# Real Time Automated Traffic Signal Controller

[1]Manasi Adsul, [2]Isha Chhawchharia, [3]Nidhi Chauhan, [4]Kumud Wasnik

[4]Assistant Professor, [1,2,3] UG Student, Department of Computer Science and Technology,
Usha Mittal Institute of Technology, SNDT University, Mumbai, India.

***Abstract:*** With advanced technology coming into play so frequently, there is a prominent need to develop a solution to reduce traffic congestion by high volumes of vehicles, especially on urban roads. The time delay of the signal lights and the go-time is hard-coded on the signal and does not depend on the actual traffic at a given time. The development of an Intelligent Traffic Signal Control (ITSC) system is needed. The objective of the paper is to create an Automated Traffic Signal Control (ATSC) and automate the process of video surveillance in real-time. This study aims to develop a solution to control traffic congestion using computer vision and deep learning. The signal timing of each lane decided on the density of the vehicles is found using deep CNN called You Only Look Once (YOLO). Vehicles are classified using the ReLU activation function in the Alexnet model as Emergency or Non-Emergency. Based on this classification, emergency vehicles get the highest priority.

***IndexTerms* - Intelligent Traffic Signal Control, ATSC, Video-surveillance, Computer vision, Deep learning, YOLO, Deep CNN, Alexnet CNN, ReLU.**

## I. Introduction

In this developing era, people hardly have time for themselves, but most of their time is wasted in traffic. The time delay of the signal lights and the go-time is hard-coded on the signal and does not depend on the actual traffic at a given time. One impact this problem leads to is, often due to this mismanaged untimely traffic, ambulance and priority vehicles are stuck in traffic. With advanced technology coming into play so frequently, there is a prominent need to reduce traffic congestion, especially on urban roads.

Mainly, there are three types of traffic signal control strategies-

- Pre-time (predefined signal plan)
- Actuated (triggered signal control)
- ATSC (adaptive traffic signal control)

At present, the traffic signals in most of the cities work on fixed time mode. In this mode, the traffic signal changes irrespective of the volume of vehicles. It impacts the economy and is a grave danger for the environment- as many drivers keep their engines running, idly wasting fuel and energy. In an adaptive traffic signal control strategy, traffic signal changes are based on actual traffic. ATSC method promises to give better results than the pre-time signal plan. The main objective of this study is to maximize the efficiency of existing systems, by using real-time solutions, to reduce heavy traffic jams and vehicle queue length, to implement a smart traffic signal controller using real-time detection of the actual traffic of vehicles.

## II. Literature Review

One of the ways to implement an adaptive signal control strategy is by using fuzzy logic[1] where using variables such as distance of road and number of vehicles, the signal time is defined. The fuzzy controller regularly sees the traffic conditions and decide the action of the phase. Machine Learning algorithms have greatly helped in solving traffic congestion. An intelligent traffic system using machine learning techniques was proposed in[2] where they created a traffic scenario with SUMO and OSM then classified the intersections into phases as per the traffic using the SVM model and then optimized signals with high traffic to low or medium traffic zones. The system yields better results than fixed time and Vehicle Actuated Controlled (VAC) systems. With supervised learning algorithms, reinforcement learning has been used to develop a multi-agent system that learns from its environment[3][4][5]. And image acquisition using canny's edge detection algorithm and applying Q learning helped in getting better results compared to the current system[6]. A research survey shows various machine learning methods used in surveillance systems[7]. In recent research work, deep learning algorithms are used to deal with CCTV video feeds. [8]For this problem statement, CNN[9], R-CNN [10] and faster R-CNN are some methods which are also used. You-look-only-once, YOLO, [11]is an object detection algorithm used for traffic volume detection[12][13].

