Effective Traffic Signal Control System Using Deep Learning

Sk. Sadik[1], P. Akash[2], P. Persi, and N. Prasanth[4]
[1,2,3,4]ECE, RVRJCCE, INDIA
E-mail: sadiqsk94276@gmail.com[1]
E-mail: akashbabupurimitla@gmail.com[2]
E-mail: persi@gmail.com[3]
E-mail: naradasup@gmail.com[4]

*Abstract—***Traffic congestion is becoming a serious problem with a large number of vehicles on the roads. The present traffic system is a timer-based system that operates irrespective of the amount of traffic if there exists an ambulance. So, this Deep Learning project is designed in such a way that the traffic control system is based on vehicle density in a lane and also detecting the ambulance's lane and let that particular lane pass considering as a first priority. In fact, we use computer vision to have the characteristics of the competing traffic rows at the signals. This is done by a object detection model based on a Deep Learning model called You Only Look Once (YOLO)v8. Then traffic signal phases are optimized according to collected data, mainly queue density and waiting time per vehicle, to enable as much as more vehicles to pass safely with minimum waiting time**.

Keywords— Object detection, ambulance, YOLOv8, Deep learning

I. INTRODUCTION

Traffic congestion is a pervasive issue arising when the number of vehicles on the road surpasses the road's capacity, particularly prominent in densely populated urban areas with limited road infrastructure. This phenomenon leads to frustrating traffic jams, delays, and extended travel durations. Its ramifications span economic, social, and environmental domains, elevating transportation costs, squandering fuel, and inducing productivity losses due to prolonged travel periods. Moreover, congestion detrimentally affects air quality, public health, and safety.

Various factors contribute to congestion, encompassing the volume of vehicles, road network capacity, and quality of public transportation services. Particularly in urban settings, the challenge of detecting emergency vehicles amidst congested traffic is pronounced. Sirens and flashing lights may go unnoticed amid urban cacophony and crowded thoroughfares, impeding emergency response and potentially jeopardizing lives.

Hence, the implementation of an efficient ambulanc detection system holds paramount importance in traffic management. Such systems not only expedite emergency response times but also enhance public safety by mitigating the risk of accidents involving emergency vehicles and other motorists. Moreover, they bolster the efficiency of emergency services, enabling quicker response, enhanced patient throughput, and ultimately, the preservation of lives. Consequently, the development and deployment of robust emergency vehicle detection systems should be a foremost priority for transportation authorities and emergency services.

II. RELATED WORK

In the field of traffic management, there are ways to use computers to spot emergency vehicles like ambulances. Some recent studies have suggested using special computer programs called CNNs to do this. These studies, papers [1] and [2], tried different versions of this program called YOLOv5 and YOLOv3.

Here's how it works: First, the computer program looks at pictures taken from cameras on the road. It figures out if each vehicle is a car, bus, or truck. If it thinks something is a truck, it looks closely at that part of the picture. It then checks if it might be an ambulance by comparing it to lots of pictures of ambulances it's seen before. The downside of this method is that it has to save each picture individually in a folder. This means it uses a lot of storage space.

[3] developed a system to help traffic flow when emergency vehicles need to get through. They used a special kind of computer program called a Convolutional Neural Network (CNN), which is like a smart brain. This

program runs on a small computer called a Raspberry Pi.

Here's how it works: You give it a video showing traffic, and it quickly decides whether to let emergency vehicles pass or not. If there's an emergency vehicle in the video, it shows a green light. If not, it shows a red light. But it only pays attention to emergency vehicles, not regular ones. So, it doesn't handle situations where there's other traffic or how to manage that.

[4] suggested a new way to manage traffic using a system that counts vehicles using an IR sensor. When you turn it on, it gets power from a 5-volt battery. The IR sensor keeps track of how many vehicles are on the road. The main part of this system is a small computer called an Arduino UNO. It looks at the vehicle counts to figure out what's happening on the road. But there's a problem: the IR sensor can be expensive to install, and if it gets damaged, the system won't work anymore.

