TRAFFIC LIGHT PRIORITY FOR EMERGENCY VEHICLE

Manoj Kumar M[1], Anandateertha[2], Abhishek B Kamble[3], ChennaKeshava Reddy[4]
Dr. Saneesh Cleatus Thundiyil[5]

[1234]U.G. Scholar, Department of Electronics and Communication Engineering
BMS Institute of Technology and Management, Avalahalli, Yelahanka, Bengaluru-560064
[5]Associate Professor, Department of Electronics and Communication Engineering,
BMS Institute of Technology and Management, Avalahalli, Yelahanka, Bengaluru-560064

---***---

Abstract - *Traffic Light Priority for Emergency Vehicles is a system that allows emergency vehicles to have priority at traffic signals by controlling the traffic lights in their favor. This system aims to reduce emergency service response times and improve the safety of both emergency responders and the public. This paper will explore the implementation of traffic light priority for emergency vehicles, including the benefits, challenges, and potential solutions for this technology. The study will also examine the impact of traffic light priority on traffic flow and the overall efficiency of emergency response systems. Finally, the paper will discuss the potential future developments of this technology and its potential to improve emergency services globally.*

***Key Words*: Emergency, Ambulance, Traffic light, Machine Learning Model, Images, Green, Red**

1. INTRODUCTION

In emergency situations, every second counts, and the ability of emergency services to respond quickly and efficiently can be the difference between life and death. One of the major challenges that emergency responders face is navigating through traffic and arriving at their destination as quickly as possible. Traffic congestion and stoplights can cause significant delays, which can be detrimental to emergency response times.

Traffic Light Priority for Emergency Vehicles is a system designed to help overcome this challenge by providing a way for emergency vehicles to have priority at traffic signals. When activated, the system can override the standard traffic light pattern, allowing emergency vehicles to pass through the intersection safely and efficiently.

In this paper, we will explore the implementation of Traffic Light Priority for Emergency Vehicles, including the benefits, challenges, and potential solutions for this technology. We will also examine the impact of traffic light priority on traffic flow and the overall efficiency of emergency response systems. Finally, we will discuss the potential future developments of this technology and its potential to improve emergency services globally.

1.1 Problem Statement

Emergency services play a critical role in saving lives and ensuring public safety, and their ability to respond quickly and efficiently is of utmost importance. One of the major challenges faced by emergency responders is navigating through traffic and reaching their destination promptly. Traffic congestion and stoplights can cause significant delays, which can be detrimental to emergency response times.

As a result, there is a need for a solution that allows emergency vehicles to have priority at traffic signals to reduce response times and improve the safety of both emergency responders and the public.

The problem statement of this study is to explore the implementation of traffic light priority for emergency vehicles, including the benefits, challenges, and potential solutions for this technology, and its impact on traffic flow and the overall efficiency of emergency response systems.

2. CONVENTIONAL SYSTEMS

2.1 Manual Controlling.

Manual controlling as the name suggests requires manpower to control the traffic. The traffic police are allotted a required area to control traffic. The traffic police carry sign board, sign light and whistle to control the traffic.

2.2 Automatic Controlling

The automatic traffic light is controlled by timers and electrical sensors. In a traffic light, a constant numerical value is loaded in the timer. The lights are automatically getting ON and OFF based on the timer value.

2.3 Electronic Controlling

Another advanced method is placing some loop detectors or proximity sensors on the road. This sensor gives data about the traffic on the road. According to the sensor data the traffic signals are controlled.

2.4 Drawbacks of Conventional Systems

The manual controlling system requires a large amount of manpower. Conventional traffic lights use a timer for every phase, which is fixed and does not adapt according to the real-time traffic on that road. Electronic sensors/proximity sensors or loop detectors have less accuracy and require a high budget.

3. PROPOSED SYSTEM

The proposed system for traffic light control using machine learning for ambulance detection can detect an ambulance in real-time using image processing techniques. The system will consist of a camera mounted at the intersection, which will capture the live video feed of the intersection.