## III. Methodology

### 3.1 Proposed System

This system proposes a solution to reduce the traffic con- gestion and manual work behind it. We can implement this on existing controllers, where there are fewer lanes. This system has the feature of identifying emergency vehicles such as fire engines and ambulances and gives them a priority. Objects are detected using computer vision. We have used the YOLO algorithm for real-time object detection of different types of vehicles. It has around 80 classes to be classified. To classify the vehicles as emergency and non-emergency, we worked and compared three models.

1. AlexNet CNN
2. VGG-16 architecture
3. A manually created model.

AlexNet CNN gave maximum accuracy and minimum loss. Our model counts the density of vehicles in each lane using the YOLO algorithm. It then runs throughout the AlexNet model to determine the presence of Emergency Vehicles. The timer algorithm calculates timing for each lane using the density. Then the lanes are assigned priority based on the detection of Emergency Vehicles, or they open in sequence.

### 3.2 Data Collection

The procedure is gathering data and analyzing its accuracy for research purposes. We have derived our database. The dataset used is manually created of 2875 images. In this, we have 1,380 images of Emergency Vehicles and 1,495 Non- Emergency Vehicles from google. Data Augmentation is also done, which means to increase the diversity of data used for training the model without adding new data. It prevents the model from memorizing. If we have an image of an ambulance in our dataset, then its reflection, cropped, or resized image is additionally a legitimate image of that ambulance. This way, we can double our training dataset by simply flipping the image. This image set obtained is divided for the training and testing of the model. This dataset is used in the AlexNet Architecture.

### 3.3 Flow Control

#### 3.3.1 Process

The AlexNet model is trained with the ReLU activation function to classify the images as Emergency and Non-Emergency. The model generated is used in the video module back-end of the implementation.

1. The system takes you to the Main Page. Select videos for all lanes. Each video process as a thread. The timer initializes to 0.
2. YOLO detects all the images and the video module back- end classifies those images. YOLO re-runs at short, regular intervals as the camera keeps sending fresh images.
3. Based on the density of the vehicles in each lane, the time for which that lane should run is calculated using the timer algorithm. The algorithm re-initializes every 30 seconds.
4. If there are no Emergency Vehicles in any lanes, the lanes open in the normal sequence as 1 → 2→ 3→ 4.
5. However, if there is an emergency vehicle in any of these lanes, that lane is given top priority. Consider two scenarios.

a. There is an emergency vehicle in lane no. 3. The sequence of the lane that would follow is 3→ 1→ 2→ 4.
b. There are emergency vehicles in multiple lanes. The first lane with an emergency vehicle would open first and then the next. Now, assuming there are emergency vehicles in lane no. 2 and lane no. 4. The sequence would be 2 → 4→ 1→ 3.

