Proceedings of the Second International Conference on Applied Artificial Intelligence and Computing (ICAAIC 2023)
IEEE Xplore Part Number: CFP23BC3-ART; ISBN: 978-1-6654-5630-2

Smart Traffic Management System using YOLOv4 and MobileNetV2 Convolutional Neural Network Architecture

Varun Chava
UG Student, Department of ECE
Gokaraju Rangaraju Institute of Engineering and Technology
Hyderabad, India
janakiram1697@gmail.com

Sri Siddhardha Nalluri
UG Student, Department of ECE
Gokaraju Rangaraju Institute of Engineering and Technology
Hyderabad, India
srisiddhardha8@gmail.com

Sri Harsha Vinay Kommuri
UG Student, Department of ECE
Gokaraju Rangaraju Institute of Engineering and Technology
Hyderabad, India
sriharshavinay.k@gmail.com

Arvind Vishnubhatla
Professor, Department of ECE
Gokaraju Rangaraju Institute of Engineering and Technology
Hyderabad, India
vainfo66@gmail.com

2023 2nd International Conference on Applied Artificial Intelligence and Computing (ICAAIC) | 978-1-6654-5630-2/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICAAIC56838.2023.10141268

***Abstract*—Congestion owing to traffic is one of the crucial complications in urban cities, which is need to be addressed to improve traffic control and operation. The present traffic system is a timer-based system that operates irrespective of the amount of traffic and the existence of emergency vehicles like ambulances and fire engines. Vehicle flow discovery appears to be an important part of modern world traffic control and operation system. This design proposes a novel smart traffic system that utilizes real-time Average Vehicle Area and Emergency vehicle detection to optimize traffic flow and improve emergency response times. This system employs YOLOv4 and MobileNet V2 Convolutional neural network pre-trained model to accurately detect the number of vehicles present on the road, Average Vehicle Area and identify emergency vehicles in real-time. Using this information, this system can dynamically adjust traffic signals and reroute vehicles to minimize congestion and ensure priority access for emergency vehicles. Experimental results show that this system significantly reduces average travel times and emergency response times, making it a promising solution for modern traffic management and emergency services.**

***Index Terms*—Image Processing, YOLOv4, MobileNet V2, Convolutional neural network, Emergency Vehicle Detection, Average Vehicle Area**

I. INTRODUCTION

TRAFFIC CONGESTION is a common problem that occurs when the volume of vehicles on the road exceeds the road's capacity to accommodate them. It is a major issue in urban areas, where high population densities and limited road infrastructure lead to traffic jams, delays, and increased travel times. Congestion can have significant economic, social, and environmental impacts. It increases the cost of transportation, wastes fuel, and results in lost productivity due to delayed travel times. It can also have negative impacts on air quality, public health, and safety. There are many factors that contribute to traffic congestion, including the number of vehicles present on the road, the capacity of the road network, and the level of public transportation services.

Emergency vehicle detection is crucial in traffic management because it can significantly improve emergency response times and potentially save lives. When an emergency vehicle, such as an ambulance or a fire truck, needs to get through congested traffic, it can be challenging for drivers to hear sirens and see flashing lights, especially in noisy and crowded urban areas. This can lead to delays and even prevent emergency vehicles from reaching their destination on time. Implementing an effective emergency vehicle detection system can also improve public safety by reducing the risk of accidents involving emergency vehicles and other drivers. Additionally, it can improve the efficiency of emergency services by reducing response times, increasing the number of patients that can be treated, and potentially saving lives. Overall, the importance of emergency vehicle detection in traffic management cannot be overstated, and efforts to develop and implement such systems should be a priority for transportation authorities and emergency services.

This model intends to consider the Average vehicle area as a measure to take appropriate decisions to curb traffic congestion. The Average vehicle area is procured as the total vehicle area in the frame divided by the total number of vehicles in the frame. Considering a scenario where either side of the road has heavy and light vehicles respectively, the Average vehicle area measure will help us clear the traffic on the heavy vehicle side sooner, as the average vehicle area of heavy vehicles will be greater than that of light vehicles. This decision will help us curb traffic congestion in a much more efficient way because heavy vehicles take more time than normal vehicles even though the count of vehicles is the same on both lanes.

