Smart Traffic Light Control System Based on Traffic Density and Emergency Vehicle Detection

K. D. S. A. Munasinghe
Department of Electrical Engineering
University Of Moratuwa
Sri Lanka
170391l@uom.lk

T. D. Waththegedara
Department of Electrical Engineering
University Of Moratuwa
Sri Lanka
170672b@uom.lk

I. R. Wickramasinghe
Department of Electrical Engineering
University Of Moratuwa
Sri Lanka
170691g@uom.lk

H. M. O. K. Herath
Department of Transport and Logistics Management
University Of Moratuwa
Sri Lanka
oshadhik@uom.lk

V. Logeeshan*
Department of Electrical Engineering
University Of Moratuwa
Sri Lanka
logeeshanv@uom.lk

* Corresponding Author

2022 Moratuwa Engineering Research Conference (MERCon) | 978-1-6654-8786-3/22/$31.00 ©2022 IEEE | DOI: 10.1109/MERCon55799.2022.9906184

***Abstract*— Transportation is one of the main aspects of a country's economy. Most economic sectors are laid upon detrimental results due to an unorganized transportation network. This is a crucial issue faced by developing countries. There is no doubt that highways should be built in order to maximize the throughput of the transportation network; nevertheless, expansion of existing roads is also not applicable in countries like Sri Lanka due to its ceasing land area with increasing population. Thus it is essential to switch to a more efficient, technologically advanced approach to solve this issue. In addition to the typical congestion scenarios, the prevailing pandemic situation has realized the importance of prioritizing ambulances when it is caught amidst a traffic jam. Pedestrians are another vital part of the road network. Effective and safe pedestrian crossing will ensure the reduction of road accidents while improving the existing heavy traffic. A smart traffic monitoring system integrated to control the traffic signals is the ideal solution in this context. This paper proposes a smart adaptive traffic monitoring and control system to detect vehicles and pedestrians and prioritize emergency vehicles. A new Convolutional Neural Network is trained with YOLOV3 architecture to achieve 91.3% detection precision.**

Keywords— YOLO, Object detection, OpenCV, Smart traffic control system

I. Introduction

Over the past decade, the number of vehicles in Sri Lanka increased continuously. Due to this, traffic congestion can be identified as a significant problem in Sri Lankan cities. Thousands of working hours are wasted on the road due to traffic congestion. Apart from that, it causes CO2 emissions unnecessarily to the environment polluting it, wastage of fuel resulting in increased transportation costs and eventually results in stressed drivers. If traffic is handled efficiently, traffic congestion can reduce to a minimum level. Currently, there are three strategies to control traffic in a junction [1].

- Right-hand rule
- Traffic Light Control
- Manual Control

The right-hand rule is only acceptable in low-traffic situations. Traffic Light control methods and manual control by a policeman can be identified in high traffic situations. But the existing traffic light control method in Sri Lanka is a fixed time method; it does not adapt to the condition of the road. There are situations where lanes with less traffic than the other lanes are also given the same green light duration due to insufficient adaptability. Manual control by a human is not adequate. In some instances, they struggle with making decisions. Frequently, this fixed-time controlling system needs to adjust its timing after a survey due to changing the traffic pattern in the intersection [1],[2]. With the current fixed-time traffic control system, emergency vehicles are also treated as the other vehicles, where they struggle to pass the junction during the stop signal. To overcome these problems, we propose an adaptive traffic light controlling system based on traffic density on the road that uses image processing and Machine Learning to detect and count the number of vehicles in an intersection; meanwhile, the proposed system detects emergency vehicles separately to prioritize emergency vehicles, among others. An adaptive traffic signal control system can efficiently reduce traffic congestion by adjusting the traffic signal timing in response to variations in traffic patterns on the road. The adaptive traffic control system can be divided into two subsystems.

- Vehicle Detection System
- Traffic Light Control System

Under the literature review, it is discussed already existing vehicle detection models and technologies that are under research. The following section covers a detailed overview of the proposed detection model followed by the control algorithm. The final section includes details about the results and future work intended for the project.

II. Literature review

Finding solutions to eliminate the rapidly increasing traffic congestion has been vital for many researchers lately. Intelligent Transportation Systems use different sensors to get the input required [3]. Various detection methods have been studied, each having its pros and cons.