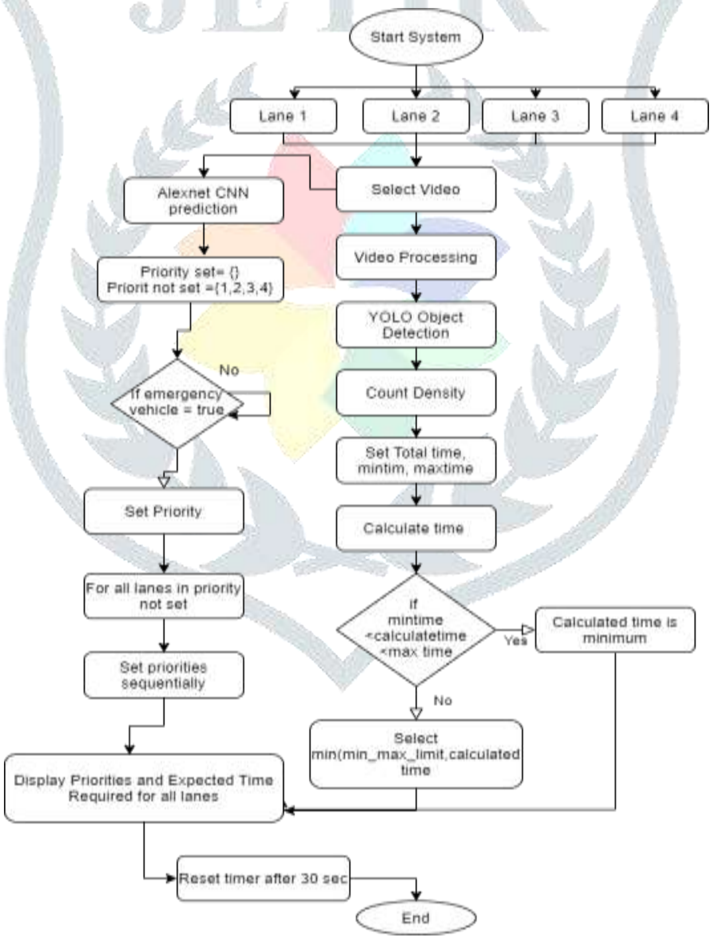


**Fig 1**: Flowchart

### 3.4 AlexNet CNN Model

In the proposed system, we have used the Alexnet CNN model to classify emergency and non-emergency vehicles. This CNN model has trained for over 1000 classes. Hence when used to classify only these two vehicles' classes, it resulted in maximum accuracy. The Alexnet model is trained and tested on the ImageNet dataset. It is to take in color images of size (224X224X3).

#### 3.4.1 Architecture

The first and second convolution layers are followed by a max-pooling layer. Following the second layer of max-pooling, there are three directly connected convolution layers. The last convolution layer is followed by one max- pooling layer. Each of the five convolution layers have an activation function ReLU (Rectified Linear Unit) to introduce non-linearity for the output to achieve a faster training time. After these eight layers, there are two fully connected layers due to which there are fixed input size. The eleventh or the final layer is also a fully connected layer that performs the same operations as previous fully connected layers. It is a dense layer of 1000 neurons which is equal to the number of classes in ImageNet. The output of the eleventh layer passes into a SoftMax activation function whose output is a vector that contains the predictions of the network. The CIFAR-10 dataset has some 60,000 color images from 10 classes each class consisting of 6000 images of 32x32 pixels. CIFAR–10 has an accuracy rate between 80-90%. This was used in the ReLU activation function.

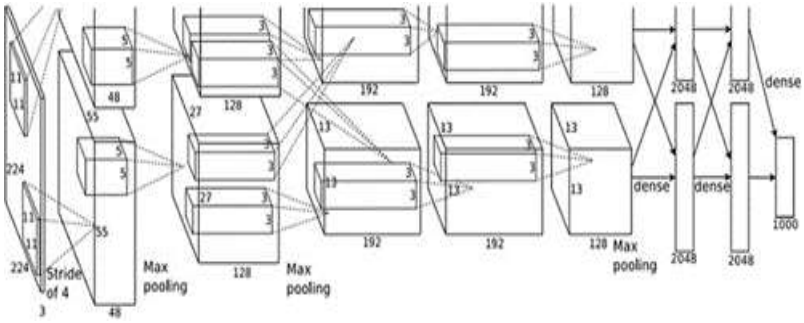


**Fig 2**: AlexNet CNN Architecture [14]

Alexnet model is trained on 2 GPUs where half neurons were on one GPU and the other half on another. Hence some layers in the architecture need to perform some cross GPU communication without going through the host machine. Max pooling layers are used to down-sample an input representation and to reduce its dimensionality. The overlapping nature of pooling harder to overfit. The rectified linear activation function or ReLU for brief could also be a piece-wise linear function that can output the input directly if it's positive, otherwise, it'll output zero. Data augmentation and dropout are two techniques used in this architecture to reduce over-fitting.