II. Related Work

With the evolution of Image Processing, there have been many methodologies employed for smart traffic management systems. The prominent aspect of this domain is to take decisions based on the results of monitoring emergency vehicles. [1] and [2] proposed a model based on CNN for ambulance detection. [1] uses the YOLOv5 algorithm whereas [2] uses the YOLOv3 algorithm. In both papers firstly, the pre-defined algorithm is applied to the pictures taken from the footage. It labels whether a vehicle is a car, bus, or truck. If any vehicle is found to be a truck, then the cropped image according to the bounding box is sent to the model which is pre-trained on a substantial set of images of the ambulance, it then predicts whether the cropped image is an ambulance or not. Both [1] and [2] have a setback as this is processed on a folder of images and every single time the images have to be saved, which makes this a storage-consuming process. [3] worked on a traffic management model for the detection of emergency vehicles employing a Convolutional neural network (CNN). The CNN model is deployed in the Raspberry Pi. It will take an input video of traffic and provide a swift decision to allow the passage of emergency vehicles. But this traffic system is made only for emergency vehicles. If an emergency vehicle is present in the input video, then it shows green, else the output is red. This model doesn't consider any other vehicles and traffic management of such a scenario is not provided.

[4] proposed a density-based system for traffic control employing an IR sensor. When the circuit is turned on, a 5-Volt DC source is used for the power supply. The IR sensor then begins to count the number of vehicles on the road as part of its function. The prominent density-based module in charge of all processes leading to this system's output is the Arduino UNO. The microcontroller then examines the results of the vehicle counts. The flaw of this system is that the installation of an IR sensor is expensive and if any damage occurs, then the system fails to detect vehicles. The paper [5] provided a model based on YOLOv3. In this, the CNN model is applied to all cars, buses, and trucks. A Comparison of performance is made when 2-layer CNN is used it is 98.39% accurate. While Vgg-16 has 99.73%, inception -V3 has 97.57%, and xception has 98.84% accuracy. This model's dataset has two categories with 1500 images of emergency vehicles and 8144 images of non-emergency vehicles, where non-emergency vehicles comprise cars, trucks, bikes, etc.

[6] proposed a system in which YOLO is trained on ambulance and fire engine images. If an ambulance or fire engine is detected, it gives output as the green light, or else it is operated based on the count of vehicles. This count-based traffic management may be the accurate majority of the time, but if there are heavy vehicles in one lane then the camera cannot see the vehicles behind these heavy vehicles so the count of vehicles will be inaccurate. In this scenario, ideally, the preference should be given to the side with heavy vehicles as this side takes much time to clear the heavy traffic congestion. But this model doesn't take this into consideration.

The model proposed by [7] uses image processing tools in MATLAB to count the vehicles in the frames, and different timings are assigned according to the count along with a green signal for vehicles to pass. This system works using background reference and background subtraction using threshold and enhancing the foreground. This system has drawbacks as, over time with the change in the lighting, the threshold should be changed because the reference background image lighting may not be maintained throughout real-time video processing. Also, this model takes decisions based on the count.

[8] proposed a traffic management system for controlling traffic signals that makes use of hardware like the RFID Tag, Node MCU, Raspberry PI, and Reader. The four possible routes that this system can take are A, B, C, and D. An RFID tag is allocated to each emergency vehicle if it is approaching path B. The reader determines and recognizes the tag that is attached to emergency vehicles when the vehicle is within a certain range of the reader. The system then starts communication with Node MCU. It will alter the path B signal's status, and it will increment the interval between green lights if the signal is already green. Or, if the signal is red, it will automatically change to a green light if the RFID reader detects an emergency vehicle. After this process, the MQTT protocol will be used by the Node MCU to communicate with the Raspberry Pi. This Smart traffic management system IoT is a system that makes extensive use of hardware sensors. The major setback of this system is that every emergency vehicle should be given an RFID tag, or else the emergency vehicle will not be detected. Also, if there is any damage in the system, then the whole system becomes futile. And, the inclusion of RFID tags makes the system expensive.

A timer-based hard-coded system that is independent of the volume of traffic present is the current exiting approach for real-time traffic control. Hence, the proposed work intends to provide a model which eliminates the complexity of existing traffic management systems and furnishes an efficient solution addressing the issues and drawbacks of the existing models. This paper presents a model using YOLOv4 and MobileNet V2 Convolutional Neural Network Architecture for the detection of emergency vehicles along with the computation of Average Vehicle Area, whose collective results can provide optimized results of traffic management.