The most famous sensor systems recorded in transportation detection are the inductive loops. They depict better accuracy rates in detecting vehicles though incur higher

978-1-6654-8786-3/22/$31.00 ©2022 IEEE

installation and maintenance costs and have tedious installation processes obstructing traffic flow. The changes in the inductance of the loop are measured to identify vehicles passing on it [4]. The other detection method involves traffic monitoring cameras. Incorporating computer vision and Faster R-CNN and Tensor flow is the base of the training model. The paper further discusses how the model identifies ambulances through features like the Ambulance light, Red Cross symbol, and AMBULANCE text [5]. This system gives a lower detection accuracy due to misinterpretation of the features separately, i.e. insufficient training would immensely reduce accuracy in the detection of ambulance lights where different designs differ significantly. Some other researchers focussed on using magnetic sensors to detect the position of vehicles by analyzing the earth's magnetic field [6], [7]. When the vehicle passes over the magnetic sensors, the distortion of the magnetic field predicts the presence of a vehicle [6], [7]. As identified, these sensors show very low interference with weather conditions. Other sensor methods, including acoustic sensors, ultrasonic sensors, laser beams, and Radio Frequency Identification (RFID) techniques, have been studied and implemented; however, the detection accuracy is not adequate to control the traffic flow efficiently [3].

On the other hand, different implementation techniques have made the detection models witness their performance in real-time. System implementation using Raspberry Pi integrated with the Internet of Things (IoT) was proposed in [8]. Real-time images are sent to and retrieved through an M2X cloud platform, and digital processing was done in MATLAB. Here black and white pixel differentiation technique in grayscale images were used to get the vehicle density. In addition, an adaptive traffic light control optimization algorithm is introduced using reinforcement learning (RL) [9]. Under this research, the performance of the RL model was analyzed in an extended road network and city network, explaining how co-learning improves performance. Though they were able to sufficiently model the traffic conditions with this project, drawbacks may occur in implementation as real-time data processing was not done.

Currently, there are many traffic management systems installed around the world. In Sri Lanka, most of the junctions are equipped with a fixed-time traffic light control system. With the increase in urban traffic, this method has become obsolete globally. Several types of research have been carried out within Sri Lanka to find an optimum solution for the increasing traffic. Researches focus on the fact that Google maps display the traffic population in different colors [10] including other speed detection techniques to identify vehicles[11]. It can be used to identify the lanes with high traffic density and thereby control the traffic lights. This method has drawbacks in identifying special events like emergency vehicle detection. Therefore, our proposed system is likely one of the best possible options for optimizing the traffic conditions in the country with a fully automated and adaptive system.

III. Traffic detection

A. *Object Detection Algorithm*

YOLO V3 is used as the object detection algorithm [12]. The YOLO algorithm deals with object detection differently from other commonly used object detection algorithms. It takes an entire image in a single instant and detects all the objects in the image in one forward propagation. The biggest advantage of the YOLO algorithm is its superb speed. It can deal with up to 45 FPS and real-time object detection. OpenCV is the most commonly used software in image processing [13], [14], with backend coding done in Python.

B. *Training the Model*

Darknet is the deep learning framework of this model. Object detection has been done under nine different classes. Those classes are car, van, bus, truck, three-wheeler, motorcycle, bicycle, ambulance, and pedestrian. These are the most commonly seen vehicle types in Sri Lanka, and that's the reason behind selecting only these types of vehicles. Also, after training for these types of vehicles, the model can identify any type of vehicle under these types. The system can calculate weighted traffic density by separating vehicles into vehicle types. As an example, in weighted traffic density calculation, buses should get more weight than cars. The model was trained using more than 25,000 images, out of which 80% were used for training and remaining 20% of the images were used for testing of the model. Those images were mainly taken from the COCO data set from Microsoft [15]. Six data classes; car, bus, truck, motorcycle, bicycle, and pedestrian were extracted from the COCO data set consisting of 80 data classes while van, three-wheeler, and ambulance images were externally added to the training data set from local images.

After training the primary data set, the cropping and relabelling process was done to increase the model's accuracy. "LabelImg" git hub repository was used for this process. In this process, wrongly detected objects are corrected by manual method. This helps to increase model accuracy immensely.

