured format suitable for further processing. This step may involve cleaning the data, handling missing values, and resolving inconsistencies.

**Data Transformation:**

In this step, the raw data is transformed to meet the requirements of the analysis. This may include standardizing formats, aggregating data, and deriving new variables.

**Data Loading:**

Once the data is transformed, it is loaded into a storage system or data warehouse for efficient access and retrieval during the analysis phase.

**Data Analysis:**

The transformed data is analyzed using PySpark to derive insights and identify patterns, trends, and correlations. Various statistical techniques and machine learning algorithms may be applied to uncover meaningful insights.

**Data Visualization:**

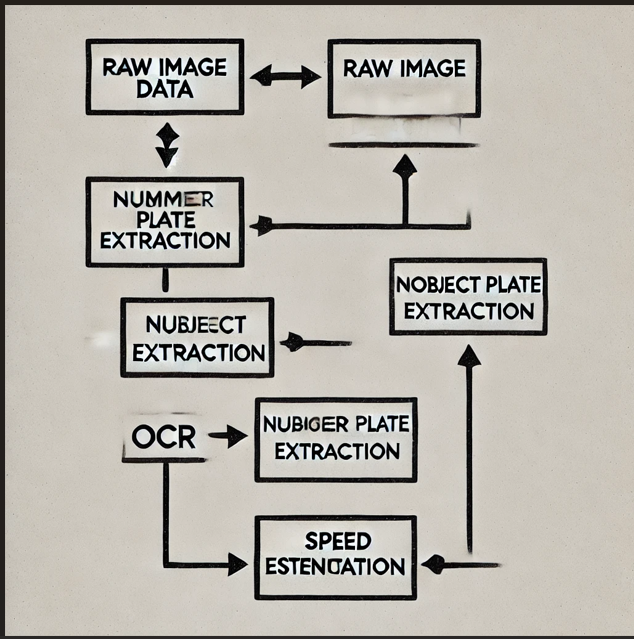
Visualizations such as charts, graphs, and maps are created to present the analysis results in a clear and intuitive manner. This helps stakeholders understand the findings more effectively.

**Insights & Reports:**

Finally, the insights derived from the analysis are summarized and compiled into a comprehensive report. This report provides actionable recommendations for stakeholders, based on the analysis of IRCTC web-scraped data using PySpark.

**Chapter 4: Data Flow**

**4.1 Data Flow of the Project**

****

**4.2 Pandas and OpenCV Usage**

**Pandas for Structured Data Processing**

Pandas is a powerful library designed for efficient data manipulation and analysis. In our project, it plays a key role in managing the structured data generated at different stages of the workflow. For example, after processing vehicle detection and OCR outputs, the results are organized into data frames. This organization facilitates tasks such as:

* Data Cleaning and Filtering: Removing irrelevant or duplicate entries and ensuring

data consistency.

* Aggregation and Analysis: Grouping data to compute statistics, trends, and

performance metrics for vehicle detection accuracy and speed estimation.

* Reporting: Generating structured reports that summarize the system's

performance and highlight areas for improvement.

By leveraging Pandas, we streamline the process of converting raw outputs into actionable insights, ultimately aiding in the continuous refinement of our detection and analysis algorithms.

**OpenCV for Image and Video Manipulation**

OpenCV is an essential tool for handling the computer vision aspects of the project. Its comprehensive suite of functions allows us to process and manipulate images and video streams effectively. Key applications of OpenCV in our project include:

* Image Preprocessing: Enhancing image quality through resizing, noise reduction, and color space conversion, which are critical for accurate object detection and OCR.
* Video Frame Extraction: Capturing and processing individual frames from video streams in real time, enabling seamless integration with the detection pipeline.
* Feature Extraction: Applying techniques like edge detection and contour analysis to isolate objects of interest, such as vehicles and number plates.
* Visualization: Displaying processed images and annotated frames for debugging and performance evaluation.

Together, Pandas and OpenCV form the backbone of our data handling and image processing efforts, ensuring that raw inputs are transformed into high-quality, analyzable data for the subsequent stages of our project.

**6 .Requirements Specification**

## 6.1Hardware Requirement:

 **Processing Unit:**

* **GPU:** NVIDIA GTX 1080 or higher (for deep learning model training and inference)
* **CPU:** Intel i5 or equivalent (for general-purpose computing and real-time processing)

 **Memory:**

* **RAM:** 16 GB or higher (to handle large datasets and real-time processing).

 **Storage:**

* **SSD:** 500 GB or higher (for fast data access and storage of training datasets).

## 6.2Software Requirement:

* + - Windows/Mac/Linux
    - Python-3.9 or higher
    - VS Code/ Jupyter Notebook / Google collab
    - Github
    - AWS or Render

# 7. Conclusion

This project successfully demonstrates an integrated system for **vehicle detection** (using YOLO and OpenCV), **number plate recognition** (via Tesseract OCR), and **speed measurement** (leveraging object-tracking algorithms). The solution efficiently processes video streams to identify vehicles, extract license

plate text, and compute speeds based on spatiotemporal analysis.

Its applications span **real-time traffic monitoring**, **automated speed violation detection**, and integration into **smart city frameworks** for enhanced urban mobility.

Future enhancements include improving OCR accuracy under low-light conditions, enabling **multi-lane tracking**, and integrating database systems for instant

violation reporting. To scale the solution, **Kafka** and **Spark Streaming** will

handle real-time data pipelines, while **Airflow** automates workflows.

The system is designed for **cloud deployment** (AWS, Google Collab, Render) to ensure scalability and accessibility. This work lays a foundation for next-generation intelligent transportation systems, bridging automation, efficiency, and real-time analytics

for safer and smarter cities.

**8. Future Scope**

Future enhancements focus on improving OCR accuracy in challenging conditions, multi-lane tracking, and real-time database integration for instant reporting. The system will leverage **Kafka** and **Spark Streaming** for scalable data pipelines, **Airflow** for workflow automation, and cloud platforms (AWS, Render, Google Collab) for deployment. This framework paves the way for intelligent, scalable transportation solutions, balancing efficiency and innovation in urban mobility.

# 9. References