III. Proposed Methodology

This section presents smart traffic management combining YOLOv4 and MobileNet V2 to handle dense, heavy traffic and recognize emergency vehicles. The System Architecture is presented in Fig. 1. The functionality of this system is broadly

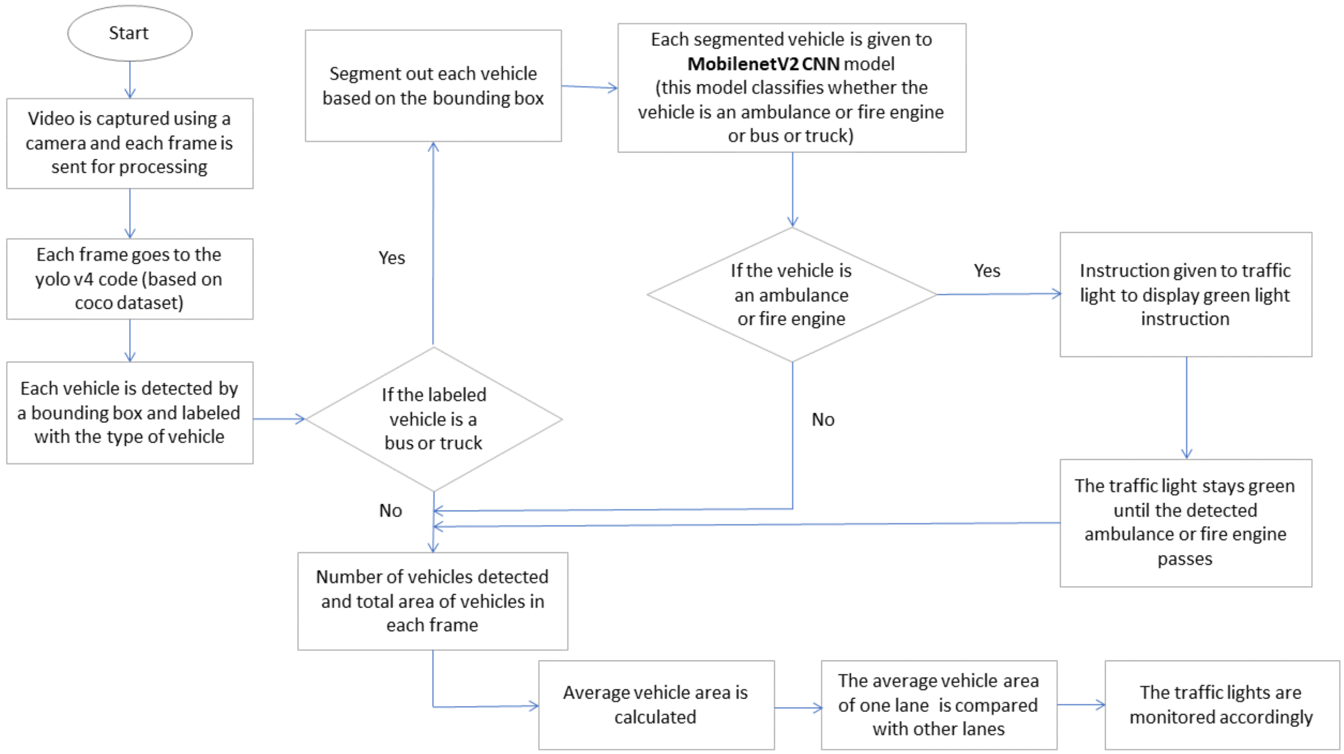


Fig. 1: System Architecture for detection of Emergency Vehicles and Traffic Management

classified into three sections. Firstly, The Input Acquisition and Vehicle Classification section performs input image processing using the OpenCV module in python and classifying the type of vehicles detected. Next, the Classification of Emergency Vehicles section performs further classification of vehicles for the detection of emergency vehicles. Finally, the Computation of Average Vehicle Area section calculates the average vehicle area which is used as a parameter for the decision of traffic light.

A. *Input Acquisition and Vehicle Classification Section*

Input acquisition is done using a camera. The camera is typically mounted on a pole or other structure near the intersection and captures video of the traffic which is processed frame by frame in real-time. Each frame is passed over to the YOLOv4 algorithm as input by reading that frame using the OpenCV's imread() method in python.