The captured images will then be processed using a pre-trained deep-learning model, which will detect the presence of an ambulance on the scene. Once an ambulance is detected, the system will communicate with the traffic light controller and change the traffic light to green, allowing the ambulance to pass through the intersection quickly and safely.

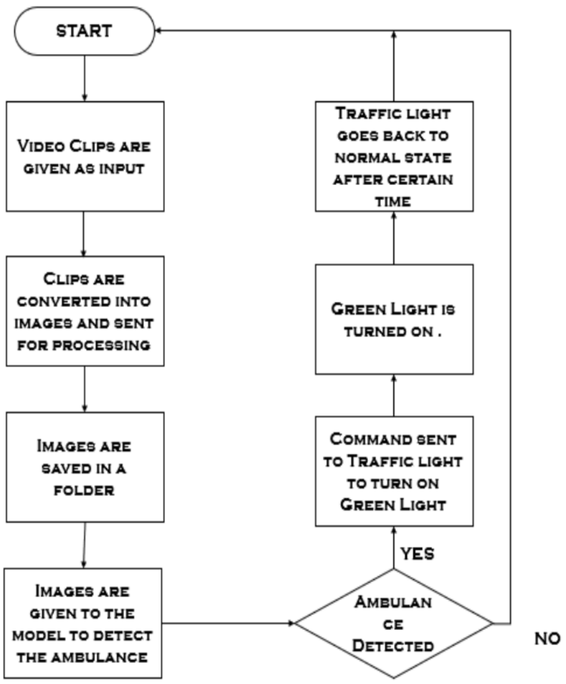

Fig-1: Flow Chart of Proposed System

4. METHODOLOGY

Block 1: Importing Libraries

First, the libraries are imported to train the model and perform other tasks such are splitting the images into training sets and test sets, where this is done by using a module named train_test_split function which is a sub-module in the sci-kit-learn module respectively.

Some other modules include config which is used to handle variables and constants across the code. Next is NumPy which is used to perform numerical calculations in the code.

The path module is used to locate the location of any folder or images in the main folder. The pickle module is used to store the processed data in a particular folder.

VGG16 is a convolutional neural network model. The architecture of the VGG16 model is based on the idea of using very small (3x3) convolutional filters with a stride of 1, which allows the model to learn more complex features and patterns in the images.

Math-plot lib is another module that is used to plot the graph of the result provided by the machine learning model. Here results refer to the accuracy and loss of the model which recognizes the ambulance in the given image.

Apart from these modules many other modules and sub-modules are being used in the machine learning model.

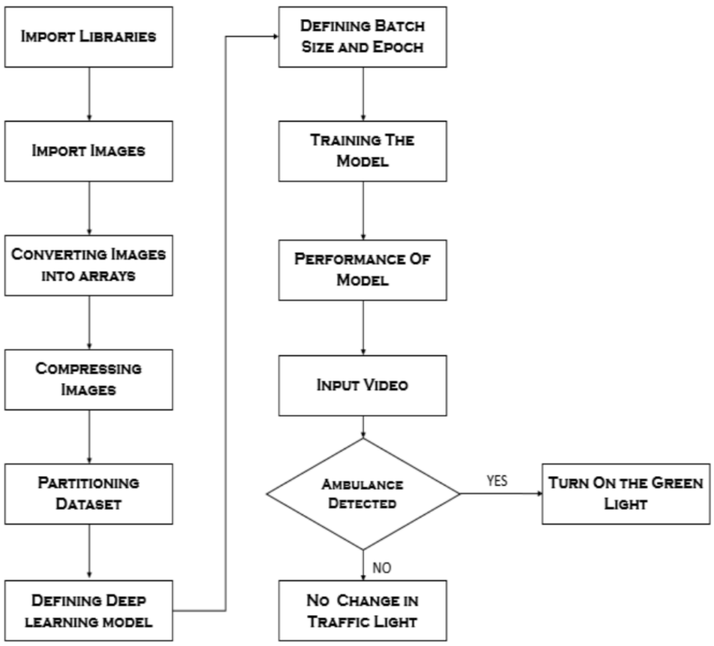

Fig-2: Flow Chart of Methodology

Block 2: Importing Images

Here is block 2 of the flow chart of methodology, the images are imported into the code. These images of ambulances are taken from different countries, different types of ambulances, from different angles and scenarios so that when the model is made to identify a given image, it can identify it easily and in turn, it increases the accuracy of the machine learning model.

