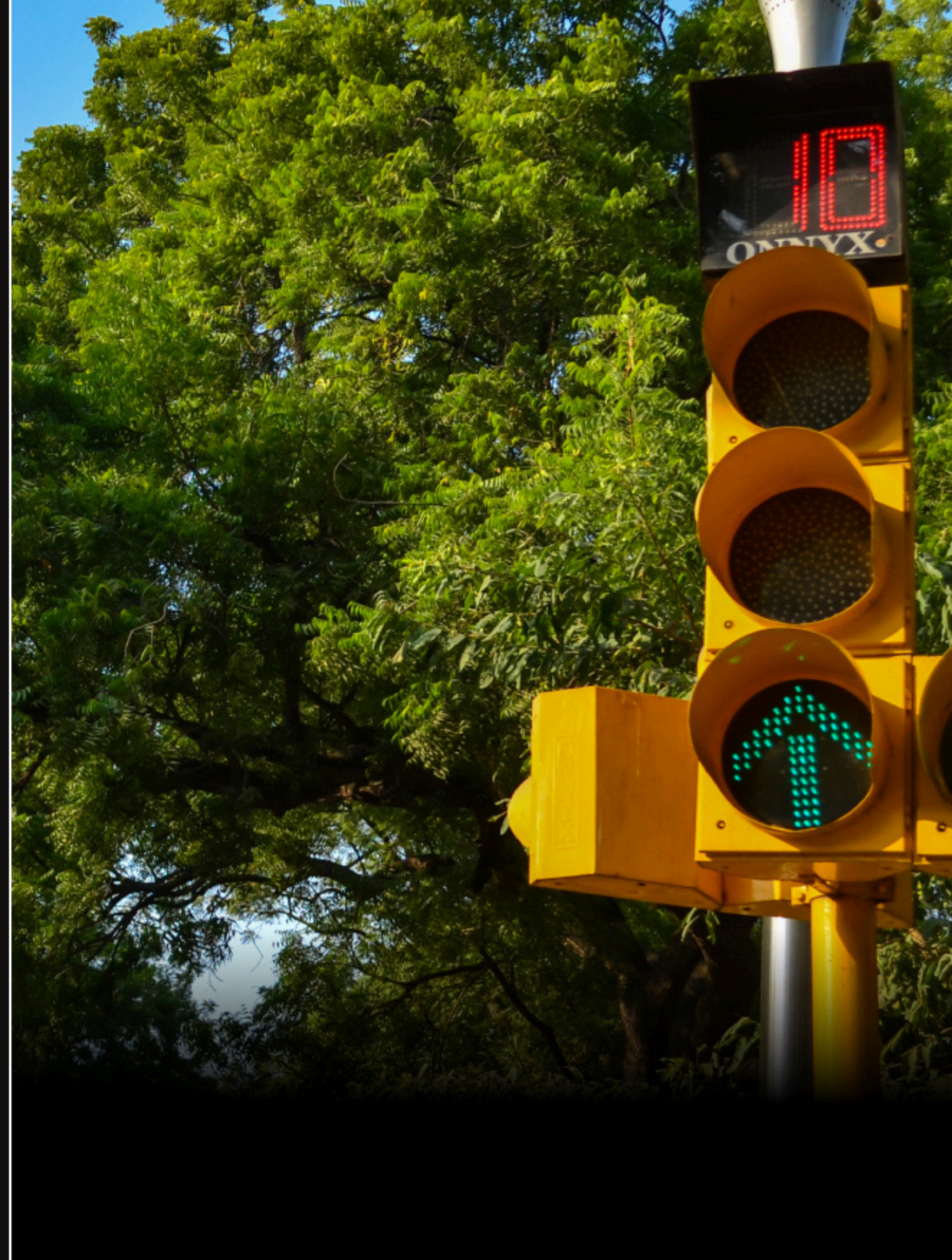
ADAPTIVE DYNAMIC TRAFFIC SIGNAL CONTROL FOR INDIAN ROADS

BY :- ADITYA PANDEY, TO PROF. AMIT MANKODI



INTRODUCTION

Overview:

- Traffic congestion is a critical issue in urban areas.
- Traditional signal systems use fixed timers, leading to inefficiency.
- Our project aims to dynamically estimate traffic signal clearance time by detecting and classifying vehicles from real-world images.