YOLO (You Only Look Once) is a popular object detection algorithm that can detect and classify objects in real time. [9] YOLOv4 is a version of the YOLO algorithm, which was released in April 2020. YOLOv4 builds upon the strengths of previous versions of YOLO, including speed and accuracy, while also addressing some of the limitations of the earlier versions. Both [3] and [5] are built on YOLO v3 version. One of the key improvements in YOLOv4 is the use of the COCO (Common Objects in Context) dataset for training. [10] the COCO dataset, which has about 330,000 images and over 2.5 million object instances, is a large object detection, segmentation, and captioning collection. By training YOLOv4 on the COCO dataset, the algorithm has access to a much larger and more diverse set of objects and scenarios, which improves its ability to detect and classify objects accurately. The main purpose of YOLOv4 is to detect each vehicle which is categorized as either a car or a bus or a truck or a motorbike and a bounding box is provided for each detected vehicle. A bounding box is drawn around the identified vehicle because YOLOv4 has already trained on these classes and can identify the type of vehicle. Following that, the vehicle is segmented out based on the bounding box's coordinates.

Then the condition whether the vehicle is a truck or a bus is validated. Each vehicle that satisfies this condition is passed to the Classification of Emergency Vehicles section and further passed to the Computation of Average Vehicle Area section. If the condition is not satisfied then it is directly passed to the Computation of Average Vehicle Area section.

B. *Classification of Emergency Vehicles Section*

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 YOLOv4 algorithm. A CNN model is needed as they can learn to recognize features in images by using convolutional filters that capture spatial

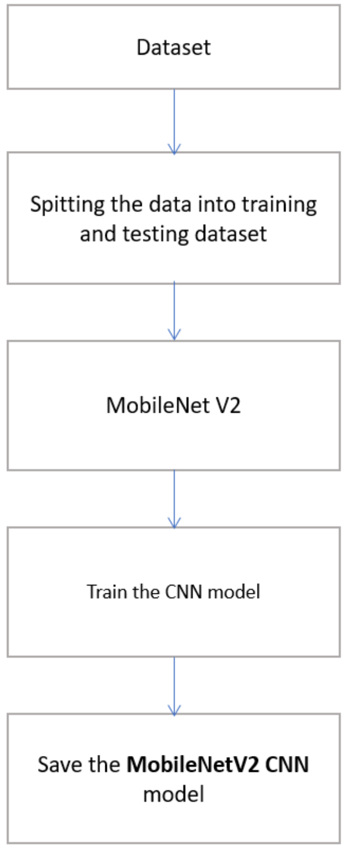


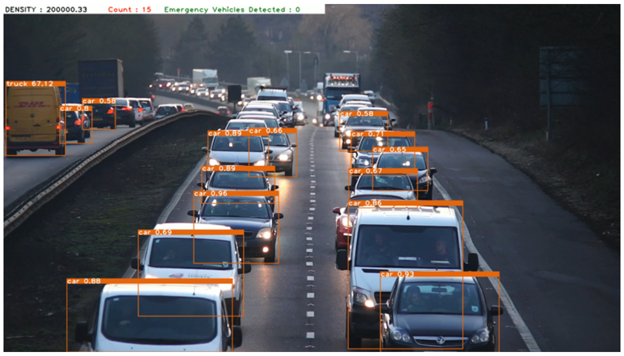
Fig. 2: Block Diagram of creating MobileNet V2 CNN Model

Fig. 3: Sample images of the dataset used for training MobileNet V2 CNN model

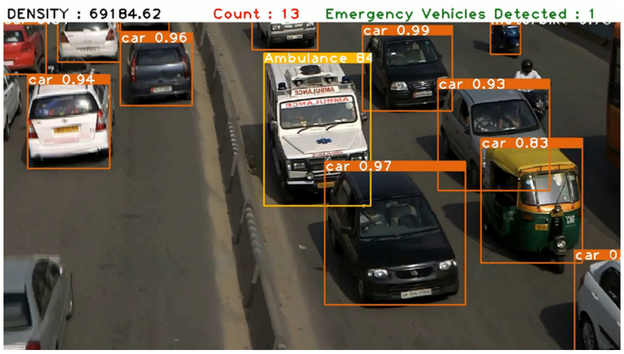
patterns and structures. so, this segmented image is further sent to the MobileNet V2 CNN model. There are further steps such as Training and Prediction.