Block 3: Converting Images into Arrays

Here in block 3, the images which are being stored in the folder are pushed into the arrays. First empty arrays are initialized using the config library and then using some of the loops the images are pushed onto those arrays, which are used to train the machine learning model. Through this, the model learns the patterns of the ambulance one by one.

Block 4: Compressing Images

The next step is Compressing the images. Here the input pixel intensities are scaled from the range [0, 255] to [0, 1] using the NumPy array division operation. The data array is converted to a float32 data type, while the labels array remains as it is.

Converting the data and labels to NumPy arrays and scaling the input pixel intensities from [0, 255] to [0, 1] are important preprocessing steps in preparing the data for use in a machine-learning model.

Scaling the input pixel intensities to the range [0, 1] is also important as it can help improve the performance of the machine learning model. This is because large input values can cause the model to converge slowly, while small input values can lead to numerical instability. By scaling the input pixel intensities to a smaller range, the model can learn more effectively and converge faster.

Block 5: Partitioning Images

The data is partitioned into training and testing splits using the train_test_split function from the sci-kit-learn library. The function splits the data into two sets - one for training the model and another for testing the model's performance. The size of the testing set is set to 10% of the total data using the test_size parameter, and the remaining 90% of the data is used for training the model.

The random_state parameter is set to 42, which ensures that the random splitting of the data is reproducible, and the shuffle parameter is set to True, which shuffles the data before splitting it into training and testing sets.

Block 6: Defining the Model

It starts by loading the pre-trained VGG16 model, which was trained on the ImageNet dataset, and freezing all its layers so they will not be updated during the training process. Then, the code flattens the output of the VGG16 model and constructs two fully connected layers: one for predicting the bounding box coordinates of the object in the image, and the other for predicting the class label of the object.

The bounding box output layer consists of 4 neurons, and uses the sigmoid activation function, while the class label output layer consists of a number of neurons equal to the number of classes in the dataset and uses the SoftMax activation function. Dropout layers are also added to reduce overfitting. The final model takes an input image and outputs both the predicted bounding box coordinates and the predicted class label of the object in the image.

Block 7: Training the model

This code creates a Keras model with two inputs and two outputs. The inputs are images of sizes (224, 224, 3), which will be passed through a pre-trained VGG16 convolutional neural network. The output of the VGG16 network is then passed through two separate fully connected layers: one to output the predicted bounding box coordinates, and another to output the predicted class label.

Block 8: Performance of the Model

The model has two outputs - one for the class label and another for the bounding box coordinates. The loss function for the class label output is categorical cross-entropy and for the bounding box output, mean squared error is used.

Finally, two dictionaries are constructed, one for the target training outputs (class label and bounding box) and another for target testing outputs (class label and bounding box). These will be used during the training and evaluation process of the model. The summary of the compiled model is also printed.

Block 9: Input Video

Now the video will be provided as input to the model. So as our model is trained to detect the ambulance using images, we must convert the video into images. This can be done by converting the video into frames so that the model can detect the ambulance and turn the red light into green light.

Block 10: Ambulance Detected?

After processing the images, we end up with the decision block where if the ambulance is detected the traffic light will turn green and the traffic will be released. If the ambulance is not detected in the image, then the traffic lights work normally without any changes.