### 3.5 YOLO Technology

You Only Look Once (YOLO) is a state-of-the-art, real- time object recognition system, a new approach to object recognition. The spatial separation problem of bounding boxes and associated class probabilities A single neural network predicts bounding boxes and class probabilities directly from full images in an evaluation. Since the entire detection channel is a single network, it can be optimized directly for the board's detection performance. Earlier methods like R-CNN and its variations used a pipeline to accomplish this task in multiple steps. These can be slow to run and difficult to optimize for each component must be trained separately. YOLO does it all with a single neural network.

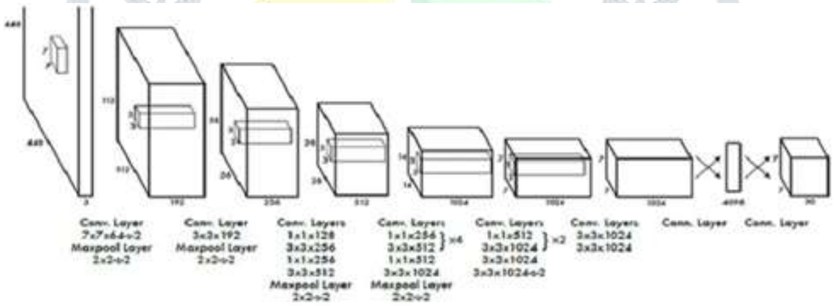

**Fig 3**: YOLO Architecture [15]

#### 3.5.1 Working

1. YOLO first processes the image passed on as input.
2. The framework divides this input into grids.
3. Each grid then goes through image classification and localization.
4. Prediction of class probabilities of objects, for their corresponding bounding boxes, is done by YOLO.

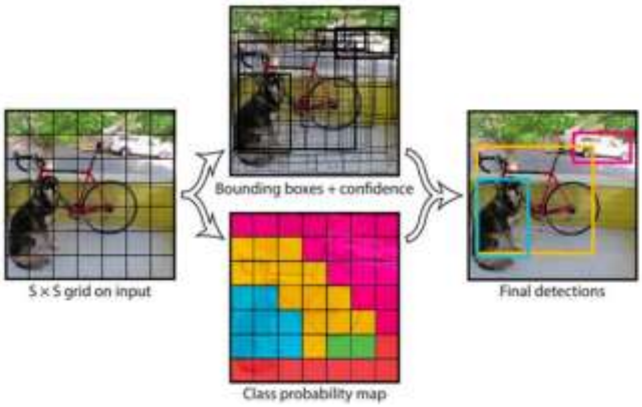


**Fig 4**: Working of YOLO [12]

### 3.6 Timer Algorithm

We have created an algorithm to calculate the signal time required for each lane. The algorithm needs the density of vehicles obtained from the YOLO Object Detection model.

### 3.6.1 Equation of Timer Algorithm

t = [((i / sum (VEHICLES_COUNT)) * TOTAL_TIME_LIMIT )
if MIN_TIME_LIMIT < [(i / sum (VEHICLES_COUNT)) * TOTAL_TIME_LIMIT] < MAX_TIME_LIMIT
else
min (MIN_MAX_TIME_LIMIT, key=lambda x: abs(x - (i / sum(VEHICLES_COUNT)) * TOTAL_TIME_LIMIT))
for i in VEHICLES_COUNT]

constants = TOTAL_TIME_LIMIT, MIN_TIME_LIMIT and MAX_TIME_LIMIT
variable i = VEHICLE_COUNT

Total time limit= The time taken to complete one cycle. Vehicle count= Total density of all vehicles at a lane.
Sum (Vehicle count) = Sum of vehicle density at all lanes.
Min Time Limit= The minimum time given to a signal.
Max Time Limit= The maximum time given to a signal.
The time signal for any lane will not exceed the range of (Min Time Limit, Max Time Limit).

The total time limit is set to 120 seconds and a minimum, a maximum time limit of 10 secs and 30 secs respectively. Assume the following vehicle counts for respective lanes.