1) Training: The block diagram of creating the MobileNet V2 CNN model is shown in Fig. 2. In this Training section, a dataset of images with four categories such as ambulance, bus, fire engine, and truck is considered. Each category contains 576 images in JPG file format in the dataset. The images in the dataset are gathered using a variety of online resources and Fig. 3. shows sample images of the dataset. The total dataset is diving into the training dataset and validation dataset with a validation split of 0.25. That means 75% of images are in the training dataset and 25% are in the validation dataset. For the training dataset, the parameters like shear range and zoom range values are given as 0.2 and the horizontal flip parameter is specified as true. The target size is maintained as 224 x 224 pixels and the batch size of 16 is used. This training dataset is trained on MobileNet V2 neural network architecture [11]. Unlike [1], [2], and [5] this used MobileNet V2 convolutional neural network (CNN) architecture because it is designed to be efficient and lightweight. This architecture is selected for its efficient computational design, it is suitable for usage on embedded and mobile devices with constrained RAM and CPU. It is an improvement over the original MobileNet architecture, which was also designed for efficient computation. The model is compiled by [12] Adam optimizer over 10 epochs. The model fitting of the training dataset with final validation accuracy of 95.49% and validation loss of 0.1810. This CNN model is saved using the Keras library.

2) Prediction: The trained MobileNet V2 CNN model is loaded and the segmented image is resized to the target size which is 224 x 224 pixels and further converted to an array which is given for the prediction method of the CNN



(a) Normal Vehicle Detection



(b) Ambulance Detection

(c) Fire Truck Detection

Fig. 4: Vehicle Detection and Displaying Average Vehicle Area

model. The output of the model is either of the four classes that is either an ambulance or a bus or a fire engine or a truck.

The classification is done over these four categories because the input of this section is either a truck or a bus. For each category, it is trained specifically whereas in [5] all the cars, buses, trucks, and bikes are given as one category and emergency vehicles as another category. 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. Next, the segmented image is further passed to the Computation of Average Vehicle Area section.

C. Computation of Average Vehicle Area Section

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 v4 algorithm. This segmented image is further sent to the MobileNet V2 CNN model. There are further steps such as Training and Prediction.

The Average Vehicle Area refers to the amount of space that a typical vehicle occupies on a road. That is the average of all vehicle areas, it can be calculated by the formula
Let,
Avg be Average Vehicle Area,
It is given as;

$$Avg = (X/N) * 100 \tag{1}$$

Where,
X be the Total area occupied by vehicles
N be the Number of vehicles.

computation of Average Vehicle Area requires the total area utilized by vehicles and the number of vehicles. The vehicles detected by the YOLOv4 algorithm consist of the bounding box, the area of each vehicle is calculated by multiplying the length and width of the bounding box and the sum of all such vehicle areas is the total area occupied by vehicles. The number of bounding boxes provides the count of vehicles. The Ratio of these two values will give Average Vehicle Area.

In [6] and [7] based on the number of vehicles present traffic scheduling is done but the proposed approach suggests using Average Vehicle Area as a parameter for traffic scheduling because heavy vehicles need comparatively more time than lightweight vehicles. The average vehicle area is an important factor in traffic engineering and urban planning, as it can affect the design of roads. Fig. 4. shows the output of this proposed method which displays the Average Vehicle Area as Density, the number of vehicles as Count, and the Number of emergency vehicles.

IV. Results and Discussion

Initially, a CNN model with 8 layers instead of the MobileNet V2 CNN model was experimented. The 8-layered CNN structure consists of two convolutional layers, two Max Pooling layers, one dropout layer of 0.5, one fattening layer, and the last two Fully connected layers out of which one has activation functions as relu and another as a sigmoid. The Validation accuracy of Experimented 8-layered CNN model is 69.79% and the Validation loss of 1.4518 where as the MobileNet V2 CNN model has 95.49% Validation accuracy and The Validation loss of 0.181. The performance of the CNN model is very low with low validation accuracy and high validation loss when compared to the MobileNet V2 CNN model. Fig. 5. shows the Performance metrics of Experimented