5. RESULTS

So, as our model is trained with the ambulance images, and when the video is provided to it, it detects the ambulance in the frame and draws a boundary box, and gives a label as an emergency if an ambulance is detected in that given boundary box. If an ambulance is not detected in that given boundary box, then a class label names non-emergency will be given. To represent this on a graph we have plotted the accuracy and loss of the model with respect to a number of iterations or epochs. The graphs for accuracy and loss are mentioned below:

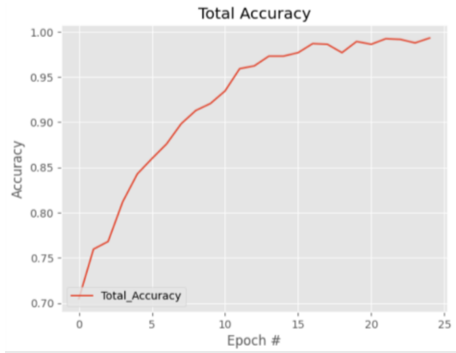

Fig-3: Total Accuracy

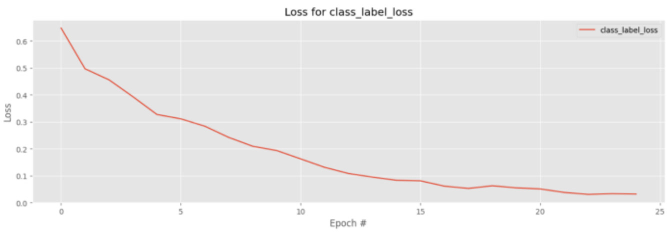

Fig-4: Loss for Class Label Loss

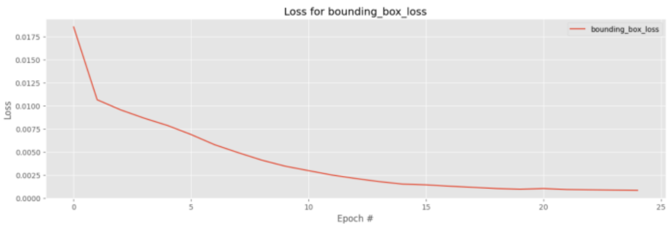

Fig-5: Loss for Boundary Box Loss

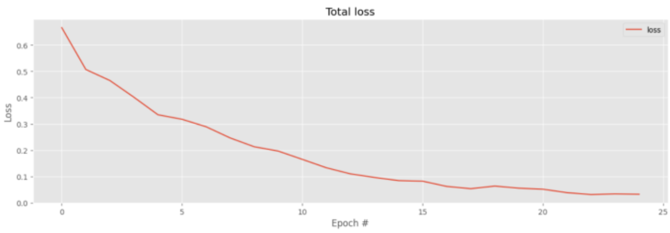

Fig-6: Total Loss of the Model

Fig-7: Ambulance Detected (Yellow Color)

Fig-8: Ambulance not detected.

Fig-9: Simulation of Real-World Traffic using Pygame

Simulation: Here we implemented the same in a simulation, where we considered real-world traffic. The ambulance will appear in any of the four lanes randomly and when it comes to the traffic signal, the traffic light turns from red to green for the respective lane.

As we can observe from Fig-3, the accuracy increases as the number of iterations or epochs increases from 0 to 25. It includes both the accuracy of the boundary box and the class label. We can say that number of epochs is directly proportional to the accuracy of the model.

On the other hand, we can see the loss of the model. It basically tells us that the model has failed to identify the emergency vehicle or the ambulance. It includes the loss in boundary box identification and naming the class label, respectively. The boundary box loss is represented in Fig-5 and class label loss is represented in Fig-4, where the loss is drawn against the number of epochs or iterations.

From Fig-7 we can observe that the model has drawn a boundary box and given a class label as an emergency as an ambulance is detected in the frame.

From Fig-8 we can see that no ambulance is detected, so a class label as non-emergency is given by the model. The accuracy depends on the number of epochs. In this case, we have considered a total of 25 epochs. So, we can conclude that the accuracy of this model lies between 80%-85%.

6. ADVANTAGES OF THIS SYSTEM

Faster response time: With the priority given to emergency vehicles, they can reach their destination much faster, which is critical in life-threatening situations.

Increased safety: By allowing emergency vehicles to pass through intersections without stopping, the risk of accidents is reduced. This is because emergency vehicles often must navigate through heavy traffic and allowing them to pass through intersections without stopping reduces the risk of collisions.