Key Goals:

- Detect and count vehicles using YOLOv8.
- Classify Indian vehicles using a custom-trained CNN model.
- Estimate clearance time based on vehicle types and counts.
- Optimize signal timing to improve real-world traffic flow.

PROBLEM STATEMENT

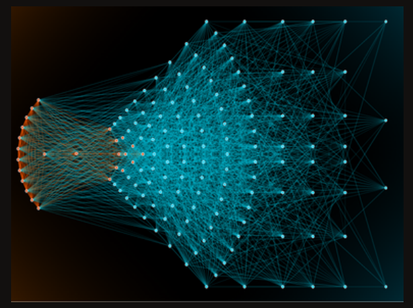
🚦 The Challenge

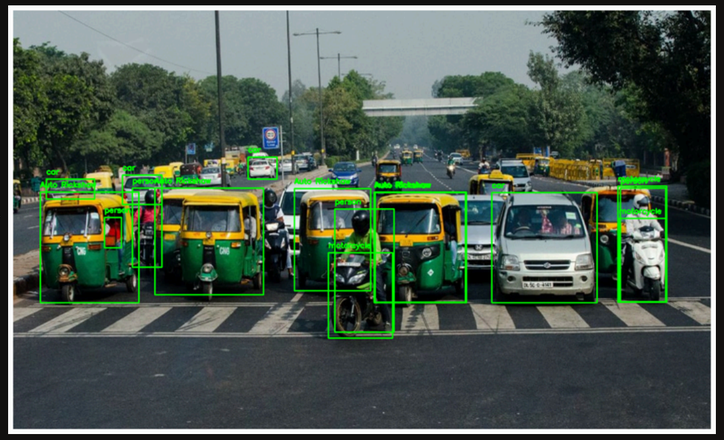
Urban intersections often face irregular traffic flow. A fixed signal timer doesn't adapt to real-time congestion, leading to:

- Unnecessary waiting during low traffic.
- Insufficient time for vehicles to clear during high traffic.

METHODOLOGY

YOLO





JSON

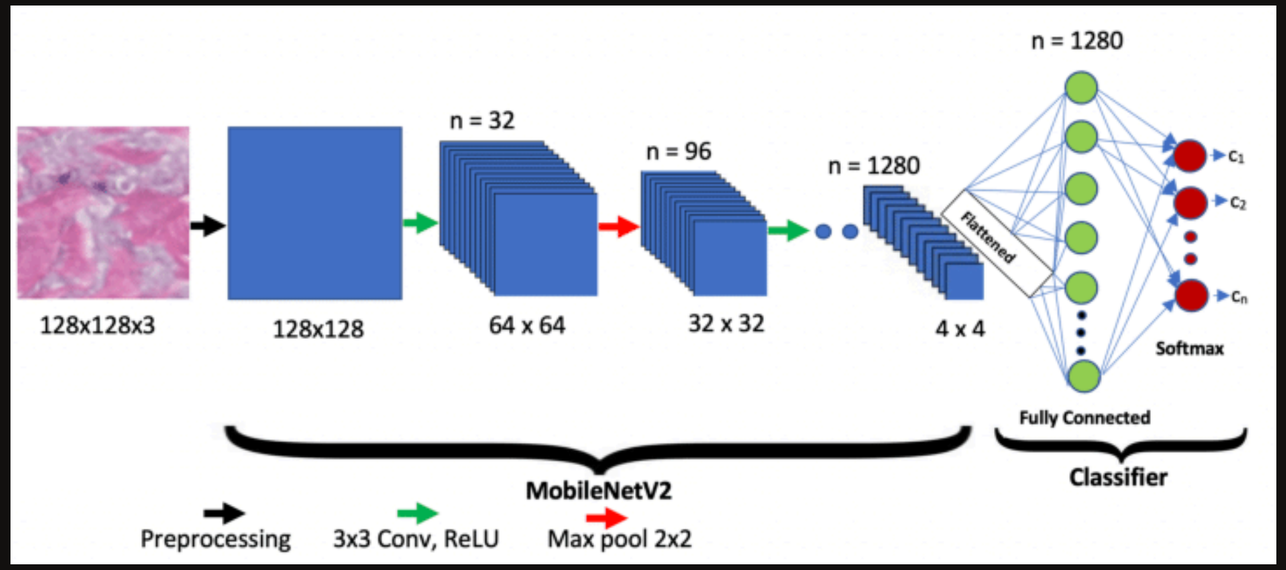
STD

```
Skipping unknown category: person

⏱️ Estimated Time for Each Category to Pass Signal (seconds):
Auto Rickshaw: 12.38 seconds
car: 9.36 seconds
motorcycle: 4.05 seconds

🟢 Total Estimated Time for All Vehicles to Pass: 25.79 seconds
```

```
🟩 Final Vehicle Count (YOLO-priority):
Auto Rickshaw: 5
person: 4
car: 4
motorcycle: 3
```