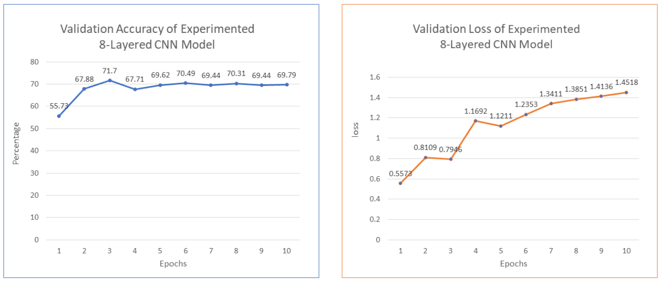


Fig. 5: Performance Metrics of 8-Layered CNN Model

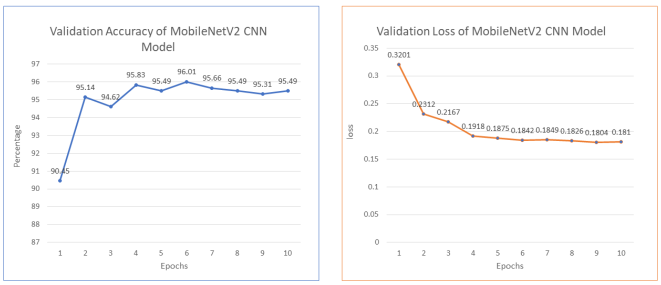
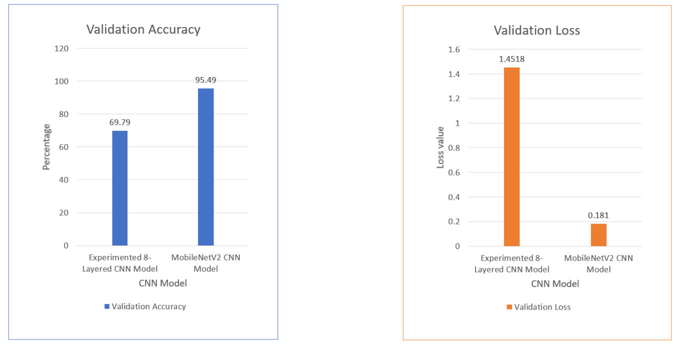


Fig. 6: Performance Metrics of MobileNet V2 CNN Model

8- layered CNN model for 10 epochs and Fig. 6. shows the performance metrics of the MobileNet V2 CNN model for 10 epochs. The comparison of both the final validation accuracy and final validation losses of the two models in the graph is represented in Fig. 7 which is generated from the fitting of the CNN model 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 MobileNet V2 CNN model. This demonstrates that the classification error rate is quite low. Each training epoch lasts about 65 seconds, and the entire training session lasts 650 seconds. As the execution time is quick, this method is more practical. MobileNet V2 is designed to be computationally efficient, making it appropriate and convenient for mobile and embedded devices with limited processing power and it can be customized. The architecture uses depth-wise separable convolutions, which split a standard convolution into separate depth-wise and pointwise convolutions. This reduces the number of parameters and computations required compared to the Experimented 8-layered CNN model while maintaining high accuracy.MobileNet V2 also includes several other methodologies to improve performance, such as linear bottlenecks, inverted residuals, and shortcut connections. These techniques help reduce overfitting, improve generalization, and increase training speed. The whole system



(a) Comparison of Validation Accuracy

(b) Comparison of Validation Loss

Fig. 7: Comparison of Experimented 8-layered CNN Model and MobileNet V2 CNN Model

is substantially more effective and accurate because the entire system uses a two-level identification of emergency vehicles rather than predicting every vehicle, the CNN model is only given the vehicles that YOLOv4 detects as trucks and buses. 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 YOLOv4 and MobileNet V2 model which is trained over 10 epochs with final validation accuracy of 95.49% and validation loss of 0.1810. The total vehicle area factor and the identification of emergency vehicle 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 emergency vehicles as soon as possible. The average vehicle area 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 emergency vehicles. 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.

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] U. E. Prakash, K. Vishnupriya, A. Thankappan, and A. A. Balakrishnan, “Density based traffic control system using image processing,” pp. 1–4, 2018.
[8] S. V. Bhate, P. V. Kulkarni, S. D. Lagad, M. D. Shinde, and S. Patil, “Iot based intelligent traffic signal system for emergency vehicles,” pp. 788–793, 2018.
[9] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “Yolov4: Optimal speed and accuracy of object detection,” arXiv preprint arXiv:2004.10934, 2020.
[10] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick, “Microsoft coco: Common objects in context,” pp. 740–755, 2014.
[11] K. Dong, C. Zhou, Y. Ruan, and Y. Li, “Mobilenetv2 model for image classification,” pp. 476–480, 2020.
[12] 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.