In another paper, [5], they talked about a different approach using something called YOLOv3. It's like a smart camera that can recognize cars, buses, and trucks. They tested different versions of this smart camera, and they found that one called Vgg-16 was the most accurate, with 99.73% accuracy. The dataset they used for this had pictures of emergency vehicles and regular vehicles, like cars and trucks. There were 1500 pictures of emergency vehicles and 8144 pictures of regular vehicles.

[6], suggested a system that uses YOLO, a smart camera, trained specifically to spot ambulances and fire engines. If it sees one of these, it shows a green light. If not, it switches to counting vehicles. However, this counting method can be inaccurate if there are big vehicles blocking the camera's view. Ideally, the system should give priority to the side with these big vehicles because it takes longer to clear the traffic. But this system doesn't do that.

[7], Another model, proposed by someone else, uses MATLAB, a computer program, to count vehicles in video frames. It assigns different timings for traffic lights based on the vehicle count. It works by comparing each frame to a reference background, then figuring out which parts have changed (the vehicles). However, this method has its problems too. It needs to adjust its settings as lighting conditions change, which can be tricky to do in real-time. Also, it only makes decisions based on vehicle count, not considering other factors like traffic congestion.

III. PROPOSED METHODOLOGY

This section presents smart traffic management combiningYOLOv4andMobileNetV2tohandledense,heavytrafficandrecognizeemergencyvehicles.TheSystemArchitectureispresentedinFig.1.Thefunctionalityofthissystemisbroadly classifiedintothreesections.Firstly,TheInputAcquisitionandVehicleClassificationsectionperformsinputimageprocessingusing the OpenCV module in python and classifying the typeof vehicles detected. Next, the Classification of EmergencyVehicles section performs further classification of vehicles for the detection of emergency vehicles. Finally, the ComputationofAverageVehicleAreasectioncalculatestheaveragevehiclearea which is used as a parameter for the decision of traffic light.

I. Input Acquisition and Vehicle Classification Section

The captured traffic image being fed into the YOLOv8 algorithm for object detection. This is achieved by utilizing OpenCV's imread() method in Python to read each frame individually. YOLO (You Only Look Once) is a widely used object detection algorithm known for its real-time capabilities. YOLOv8, an iteration released in April 2020, builds upon the strengths of its predecessors, particularly YOLOv3, enhancing both speed and accuracy while addressing previous limitations. Notably, YOLOv8 leverages the COCO (Common Objects in Context) dataset for training, a significant improvement over previous versions.

The COCO dataset, containing approximately 330,000 images and over 2.5 million object instances, serves as a vast collection for object detection, segmentation, and captioning. By training YOLOv8 on this dataset, the algorithm gains access to a broader range of objects and scenarios, enhancing its accuracy in object detection and classification. And also using “labeling” tool we label/train the data. The primary objective of YOLOv8 is to detect vehicles categorized as cars, buses, trucks, or motorbikes, providing a bounding box for each detected vehicle. These bounding boxes are drawn around identified vehicles, enabling YOLOv8 to distinguish between different types of vehicles. Following detection, vehicles are segmented based on the coordinates of their bounding boxes. After segmentation, vehicles are validated to determine whether they are trucks or buses. Those meeting this criterion are then passed to the Classification of Ambulance section and subsequently to the Computation of Average Vehicle Area section. Vehicles failing to meet the validation criteria are directly forwarded to the Computation of Average

VOLUME: 08 ISSUE: 04 | APRIL - 2024 SJIF RATING: 8.448 ISSN: 2582-3930

Vehicle Area section.