TECHNIQUES USED

- YOLOv8 (Priority)
- CNN(MobileNetV2)
- Transfer Learning
- Time Estimation
- Traffic-Signal Optimization



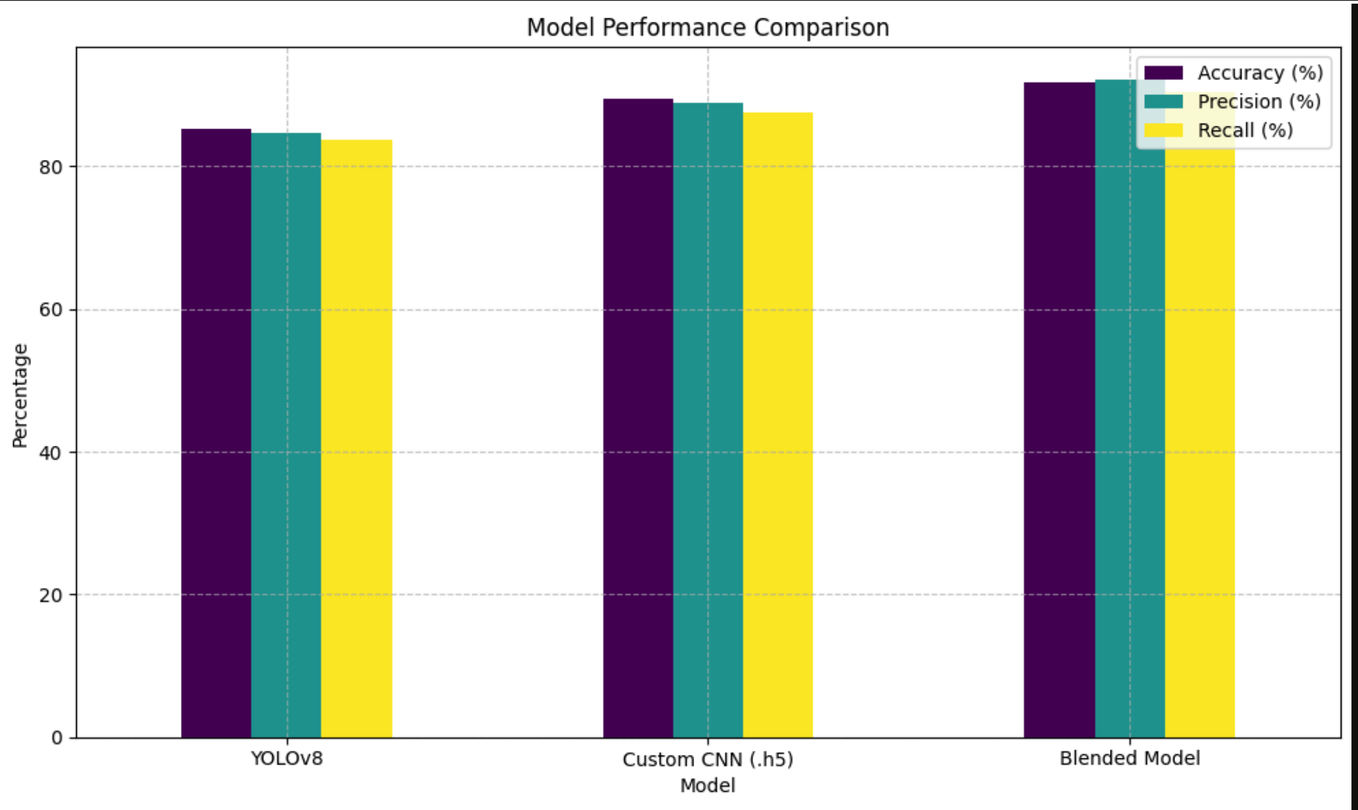
+

TABLE I: Top Layers Added on MobileNetV2 for Indian Vehicle Classification

Layer Type	Details	Activation	Output Shape
GlobalAveragePooling2D	-	-	(None, 1280)
Dropout	rate = 0.4	-	(None, 1280)
Dense	128 units	ReLU	(None, 128)
Dropout	rate = 0.3	-	(None, 128)
Dense (Output)	4 units	Softmax	(None, 4)