- Lanes [1,2,3,4] - Vehicles [40,10,20,30]
- Sum (Vehicle count) = 100.
- For i-th lane, if vehicle count is 40. The equation first calculates the ratio of Vehicle count in i-th lane to the Sum(Vehicle count) i.e. (40/100).
- After multiplying it with the total time limit, we get the timeshare for i-th lane based on the density. i.e. (40/100 * 120)
- This timeshare (48sec) runs through a conditional statement that checks if it is in the range of (Min Time Limit, Max Time Limit) i.e. (10,30).
- If the calculated time isn't in the above range, then the closest absolute value is the output. i.e., 30 secs
- Following the same procedure, timeshare for all lanes is calculated: [30, 12.0, 24.0, 30]

## IV. Experimental Results

After training the data using AlexNet with the ReLU activation function, the results obtained by adding new images for testing. The input is the vehicle image, and the output is the type of the vehicle i.e., Emergency Vehicle or Non-Emergency Vehicle.

### 4.1 Graph

These images are pre-processed, scaled, convoluted, and given as test values to the system. After processing and running approximately 2900 images of the dataset, we plotted the graphs to find the training and testing accuracy and loss of the model. The results of which came about to an accuracy of 91% and an approximate loss of 20%

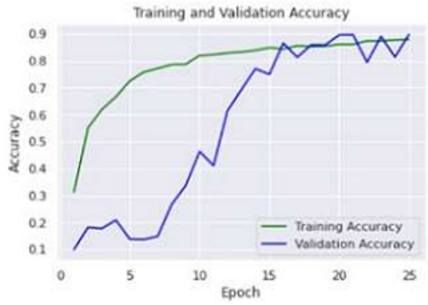


**Fig 5**: Training and Validation: Accuracy

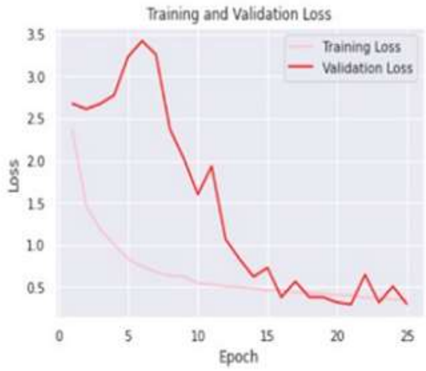


**Fig 6**: Training and Validation: Loss

### 4.2 User Interface

Our system is named real-time automated traffic signal controller or real-time ATSC, the system updates the video surveillance after every 30 seconds and automates the traffic signal timings by learning from each lane. The UI is created using the Tkinter library in python.

Various video extensions accepted in our system like mp4, MOV, WebM, Mkv, Avi. Once four videos are selected, the path displays in the containers. After successful timing calculation and the detection of emergency vehicles, this data will appear in our final user interface. The output values display Priority (P) - the sequential priority of that lane, Vehicle Count (VC) - total number of vehicles in the lane, Emergency Vehicles (EV) - number of emergency vehicles, Estimated Time (ET) - the time for which the signal will open.


**Fig 7**: Main Page

## V. Conclusion

This system uses various methods as AlexNet CNN, YOLO, ReLU. The system uses YOLO and its classes to classify the vehicles. This system successfully classifies vehicles as Emergency or Non-Emergency with an accuracy of 91% and loss of 20%. It can be used to stimulate automated traffic management in cities and for route suggestions. The throughput of the traffic is increased as the timer algorithm reduces waiting time. However, the run-time of this system is high. As compared to fixed-time controllers it is not cost-efficient. Despite this, if operated in real life, it reduces manpower. The system also achieves better results than fixed-time controllers.

Further, the system can extend to identify other classes of emergency vehicles such as police vans. A number plate recognition system can be built using the features extracted. An amalgamation of hardware sensors along with video tracking can improve accuracy. Connecting drivers of vehicles to this system and real-time route suggestions to avoid traffic congestion.