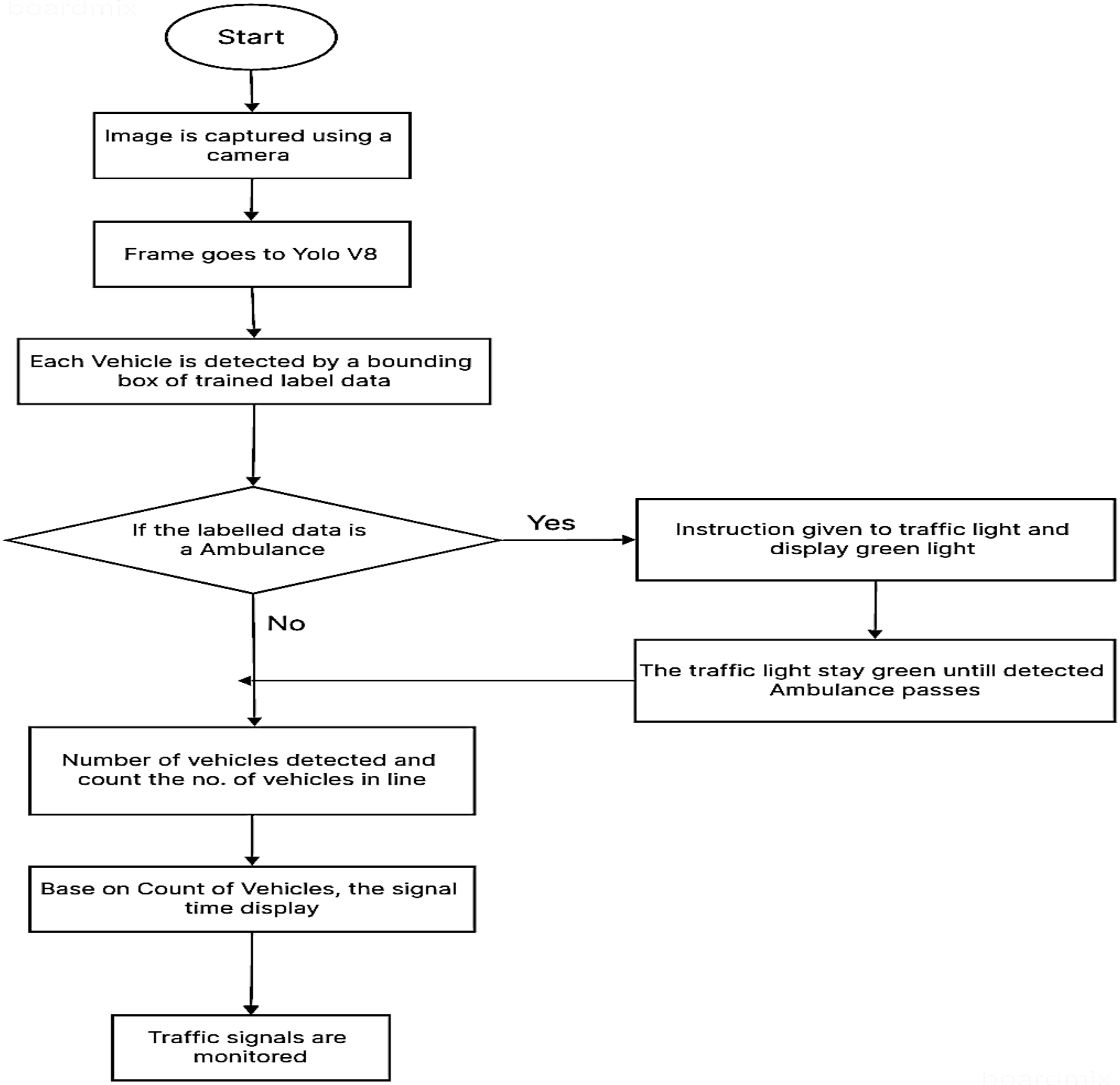


Fig.1:System Architecture fordetection of ambulance and TrafficManagement

II. Classification of ambulance Section

After detection of vehicles in an image by using yolov8 algorithm, their names are identified. Subsequently, we perform a check to determine if there is an ambulance present in the image. If the segmented image is predicted as an ambulance or fire engine then the count of emergency vehicles is updated and the command is provided to display green light in the traffic light.

Fig 2: sample images of dataset used of training YOLOv8

Fig3: Shows the detection of ambulance in the traffic lane

III. To count the number of vehicles in the lane

As most ambulances and fire engines are either similar to buses or trucks, instead of verifying every vehicle for the detection of emergency vehicles this filters out only buses and trucks for verification whether it is an emergency vehicle so, this reduces computational operations on the system. Each vehicle that passes that condition is segmented out using the bounding box coordinates provided by the YOLO v8 algorithm. If the count of vehicle is between 1-10 vehicles then we gives signal timing as 10 seconds, if the count is between 10-20 vehicles then signal timing is 20 seconds and if it is more than 20 vehicles then timing of signal is set to 30 seconds.

