[Image & video processing, analysis of traffic signal areas using deep learning ,AI Models]

Submitted
By
P. Lakshmi Vignesh [BU21EECE0100515]
Moda SriRangaManjula [BU21ECE0100104]
Lalam Jithendra [BU21EECE010487]

Under the Guidance of
(Sanhita manna, Assistant professor)
(Duration: 06/07/2024 to 13/03/2025)

Department of Electrical,Electronics and Communication Engineering
GITAM School of Technology
GITAM
(DEEMED TO BE UNIVERSITY)
(Estd. u/s 3 of the UGC act 1956)
NH 207, Nagadenehalli, Doddaballapur taluk, Bengaluru-561203 Karnataka, INDIA.

DECLARATION

I/We declare that the project work contained in this report is original and it has been done by me under the guidance of my project guide.

Name:
P Lakshmi Vignesh

Date:22-08-2024 **Signature of the Student**

Department of Electrical,Electronics and Communication Engineering

GITAM School of Technology, Bengaluru-561203

CERTIFICATE

This is to certify that () bearing (Regd. No.:) has satisfactorily completed Mini Project Entitled in partial fulfillment of the requirements as prescribed by University for VIIIth semester, Bachelor of Technology in “Electrical, Electronics and Communication Engineering” and submitted this report during the academic year 2024-2025.

[Signature of the Guide] **[Signature of HOD**

Table of contents

Chapter 1: Introduction

1.1 Overview of the problem statement:

In any part of our life, each of us has encountered high-priority vehicles like ambulances, fire engines, and police vehicles waiting during their time of emergency. We have also experienced waiting for 90 seconds at traffic signal lanes even if the other lanes are empty. We can save significant time and prioritize vehicles at traffic signals by analyzing and executing better algorithms with the help of video processing and Al models. The main motto of this project is to eliminate the excess time and prioritize the vehicles at the traffic signals. This can be achieved by training deep learning models.

The rapid exponential growth in vehicle numbers have led to increased traffic congestion, especially in urban areas. One of the key components of modern traffic management is the monitoring and analysis of traffic signal areas, where congestion, accidents, and inefficiencies often occur. This problem statement revolves around the need to develop an advanced system that can automatically monitor, process, and analyze images and videos from traffic signal areas. By applying image and video processing, deep learning, and AI to analyze the vehicle densities to predict the time required for that lane and to identify the high priority vehicles and to prioritize them at traffic junctions. Which reduces the emergency rate which can save lives by reducing the waiting time and also results in time effective advancement at traffic junctions.

1.2 Objectives and goals:

The main motto of this project is to eliminate the excess time and prioritize the vehicles at the traffic signals. This can be achieved by training deep learning models. Firstly, the models will analyze the density of the vehicles at each lane of traffic areas. Additionally, the models will identify high priority vehicles such as ambulances and fire engines.

Firstly, the models will analyze the density of the vehicles at each lane of traffic areas. Additionally, the models will identify high priority vehicles such as ambulances and fire engines etc.. while analyzing the vehicles using object detection ai model.

To display the identified vehicles at traffic signals and to calculate the time required to clear the vehicles density of a particular lane.

Main Goals :

1. To assign the time to each lane (i.e 5<=time<=120) by analyzing the vehicle density of that lane.
2. To prioritize the high priority vehicles identified while analyzing the vehicles and displaying them to clear the path for those vehicles in that lane.

Additional goals

- To reduce the emergencies for people by prioritizing the high priority vehicles.
- To reduce the time for which the vehicles are waiting at traffic junctions.

Chapter 2: Literature Review

2.1 Advanced Traffic Signal Control System for Emergency Vehicles: Published in December 2022, this study was authored by Sunil M, V Yashaswini Naidu, Vignesh R, Vishwas P, and Amitha S. It utilizes IoT and Machine Learning technologies to enhance traffic signal control for emergency vehicles. The primary drawbacks highlighted in this study include issues with the reliability of detection, challenges in integrating with existing traffic infrastructure, and concerns related to privacy.