## VI. Acknowledgment

We like to express our gratitude to the supervisor, Ms Kumud Wasnik, who guided us throughout this project. We wish to acknowledge the help provided by the technical and support staff in the Computer Science and Technology department at Usha Mittal Institute of technology. We would also like to show our deep appreciation to the supervisors who helped us finalize our project. Your encouragement when the times got rough is much appreciated and duly noted. It was a great comfort and relief to know that you were willing to guide us throughout the period whenever needed. Our heartfelt thank you.

## References

**[1]** Parveen Jain, “Automatic traffic signal controller for roads by exploiting fuzzy logic,” V.V. Das, J. Stephen, and Y. Chaba (Eds.): CNC 2011, CCIS 142, pp. 273–277, 2011.

**[2]** M. Ali, G. Devi, R. Neelapu, “Intelligent traffic signal control system using machine learning techniques, ‘In Springer nature Singapore pte ltd.2021.

**[3]** P.G. Balaji X. German D. Srinivasan, “Urban traffic signal control using reinforcement learning,’IET Intell. Transp. Syst., 2010, Vol. 4, Iss. 3, pp. 177–188’

**[4]** I. Arel, C. Liu, T. Urbanik, A.G. Kohls, ‘Reinforcement learning-based multi-agent system for network traffic signal control ‘,’IET Intell. Transp. Syst., 2010, Vol. 4, Iss. 2, pp. 128–135’.

**[5]** H. Joo, S. Ahmed, Y. Lim, ‘Traffic signal control for smart cities using reinforcement learning ‘,’https://doi.org/10.1016/j.comcom.2020.03.005’.

**[6]** N. S. Jadhao and A. S. Jadhao,” Traffic Signal Control Using Reinforcement Learning,” 2014 Fourth International Conference on Communication Systems and Network Technologies, 2014, pp. 1130-1135, doi: 10.1109/CSNT.2014.231.

**[7]** G. F. Shidik, E. Noersasongko, A. Nugraha, P. N. Andono, J. Jumanto and E. J. Kusuma,” A Systematic Review of Intelligence Video Surveillance: Trends, Techniques, Frameworks, and Datasets,” in IEEE Access, vol. 7, pp. 170457-170473, 2019, doi: 10.1109/ACCESS.2019.2955387.

**[8]** M.Ali, S. Kurokawa, A. Shafie, ‘Autonomous Road surveillance system ‘,’doi: 10.1016/j.procs.2013.06.134’.

**[9]** Keemin Sohn Hyunjeong Geon, Jincheol Lee. Artificial intelligence for traffic signal control based solely on video images. Journal of Intelligent Transportation Systems, 2017.

**[10]** Songhai Zhang1 Yifan Lu1, Jiaming Lu1 and Peter Hall. Traffic signal detection and classification using attention model. Comp. Visual Media 4, 253–266(2018). https://doi.org/10.1007/s41095-018-0116-x, 2018.

**[11]** M. Mathias, R. Timofte, R. Benenson and L. Van Gool,” Traffic sign recognition — How far are we from the solution?” The 2013 International Joint Conference on Neural Networks (IJCNN), 2013, pp. 1-8, doi: 10.1109/IJCNN.2013.6707049.

**[12]** P. Shinde, S. Yadav, S. Rudrake, P. Kumbhar, ‘Smart traffic control system using YOLO ‘,’in International Research Journal of Engineering and Technology Volume: 06 Issue: 12 Dec 2019’.

**[13]** K. Zaatouri and T. Ezzedine,” A Self-Adaptive Traffic Light Control System Based on YOLO,” 2018 International Conference on Internet of Things, Embedded Systems and Communications (IINTEC), 2018, pp. 16-19, doi: 10.1109/IINTEC.2018.8695293.

**[14]** Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2017. ImageNet classification with deep convolutional neural networks. DOI: https://doi.org/10.1145/3065386

**[15]** J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection”, in proceedings of the IEEE conference on computer vision and pattern recognition, pp. 779–788, 2016.