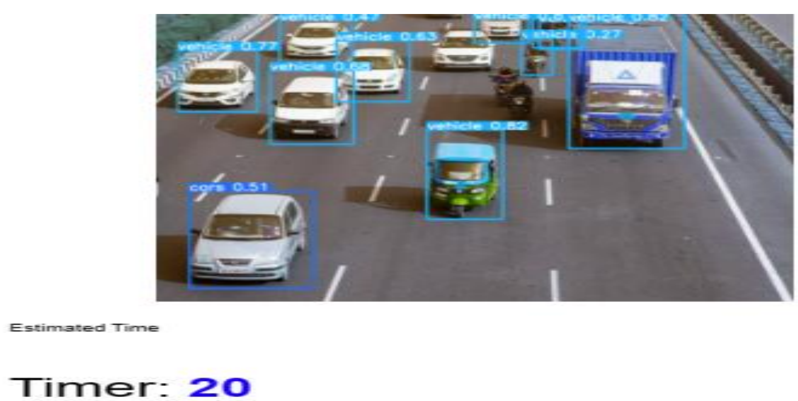
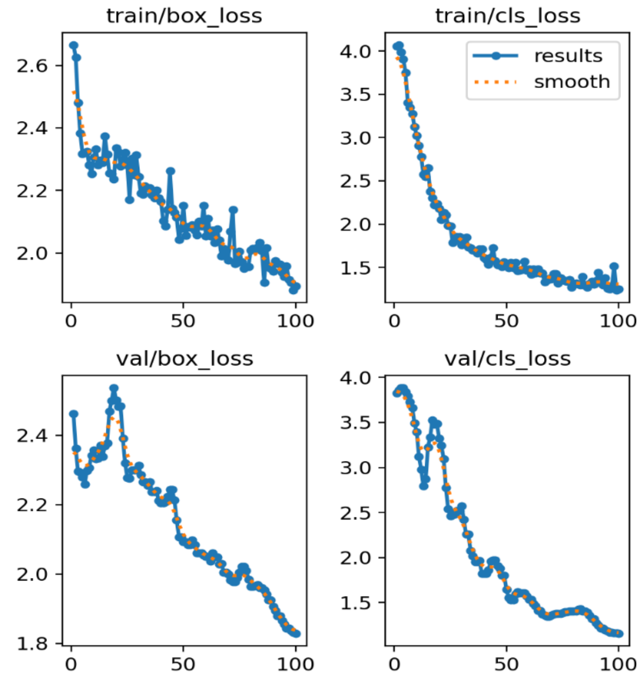
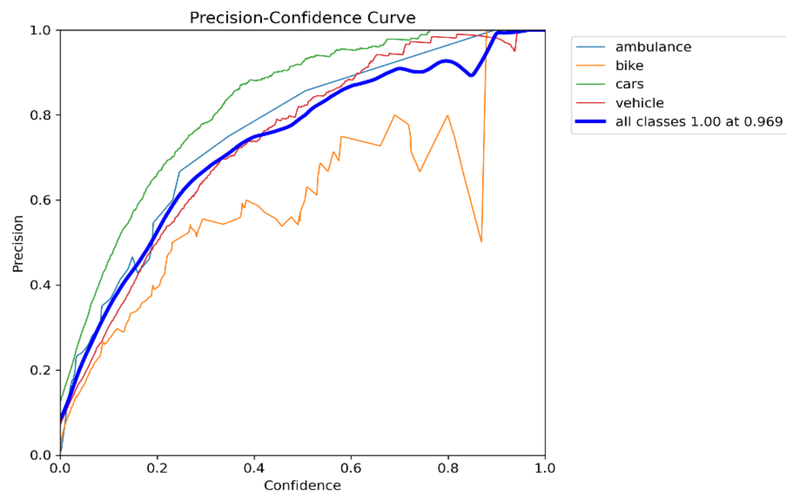