Link: Advanced Traffic Signal Control System for Emergency Vehicles

2.2 AI-Based Emergency Vehicles Detecting and Traffic Controlling System: This research, published in March 2024 by Aditya Pawar and Yogesh Sutar, explores the use of AI, object detection, and video processing for emergency vehicle detection and traffic control. The study identifies several drawbacks, such as the reliability and safety of AI, security and privacy concerns, and issues related to scalability and maintenance.

Link: AI-Based Emergency Vehicles Detecting and Traffic Controlling System

2.3 Traffic Management for Emergency Vehicle Priority Based on Visual Sensing: Authored by Kapileswar Nellore and Gerhard P. Hancke and published in August 2016, this study focuses on using object detection and image processing to prioritize traffic management for emergency vehicles. The drawbacks noted in this research include the need for MAC protocol enhancement, challenges in scalability and deployment, and high energy consumption.

Link:
Traffic Management for Emergency Vehicle Priority Based on Visual Sensing

2.4 Efficient Dynamic Traffic Control System Using Wireless Sensor Networks: This study was published in June 2014 and authored by R. Bharadwaj, J. Deepak, M. Baranitharan, and V. Vaidehi. The research focuses on the use of embedded systems to develop a dynamic traffic control system utilizing wireless sensor networks. The study identifies several drawbacks, including potential privacy concerns, data latency and processing delays, and challenges in identifying emergency vehicles.

Link: Efficient Dynamic Traffic Control System Using Wireless Sensor Networks

Chapter 3 : Strategic Analysis and Problem Definition

3.1 SWOT Analysis

Strengths:

- Real-time Vehicle Density Analysis
- Prioritization of High-Priority Vehicles
- Improved Traffic Management
- Integration of Software and Hardware

Weakness:

- Dependence on High-Quality Hardware
- Complexity of Implementation
- Data Privacy and Security Concerns
- Maintenance Requirements

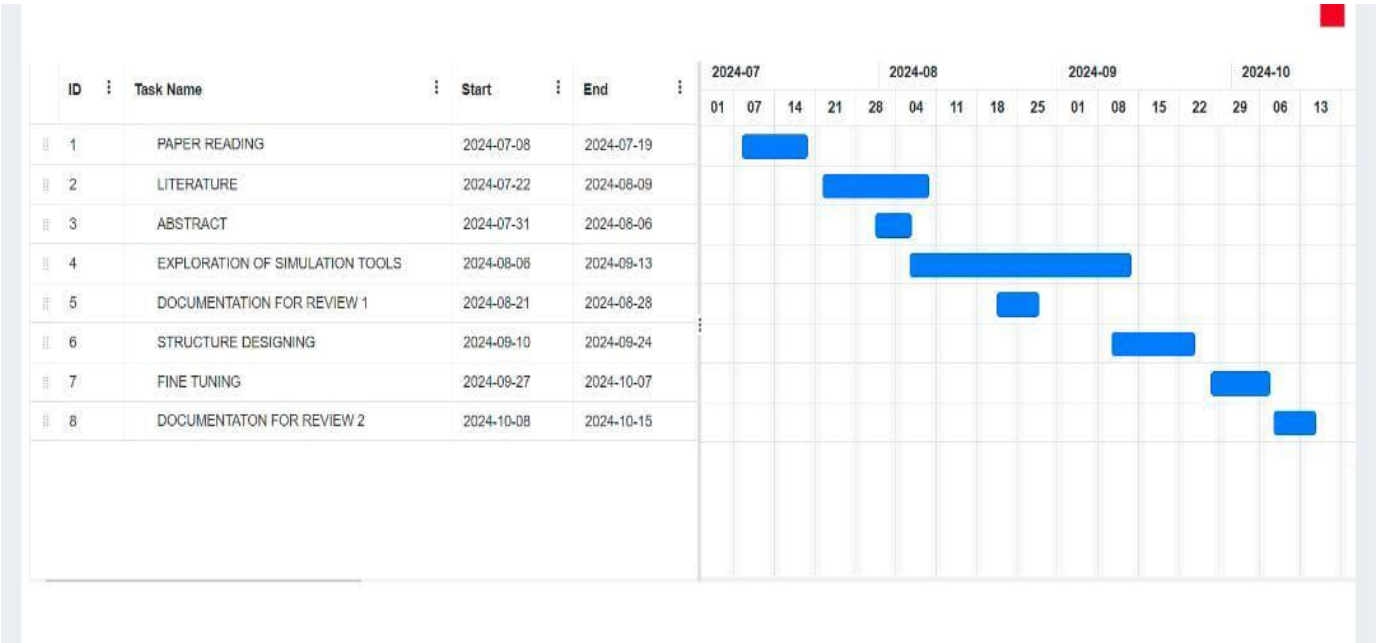
Opportunities:

- Scalability to Other Cities
- Potential for Integration with Smart City Initiatives
- Expansion to Additional Use Cases
- Collaboration with Government and Private Sectors

Threats:

- High Initial Costs
- Regulatory and Compliance Challenges
- Resistance to Change

3.2 Project Plan - Gantt chart



3.3 Analysis – 4W1H:

Why: The main goal is to reduce congestion and waiting times at traffic junctions by dynamically managing the traffic light timings based on real-time vehicle density analysis. Ensuring that high-priority vehicles like ambulances and fire trucks have a clear path, thereby reducing the response time during emergencies.

What: : The project involves developing a machine learning-based system for detecting and classifying vehicles, analyzing vehicle density, and managing traffic signals accordingly. Utilizing video and image preprocessing techniques to analyze traffic conditions in real time.

Where: The system is designed to be implemented at traffic junctions, particularly in busy urban areas, to optimize traffic flow and enhance safety.
When: The system is designed to be implemented at traffic junctions, particularly in busy urban areas, to optimize traffic flow and enhance safety.

When: The system is designed to be implemented at traffic junctions, particularly in busy urban areas, to optimize traffic flow and enhance safety.

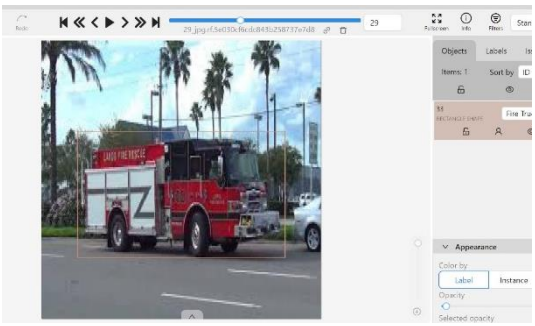
How: The system will use CCTV cameras and other security cameras installed at traffic junctions to capture real-time video footage. The system will use CCTV cameras and other security cameras installed at traffic junctions to capture real-time video footage. Deploying machine learning models to analyze the captured footage and make decisions regarding traffic light timings and high-priority vehicle management.