CCTV cameras of the junction take inputs as videos, and video frames are given into the object detection algorithm as images. Reducing the frame rate of the algorithm resulted in a high speed of object detection. Training details of the model are provided in Table I. The activation function, leaky ReLU used here was computed as in (1) [16]. Here, x is the training parameter and α is a very small constant value to give the positive slope for x<0. For updating the weights during training an optimizer is used. For convergence with the loss function, stochastic gradient descent with momentum is used as the optimizer [17].

$$f(x) = 1(x < 0)(\alpha x) + 1(x \geq 0)(x) \quad (1)$$

TABLE I. Data of Training criteria

Batches	64
Subdivisions	16
Window width	608
Window height	608
Learning rate	0.001
Activation function	leaky ReLU
Completed training iterations	10,000

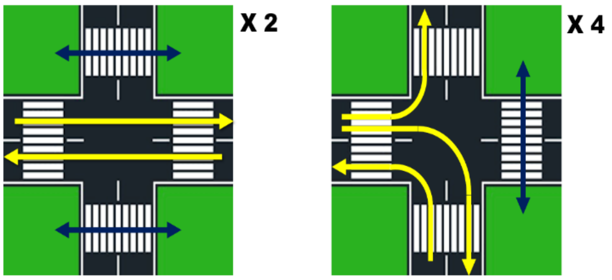
IV. Traffic Controlling

As this model can detect all the vehicles and pedestrians in the junction, we have developed different controlling algorithms instead of existing ones. Fixed time-based algorithms and fuzzy logic-based algorithms are the commonly used controlling algorithms for traffic control [14].

The controlling algorithm can be adapted to any intersection or junction and can be slightly changed according to the intersection. Here, we use a simple 4-way junction with four pedestrian crossings to implement this project. For our convenience, we labelled the junction's lanes as follows (Fig.1).

Fig. 1 – Lane labels of the 4-way junction

Here control algorithm will change the junction's state into possible scenarios according to traffic volume calculation [18]. A scenario shows what the traffic lights' colors are at that scenario and which vehicle and pedestrian paths are allowed to cross through the junction. There are 26 different possible scenarios for this type of 4-way junction. The following figure shows the type of junction that we are going to implement, and it shows two possible scenarios. Yellow arrows indicate paths allowed for vehicles, and dark blue arrows indicate paths allowed for pedestrian crossings.



(a) (b)

Fig. 2 – Types of possible scenarios in a 4-way junction

In Fig. 2 there are two possible scenario types (Fig. 2 – a and Fig. 2 – b) in a 4-way junction. The number of possible similar scenarios is shown with a multiplication mark. Therefore, Fig. 2 - a shows two possible scenarios and Fig. 2 – b shows four possible scenarios as this is a 4-way junction.

The traffic controlling algorithm takes each lane's vehicle count and pedestrian count from the detection algorithm as inputs. And it outputs a scenario for the junction according to calculated traffic volume. A scenario contains that which are the colour of each traffic light in the intersection, and traffic will control according to that. There are 12 different traffic lights for vehicles. Each has three different colours (Red – Stop, Yellow – Wait, Green – Go) and four separate traffic lights for pedestrians, with each having two different colours (Red – Stop, Green – Go).

We can divide this traffic controlling part into four different priorities as shown in Fig. 3. Those are,

- Priority Vehicle – Here, we mainly considered ambulances, as our detection model can detect ambulances. If an ambulance comes to the junction algorithm ignores all other priorities and allows the ambulance to go through the junction by changing the junction's state into an appropriate scenario.
- A vehicle with exceeding maximum waiting time – If there is no ambulance in the junction algorithm looks at whether there is a vehicle with exceeding maximum waiting time. (Pre-defined value for all vehicles. This helps to eliminate the error of vehicles staying at the junction for a long time under standard traffic control.) If there is that kind of a vehicle, then the algorithm switches the state into a scenario that can pass that vehicle from the junction.
- Pedestrians with exceeding maximum waiting time – If there are no above two priorities, the algorithm checks whether a pedestrian is exceeding maximum waiting time. This is also the same as the maximum waiting time for vehicles. If there is that kind of pedestrian, then the algorithm switches the state into a scenario that can pass that pedestrian from the junction. (We can choose maximum waiting times according to the junction that we implement the system. Here, we use 300 seconds as the maximum waiting time for vehicles and pedestrians.)
- Normal traffic controlling – If there are no above three priorities, the algorithm will switch to standard traffic control. Here highest traffic volume paths get priority and switch the scenarios accordingly. This helps to pass the maximum number of vehicles and pedestrians from the junction at a given time.