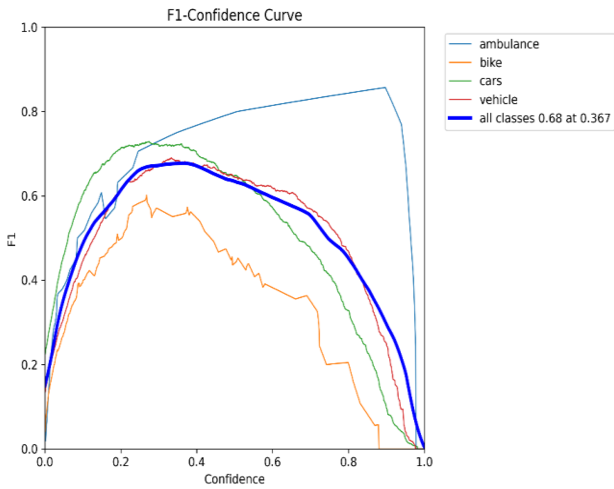
Fig 4: Shows the detection of normal vehicles and displays the time allotted to that lane

IV. RESULTS AND DISCUSSION

The graphs shown in the following fig of confidence level and precision curves over 10 epochs. The classification error is the parameter that gives the error percentage in classifying among categories in the training dataset which is calculated from predicted values. The classification error percentage is around 5% for the yolov8. This demonstrates that the classification error rate is quite low. Each training epoch lasts about 50 seconds, and the entire training session lasts 500 seconds. As the execution time is quick, this method is more practical. YOLOv8 is a model which is used for detection and computationally efficient, making it appropriate and convenient for using with limited processing power and it can be customized. The detection of ambulance by using YOLOv8 the accuracy is high and by bounding boxes we calculate the denity in a traffic. According to the density we can change the signal timing. The accuracy of detecting ambulance is high. These techniques help reduce overfitting, improve generalization, and increase training speed.. These techniques help reduce overfitting, improve generalization, and increase training speed. The whole system is substantially more effective and accurate because the entire system because detecting of ambulance is easy. This presents a smart, cost-effective, and reliable model which reduces human intervention and manages traffic by considering the density and presence of emergency vehicles, in order to provide precise results. It focuses on providing a management system with high efficiency to control real-time traffic.







V. CONCLUSION

The proposed work detects the total number of emergency vehicles and also computes the average vehicle area to provide optimized results for traffic management by employing the YOLOv8 model which is trained over 10 epochs with final validation accuracy of 96% and validation loss of 0.19. The total count factor and the identification of ambulance information are used for dynamic traffic monitoring. The system focuses on realizing a model that not only works for minimizing traffic issues but also aids the passage of ambulance as soon as possible. The count factor helps in enhancing traffic management to a much greater extent. The presented paper aims to increase cost effectiveness, and accuracy and reduce the complexity involved in Traffic Systems by giving preference to ambulance. The accuracy of prediction and model performance can be optimized by utilizing a high resolution camera, the higher the resolution, the greater the accuracy. The proposed model intends to achieve a smart traffic system that can be used for modern day optimal traffic monitoring.

VI. REFERENCES

[1] G. M. Bhoomika, "Ambulance detection using image processing," International Journal of Advanced Research in Science, Communication and Technology (IJARSCT), 2022.

[2] K. Agrawal, M. Nigam, S. Bhattacharya, and G. Sumathi, "Ambulance detection using image processing and neural networks," vol. 2115, no. 1, p. 012036, 2021.

[3] S. Deepajothi, D. P. Rajan, P. Karthikeyan, and S. Velliangiri, "Intelligent traffic management for emergency vehicles using convolutional neural network," vol. 1, pp. 853–857, 2021.

[4] N. N. S. N. Dzulkefli, S. Rohafauzi, A. N. Jaafar, R. Abdullah, R. Shafie, M. S. Selamat, N. S. Azman, and M. Z. Z. Muhammad, "Density based traffic system via ir sensor," vol. 1529, no. 2, p. 022061, 2020.

[5] S. Roy and M. S. Rahman, "Emergency vehicle detection on heavy traffic road from cctv footage using deep convolutional neural network," pp. 1–6, 2019.

[6] R. Harshitha, H. Shetty, B. Kavitharani, R. Madhura, and M. Jamuna, "Automatic traffic management system using yolo," Available at SSRN 3328586, 2018.

[7] S. Mehta, C. Paunwala and B. Vaidya, "CNN based traffic sign classification using ADAM optimizer", Proc. Int. Conf. Intell. Comput. Control Syst. (ICCS), pp. 1293-1298, May 2019