Chapter 4: Methodology

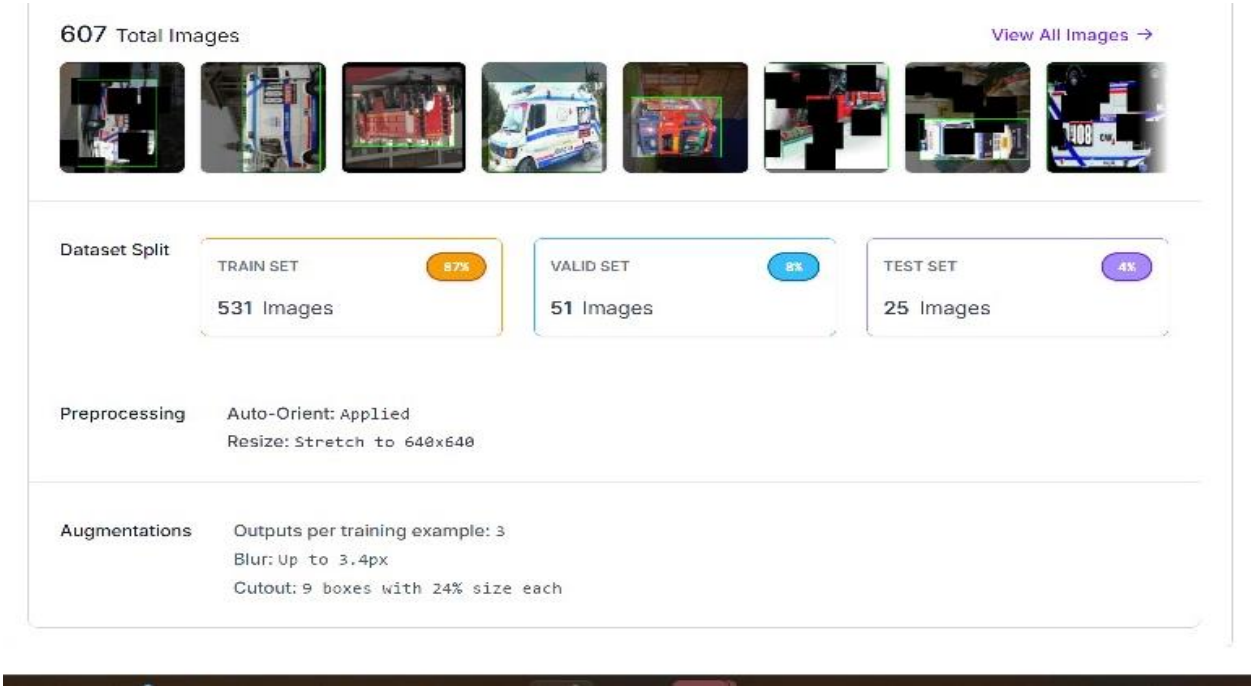
4.1 Description of the approach

Description of the Approach

1. **Data Collection**:
    - We started by collecting datasets from Kaggle, Roboflow, and various open-source image repositories. These datasets contained diverse images of vehicles, including ambulances, fire engines, and other regular vehicles, in different traffic scenario
    -
2. **Data Labeling and Annotation**:
    - The collected images were labeled and annotated using the CVAT.ai platform. We manually marked the vehicles and classified them based on their type (ambulance, fire engine, etc.) to prepare the dataset for training the object detection model.



3. **Preprocessing**:
    - Before feeding the data into the model, we applied several preprocessing steps. The images were resized to 640 x 640 pixels for uniformity. We also applied auto-orientation to correct any discrepancies in the image orientation.
    - For two of the models, we converted all images to greyscale to assess performance without color information.
4. **Data Augmentation**:
    - To make the model more robust, we used data augmentation techniques:
        - 15% of the images were converted to greyscale.
        - Blurring was applied to some images to simulate out-of-focus scenarios.