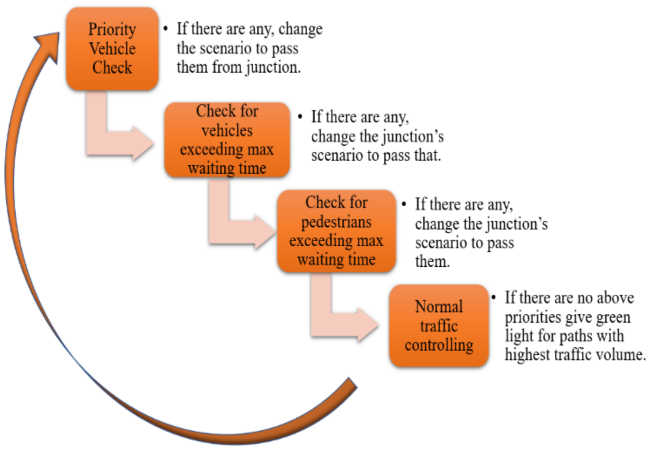


Fig. 3 – Simple illustration of traffic controlling algorithm

The green light calculation for a given path is done as follows,

$$Traffic\ Volume = \frac{number\ \ of\ vehicles}{area\ of\ the\ lane\ concidering} \quad (2)$$

$$Max.Traffic\ Volume = \frac{Max.\ number\ of\ vehicles}{area\ of\ the\ lane\ considered} \quad (3)$$

$$Max.Green\ Light\ ON\ Time = \frac{Max.\ considered\ length}{average\ vehicle\ speed} \quad (4)$$

When the number of vehicles in the lane is less than the maximum number of vehicles considered,

$$Green\ Light\ ON\ Time,$$

$$= \frac{\frac{number\ \ of\ vehicles}{number\ \ of\ lanes} X\ average\ length\ of\ a\ vehicle}{average\ vehicle\ speed} \quad (5)$$

The controlling algorithm calculates the traffic volume of a specific lane according to (2). This will help the algorithm identify which lanes have the highest traffic volume and, according to that algorithm can decide which paths should be green. Meanwhile, it calculates maximum traffic volume as shown in (3), to check whether the existing traffic volume exceeds that level. If so, the applicable green light time for that situation is the maximum green light time as shown in (4). Otherwise, normal green light on time, which can be calculated using (5) is applicable to the situation.

However, after setting the green light time for one scenario, the system will automatically check whether there are any paths with zero traffic density. If there are any paths with zero traffic density, system will switch junction's scenario automatically into another suitable scenario. Due to this reason, we cannot display how many seconds the green light or red light will last. This is one of the drawbacks of our system. By removing this option, the time limit for green and red light can be displayed even though the system efficiency will reduce.

As an example, assume that the junction is on scenario shown in Fig. 4 with five vehicle paths in green (L1 to L6, L1 to L7, L1 to L8, L2 to L7 and L4 to L5) for 40 seconds. (Lane numbering are as in Fig. 1)

Fig. 4 – An example scenario

But assume that after 20 seconds, the system detects zero traffic density in the vehicle paths of L1 to L6 and L2 to L7. Since turning on the green light for the next 20 seconds is unnecessary, the system will change the scenario into another scenario that can avoid this waste of time and control the traffic flow effectively.

In the new scenario depicted in Fig. 5, L1 to L6 and L2 to L7 vehicle paths are red with zero traffic density. Instead, the L2 to L6 pedestrian path is now green. Therefore, within the next 20 seconds, additional people can cross the junction. This will improve the efficiency of the system immensely.

- Therefore, our system's controlling algorithm always tries to pass the maximum number of people from the junction and avoid many inconveniences happening in the existing fixed time method in Sri Lanka.

Fig. 5 – An example scenario

V. Results and Discussion

Once the object detection algorithm processes the frames, it displays the count in each frame and displays the image with bounding boxes around each identified object as in Fig. 6, achieved using OpenCV. According to the program implemented in the system, it can get the total count of the vehicles and pedestrians. Also, it is possible to get the separate vehicle counts based on the Class Ids (car, van, bus, truck, three-wheel, motorcycle, bicycle, and ambulance) that we used. This is depicted in Fig. 6 – a. The model was trained to fit Sri Lankan context and trained with images from the community. The Sri Lankan transportation system can be distinguished by the buses, threewheelers and motorcycles. As displayed in Fig. 6 – b, our model detects these vehicle categories with greater accuracy. Our primary concern in training the model was to implement it to identify the ambulances with higher priority, which we were able to achieve. Fig. 6 – c shows how our model effectively differentiates ambulances and pedestrians. Blue bounding boxes are used for ambulances, while yellow boxes are used for persons.