MODEL COMPARISIONS

```
=== Model Comparison Table ===
          Model  Accuracy (%)  Inference Time (s)  Precision (%)  Recall (%)  F1-Score (%)  Avg Time to Clear Signal (s)
         YOLOv8          85.2                0.12           84.6        83.8          84.2                          21.3
Custom CNN (.h5)          89.4                0.08           88.9        87.5          88.2                          19.7
   Blended Model          91.7                0.15           92.1        90.4          91.2                          18.9
```

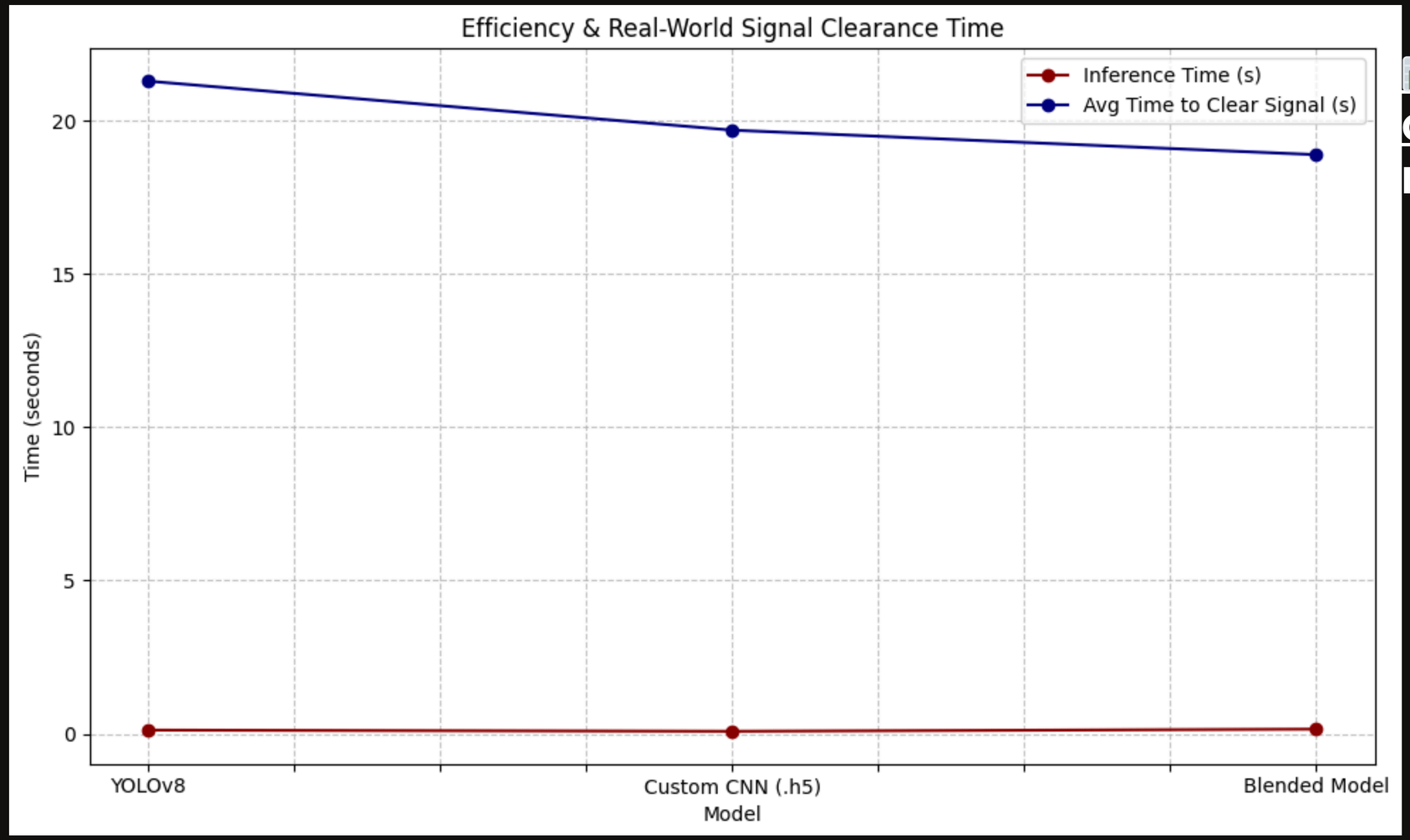


📋 2. Model Comparison Table

Meaning:

- A summary table showing the performance of different deep learning models (e.g., CNN(MobileNetV2),YOLOv8,Blended Model) on the same dataset.

GRAPHS & PLOTS

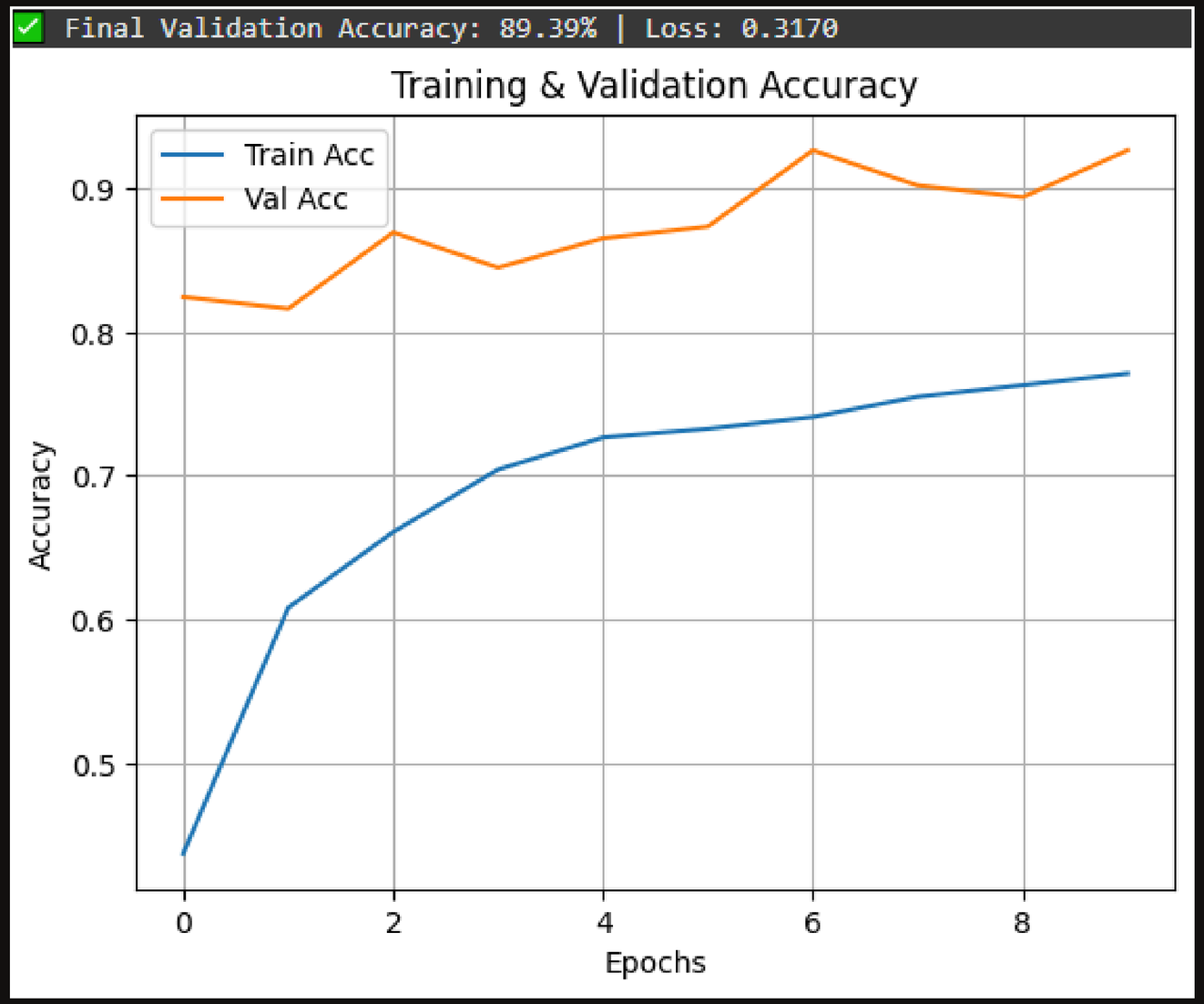


📊 1. Efficiency & Real-World Signal Clearance Time

Meaning:

- Efficiency refers to how well the model performs in detecting vehicles and estimating the time needed to clear a traffic signal.
- Real-World Signal Clearance Time is the estimated duration (in seconds) required for all detected vehicles to safely pass through an intersection when the signal turns green.

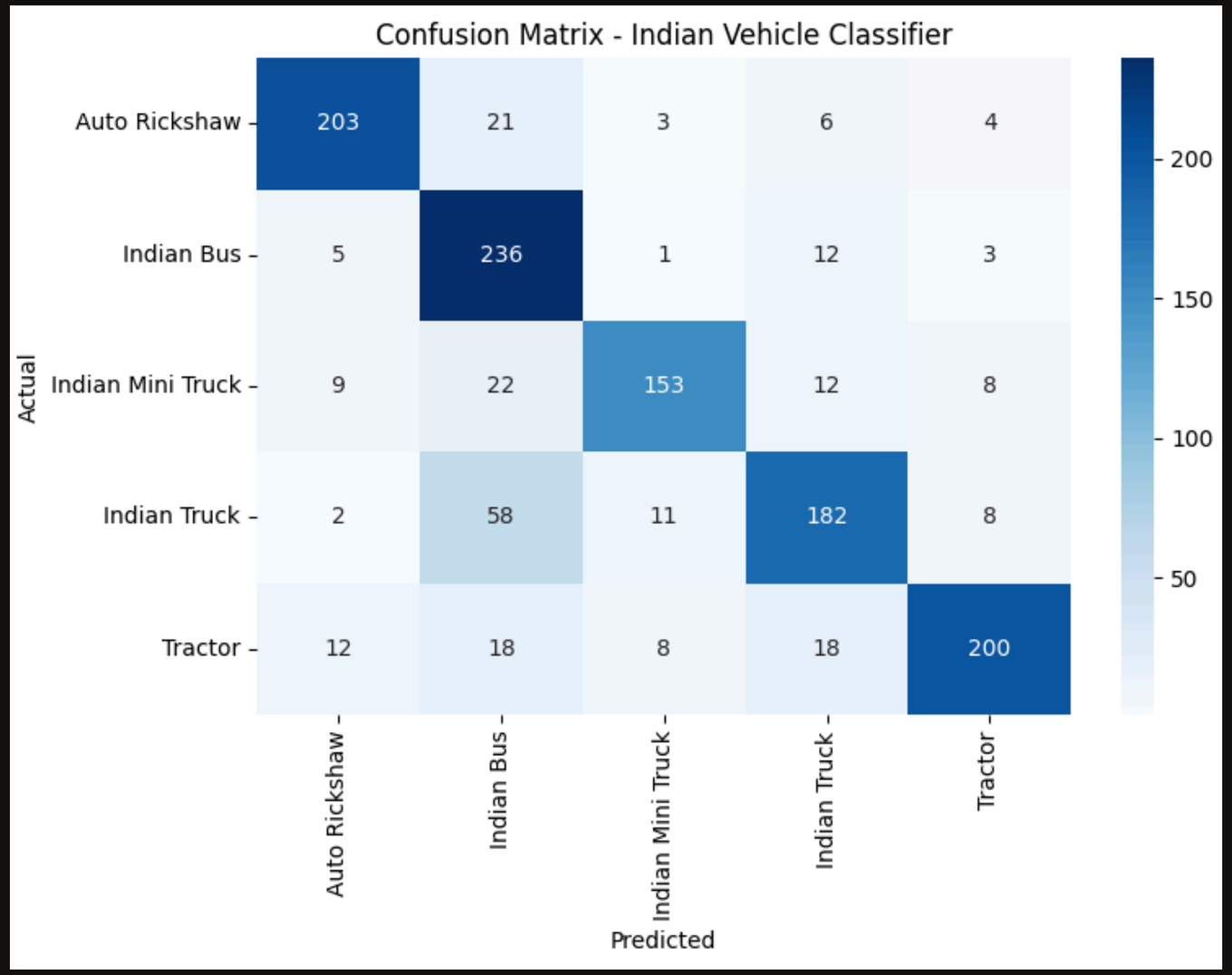
GRAPHS & PLOTS



🔹 **Accuracy Plot**

- Y-axis: Classification Accuracy (%)
- X-axis: Number of Epochs
- **Lines:**
  - Training Accuracy: Shows how well the model fits the training data.
  - Validation Accuracy: Shows how well the model generalizes to unseen (validation) data.