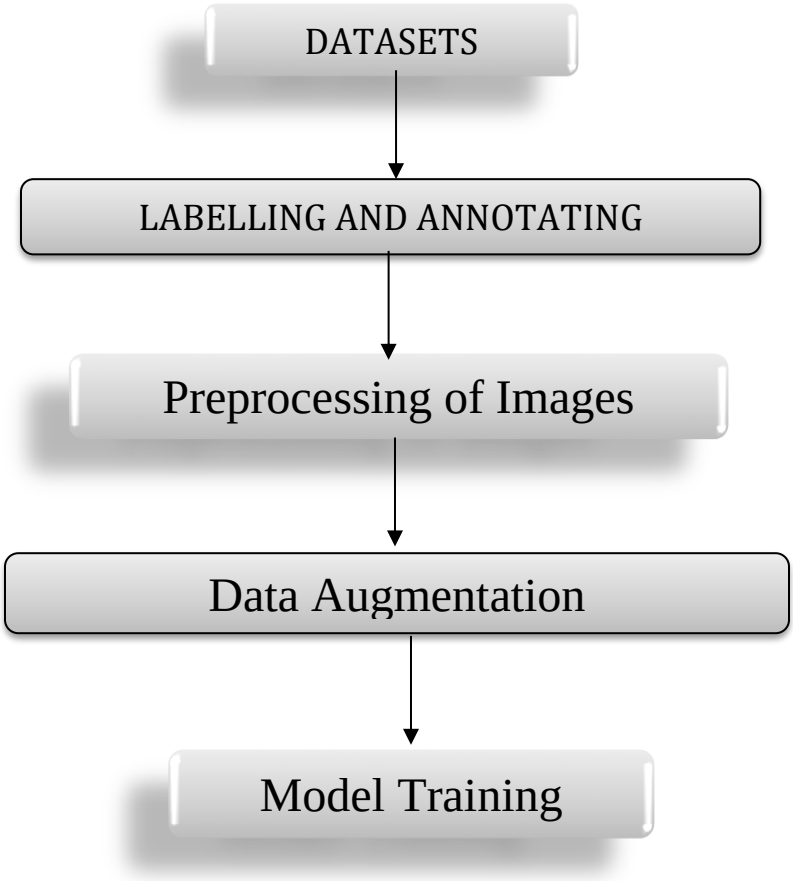


5. **Model Selection**:
   - We selected YOLOv11 for object detection, which provides multiple options such as augmentation, segmentation, posing, and classification. However, we focused specifically on the object detection model due to the need for real-time vehicle identification.
6. **Training**:
   - Initially, we attempted training with YOLOv11m, but due to system compatibility issues, we switched to YOLOv11n. After trying different numbers of epochs (iterations), we found that training with 50 epochs provided the best balance between accuracy and performance.
7. **Testing and Validation**:
   - The model was tested in real-world conditions, with a focus on detecting ambulances, fire engines, and other vehicles. Testing included scenarios with varied lighting, blurring, noise, and low visibility.
   - The system was validated against defined use cases and test cases, confirming that the model met accuracy and real-time performance requirements.

4.2Tools and techniques utilized:

1. Data Sources: Datasets from Kaggle, Roboflow, and open-source image repositories.
2. Labelling and Annotation: CVAT.ai platform for manual image labeling and bounding box annotations.
3. Pre-processing:
   - Image resizing to 640 x 640 pixels.
   - Auto-orientation adjustments.
   - Greyscale conversion for certain images.
4. Data Augmentation:
   - 15% greyscale,
   - Applying blur and noise for robustness.
5. Modelling:
   - YOLOv11 for object detection.
   - Trained with 50 epochs for optimal results.
6. System Environment: Optimized for handling real-time object detection tasks.
7. Validation and Testing: Validated against use cases and test cases to ensure accuracy.

4.3. **Structural diagram:**



CHAPTER 5 : Code and graphs

1. Code used to train the model

train.py > ...

```python
import ultralytics
from ultralytics import YOLO

model = YOLO("yolo11n.pt")
results = model.train(data="config.yaml", epochs=50)

```

2. Configuration for training

config.yaml

```yaml
path:
train: project_emergency vehicles_annotations_2024_10_14_12_12_47_yolov8 detection 1.0/images/train
val: project_emergency vehicles_annotations_2024_10_14_12_12_47_yolov8 detection 1.0/images/train

names:
  0: Ambulance
path: .
train: train.txt

```

3. Implementation and testing

`

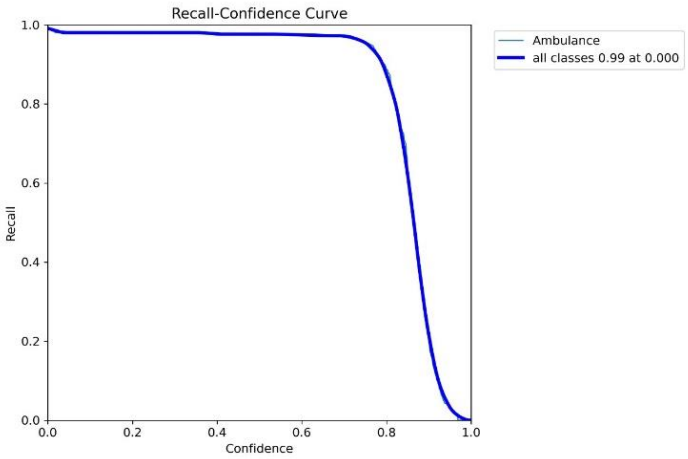
```python
import os
from ultralytics import YOLO
import cv2
import numpy as np

# Define directories and image paths
IMAGES_DIR = os.path.join('.', 'videos')
image_path = os.path.join(IMAGES_DIR, 'unnamed-file.jpg')
image_path_out = f"{image_path[:-5]}_out.webp"  # Output image path

# Load the image
image = cv2.imread(image_path)

# Check if the image was loaded successfully
if image is None:
    print(f"Failed to open image: {image_path}")
    exit()

H, W, _ = image.shape  # Get image dimensions

# Load the YOLO model
model_path = os.path.join('runs', 'detect', 'train23', 'weights', 'best.pt')
model = YOLO(model_path)
threshold = 0.5  # Confidence threshold

class_name_dict = {
    0: 'Ambulance',
}

# Perform inference on the image
results = model(image)[0]
```

```python
# Loop over detected objects
for result in results:
    if result.boxes:
        for box in result.boxes:
            conf = box.conf.item()
            cls_id = box.cls.item()  # Class ID

            if conf >= threshold:  # Only process if confidence is above threshold
                x1, y1, x2, y2 = map(int, box.xyxy[0].tolist())  # Bounding box coordinates

                # Draw a polygon (rectangle) around the detected object
                points = [(x1, y1), (x2, y1), (x2, y2), (x1, y2)]  # Four points of the polygon
                pts = cv2.polylines(image, [np.array(points)], isClosed=True, color=(0, 255, 0), thickness=2)

                # Put class name and confidence on the image
                label = f"{class_name_dict.get(cls_id, 'Unknown')} {conf:.2f}"
                cv2.putText(image, label, (x1, y1 - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 2)

                print(f"Class: {cls_id}, Confidence: {conf}, Box: ({x1}, {y1}, {x2}, {y2})")
    else:
        print("No detections found.")

# Save the output image with polygons
cv2.imwrite(image_path_out, image)
```