Improved response times for other emergency vehicles: By reducing the response time for one emergency vehicle, other emergency vehicles can also respond more quickly. This is because emergency vehicles often operate in teams, and by reducing the response time for one vehicle, the overall response time for the team is also reduced.

Improved efficiency: By reducing the time it takes for emergency vehicles to reach their destination, emergency services can operate more efficiently, which can lead to better outcomes for patients.

Increased public safety: By allowing emergency vehicles to reach their destination faster, the public is also safer. This is because emergency vehicles are often responding to life-threatening situations, and by reducing the response time, lives can be saved.

7. FUTURE SCOPE

Real-time tracking: The current model uses static images to detect and classify traffic lights. In the future, the model could be extended to work with real-time video streams, allowing for continuous monitoring and analysis of traffic lights.

Integration with GPS data: GPS data can be used to track the position of emergency vehicles in real time. By integrating this data with the model, the system could automatically detect when an emergency vehicle is approaching a traffic light and prioritize the signal accordingly.

Multi-object detection: In addition to detecting and classifying traffic lights, the model could be extended to detect other objects on the road, such as other vehicles or pedestrians. This could help emergency vehicles navigate through traffic more safely and efficiently.

Integration with autonomous vehicles: As autonomous vehicles become more prevalent, the system could be extended to work with these vehicles. For example, an autonomous emergency vehicle could use the model to detect and prioritize traffic lights as it navigates through city streets.

Expansion to other emergency services: The system could be expanded to work with other emergency services, such as police or fire departments. By providing real-time traffic light prioritization, emergency services could more quickly and safely respond to emergencies.

8. CONCLUSION

In conclusion, the implementation of a traffic light priority system for emergency vehicles using machine learning can greatly improve the response time and safety of emergency services. The system can accurately detect and classify emergency vehicles and provide them with priority at intersections, reducing the risk of accidents and increasing the efficiency of emergency services. With further development and improvements in technology, this system can be integrated into existing traffic management systems, making it an essential tool for emergency services around the world.

9. REFERENCES

[1] Zhang H., He Y., Wang Y. (2021) "Real-Time Traffic Light Priority Control for Emergency Vehicles: Recent Advances and Future Challenges." IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 1, pp. 358-377.

[2] Vashishtha, S., Aggarwal, S., Gupta, V., & Pandey, P. (2020). Intelligent traffic signal system for emergency vehicles. International Journal of Innovative Technology and Exploring Engineering, 9(1), 2621-2626.

[3] Javed, A., Khan, R. U. A., & Mughal, H. A. (2019). Traffic light priority for emergency vehicles using machine learning. International Journal of Advanced Computer Science and Applications, 10(8), 330-338.

[4] Hu, Z., & Wen, H. (2019). An intelligent traffic signal control system for emergency vehicle priority using machine learning. IEEE Access, 7, 13666-13678.

[5] Lin, K., Du, X., He, Y., & Jia, W. (2021). A machine learning-based traffic signal control strategy for emergency vehicles. Transportation Research Part C: Emerging Technologies, 127, 103259.

[6] Lee, S., Lee, Y., Lee, S., Lee, J., & Kim, S. (2019). Development of emergency vehicle signal priority system using machine learning. International Journal of Advanced Computer Science and Applications, 10(9), 73-77.

BIOGRAPHIES

Dr. Saneesh Cleatus Thundiyil Associate Professor, Dept. of ECE, BMSIT&M. 14 years of teaching experience. Specialization in Biomedical Signal Processing.

Anandateertha, Fourth Year Student, Dept. of ECE, BMSIT&M, Bengaluru.

Manoj Kumar M, Fourth Year Student, Dept. of ECE, BMSIT&M, Bengaluru.

Abhishek B Kamble, Fourth Year Student, Dept. of ECE, BMSIT&M, Bengaluru.

Chennakeshava Reddy K M, Fourth Year Student, Dept. of ECE, BMSIT&M, Bengaluru.