GRAPHS & PLOTS



🔍 3. Confusion Matrix Plots

Meaning:

- A matrix that visualizes the model's classification performance across all classes by comparing predicted vs. actual labels.

FORMULAS

$$T_{\text{green}} = \sum_{i=1}^{n} (C_i \times W_i)$$

Our CNN-based classifier (MobileNetV2) identifies the vehicle category from input images. Once the vehicles are classified, we count the number of each vehicle type (e.g., cars, bikes, trucks) in a frame. Then, based on predefined weight factors Wi for each category, we compute the required green signal time using the equation:

To make it even stronger, include:

- A table of sample weights Wi for each class (e.g., bike = 2s, car = 4s, truck = 6s)

An example:

- If 5 cars, 2 trucks → T=(5×4)+(2×6)=20+12=32sT = (5×4) + (2×6) = 20 + 12 = 32sT=(5×4)+(2×6)=20+12=32s

INDIAN VEHICLE DATASET & THEIR CONFIDENCE LEVEL

Predicted: indian_truck
Confidence: 0.72

Predicted: indian_mini_truck
Confidence: 0.88

Predicted: indian_bus
Confidence: 0.97

Predicted: auto_rickshaw
Confidence: 0.78

Predicted: tractor
Confidence: 0.99

RESULTS

Dynamic clearance time estimation from one image takes roughly 0.1 to 0.35 seconds on a standard CPU. And worse case will be 0.65 to 0.7 seconds.

```
Skipping unknown category: person

⏱️ Estimated Time for Each Category to Pass Signal (seconds):
Auto Rickshaw: 12.38 seconds
car: 9.36 seconds
motorcycle: 4.05 seconds

🟢 Total Estimated Time for All Vehicles to Pass: 25.79 seconds
```

```
🟩 Final Vehicle Count (YOLO-priority):
Auto Rickshaw: 5
person: 4
car: 4
motorcycle: 3
```

IMPACT & IMPLEMENTATION

- The proposed hybrid approach effectively detects and classifies both general and region-specific Indian vehicles, ensuring high accuracy in complex traffic scenarios.
- It assists **smart traffic signal systems** to dynamically allocate green signal time based on real-time vehicle count and type (e.g., auto-rickshaw vs. truck).
- The system supports **urban traffic planning**, congestion reduction, and emergency routing decisions in Indian cities.
- Can be deployed in **smart cities**, **automated surveillance**, and **toll automation** systems for intelligent transportation management.

ROOM FOR IMPROVEMENT

Dataset Expansion: Incorporating more diverse images (e.g., nighttime, rain, occlusions) to improve model robustness.

Model Integration: Training a unified YOLO model with Indian vehicle classes could reduce inference time.

Video Stream Optimization: Enhancing real-time processing capabilities for live traffic footage, especially on edge devices.

Accuracy Boost: Integrating ensemble learning or transformer-based models to improve classification performance.

Multi-Lane Handling: Extending the model to analyze and handle multiple lanes independently.

REFERENCES

{ultralytics2023}
Jocher, G. et al. (2023). YOLO by Ultralytics. GitHub Repository. \url{https://github.com/ultralytics/ultralytics#}.

{Adaptive_Traffic_Signal_Timer}
Mihir M Gandhi. Adaptive Traffic Signal Timer. \url{https://github.com/mihir-m-gandhi/Adaptive-Traffic-Signal-Timer}.

{sharma2021traffic}
Sharma, S., Jain, A., \& Singh, A. (2021). Intelligent Traffic Light Control Using Real-Time Vehicle Detection and Classification. \textit{International Journal of Computer Applications}, 183(5), 1–6.

THANKYOU

Join Us on the Road to Innovation