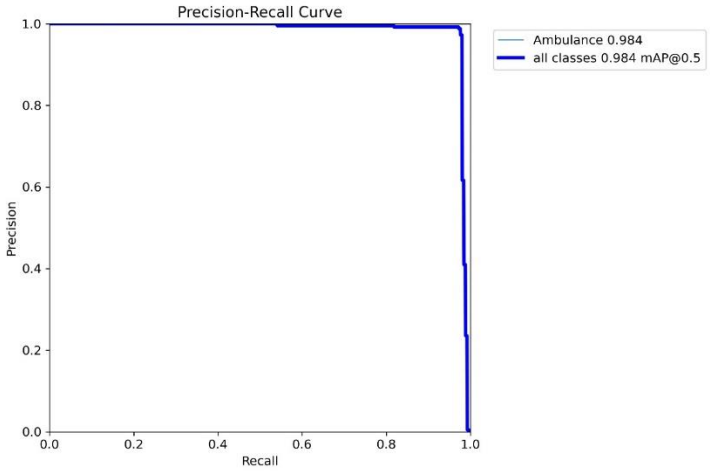
```
58    # Display the image (optional)
59    cv2.imshow('Image', image)
60    cv2.waitKey(0)  # Wait until a key is pressed
61    cv2.destroyAllWindows()
62
63    print(f"Processed image saved at: {image_path_out}")
64
```

Graphs:

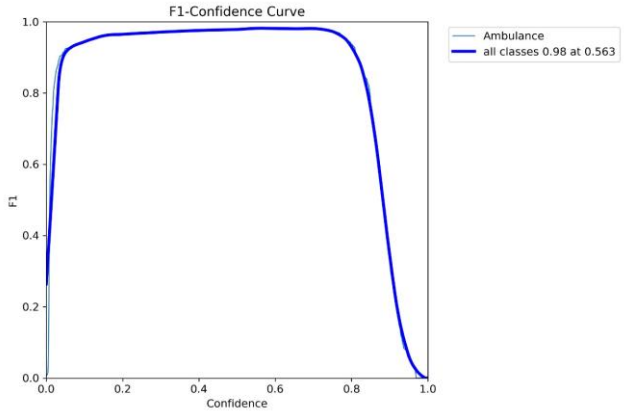
1.Recall – confidence matrix



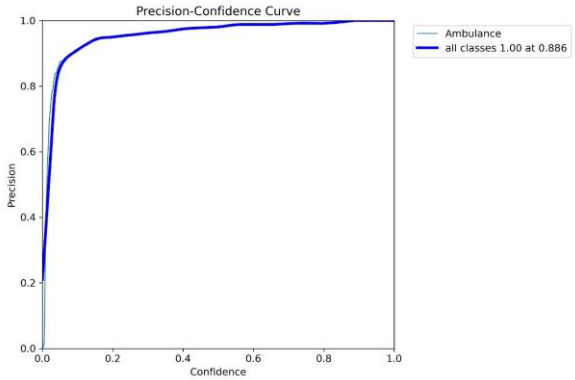
3. precision –recall matrix



4. F1- confidence matrix



5. Precision- confidence matrix



Chapter 6: Iteration +Results

6.1 Iterations:

Iteration 1 – YOLOv11n with 10 Epochs

- **Objective**: Test initial object detection accuracy and performance.
- **Result**: The model did not perform as expected. It failed to detect vehicles accurately, especially in real-time conditions.

Iteration 2 – YOLOv11n with 20 Epochs:

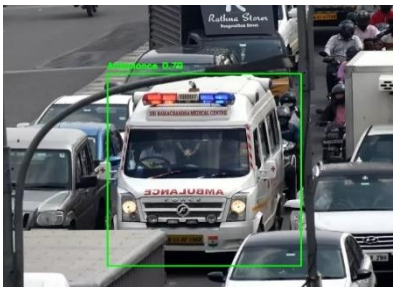
- **Objective**: Improve detection accuracy by increasing the number of epochs.
- **Result**: The model showed slight improvement but still had issues with accuracy, especially with detecting smaller vehicles and under noisy or blurry conditions.

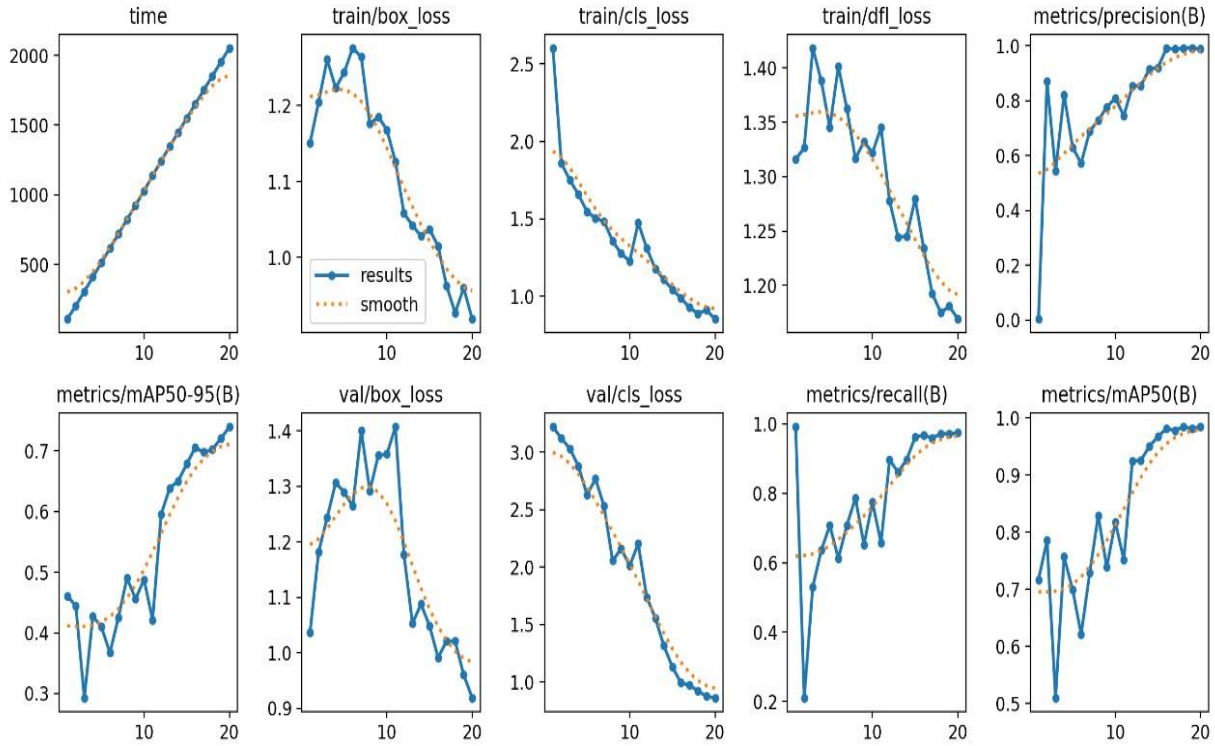
Iteration 3 – YOLOv11n with 50 Epochs:

- **Objective**: Optimize model performance with 50 epochs and further data augmentation.
- **Result**: The model achieved significant improvements, with over **90% accuracy** in detecting vehicles such as ambulances, fire engines, and regular cars. It performed well in real-time, even in varied lighting and environmental conditions.
- **Validation**:
  - **Use Case Validation**: The model successfully detected high-priority vehicles and adjusted traffic signals dynamically, as per the defined use cases.
  - **Test Case Validation**: The system passed all critical test cases, including vehicle detection under noise, blur, and various environmental conditions. The performance in terms of accuracy and real-time object detection was validated to be within the desired range.

6.2 Results:

- The final YOLOv11n model, after 50 epochs, was able to accurately detect ambulances, fire engines, and regular vehicles with high precision.
- The model was able to achieve over 90% detection accuracy, making it highly reliable for real-time traffic management.





Chapter 7: Summary and Conclusion :

7.1 Summary:

In this project, we developed a smart traffic management system capable of real-time vehicle detection using advanced YOLOv11 models.Data augmentation techniques such as greyscale conversion, blurring, and noise addition were applied to improve the model's robustness.

Through several iterations, we improved the model's accuracy, achieving over 90% detection accuracy after training for 50 epochs.The system successfully detects ambulances, fire engines, and other vehicles, adjusting traffic signals dynamically in real-time based on vehicle priority and proximity.

7.2 Conclusion:

The implementation of this smart transportation system demonstrates the potential to optimize traffic flow and reduce response times for emergency services, such as ambulances and fire engines. By using object detection technology, the system enhances safety and efficiency in urban traffic management.Future work will focus on extending the detection capabilities to more vehicle types, such as fire engines, and integrating distance estimation between vehicles to further improve signal optimization.

The results of this project show a promising step towards automated traffic management that can significantly improve urban mobility and safety.

Chapter 8: Future Work

Future Work: In future work, we aim to increase the accuracy of detecting fire engines and further enhance the system's performance. Additionally, we will work on finding the distances of high-priority vehicles from traffic signals. Based on this, the system will dynamically assign traffic signal time to prioritize emergency vehicles efficiently. This will lead to better traffic optimization and quicker response times for emergency services.

Chapter 9 :

9.1 Team Progress and Movement

- Dataset Collection
- Data labelling and Annotation
- Model Training
- Testing

9.2 Individual Contribution :

Key contributions: Putlooru Lakshmi Vignesh

- AI Model Training
- Model Testing

Key contributions: Lalam Jithendra

- Dataset Collection
- Model Testing

Key contributions: Moda Sri Ranga Manjula

- Data labelling and Annotation
- Model Testing

10.REFERENCES

- [1] Nellore K., Melini S.B. Automatic Traffic Monitoring System Using Lane Centre Edges. IOSR J. Eng. 2012; 2:1–8. [iosrjen]
- [2] Sangamesh S B, Sanjay D H, Meghana S, M N Thippeswamy “Advanced Traffic Signal Control System for Emergency Vehicles” International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-8 Issue-3, September 2019 [IRJET]

- [3] R. Bharadwaj; J. Deepak; M. Baranitharan; V. V. Vaidehi. "Efficient dynamic traffic control system using wireless sensor networks" Doi: 10.1109/ICRTIT.2013.6844280 [IEEE]

- [4] Krishnendu Choudhury; Dalia Nandi "Detection and Prioritization of Emergency Vehicles in Intelligent Traffic Management System" 2021 IEEE Bombay Section Signature Conference (IBSSC) DOI: 10.1109/IBSSC53889.2021.9673211 [IEEE]

- [5] Lingani, Guy M., Danda B. Rawat, and Moses Garuba. "Smart traffic management system using deep learning for smart city applications." 2019 IEEE 9th annual computing and communication workshop and conference (CCWC). [IEEE] , 2019.