The accuracy of the model was calculated using the mean Average Precision formula (mAP) in (6) and is shown in Table II. Varying accuracy for different vehicle types is a result of the angle of capture of image which hinders the unique identity of each type. It is recommended to re-categorize the classes as ambulance, person and other to increase the accuracy in such situations.

TP: True positives
FP: False positives

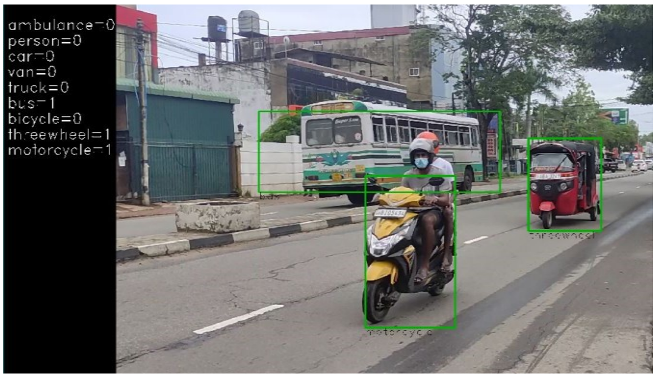
$$mAP = \frac{1}{|classes|} \sum_{c \in classes} \frac{|TP_c|}{|FP_c| + |TP_c|} \quad (6)$$

Table III provides a summary of the whole model output. Accordingly it is clear that we have achieved a significant precision value. The model requires further tuning to improve its speed in retrieving the output.

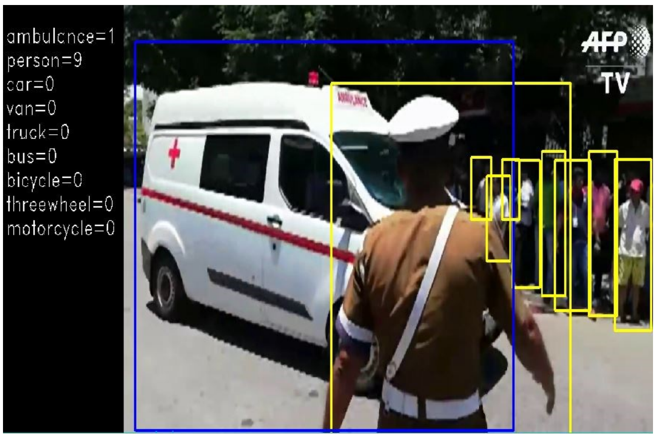
TABLE II. CLASS ACCURACY

Vehicle type/ Person	Accuracy
Ambulance	98.04%
Person	83.13%
Car	90.37%
Van	93.55%
Truck	96.67%
Bus	91.57%
Bicycle	88.82%
Threewheeler	92.75%
Motorcycle	86.67%

(a)



(b)



(c)

Fig. 6 – Vehicle detection model output

This paper presents a vehicle detection model that can analyze the traffic density at a particular junction and the control algorithm. Traffic flow at a junction depends upon the priority vehicles, illegal parking, pedestrians, unpredictable accidents, etc. Thus, getting the surveillance videos in real-time for traffic density analysis is necessary. The proposed model uses computer vision to process live video feeds and accurately classify eight vehicle types and pedestrians. Currently, for training, we are using the CCTV footage obtained directly from the junctions in Sri Lanka to improve the model accuracy and calibrate the model to a live situation. At preliminary training stage, videos were captured through the camera of smart phone with a resolution of 1080p. Videos were recorded at Golumadama Junction, Dehiwela Flyover and Piliyandala Junction.

TABLE III. IMPORTANT RESULTS OF THE MODEL

Time to load model	0.375 s
Average time to read one frame	3.86×10^{-6} s
Time to set input	0.0488 s
Time to forward pass	1.43 s
Time to post-process	0.105 s
The average loss of the model	0.3
Mean average precision	91.30%
Ambulance detection precision	98.04%

Meanwhile, we are in the stage of integrating the detection model for simulation and validation before actual implementation. For this purpose, we have identified the Simulation of Urban Mobility (SUMO) software as a powerful application with efficient features for traffic simulation [19]. SUMO is open-source software that allows numerous user-friendly options to build our own network by utilizing their inbuilt tools. The software has an excellent visualization mode enabling the user to analyze traffic data accurately. It has a default method to output important criteria like vehicles count and waiting times. Unlike other traffic simulation software, SUMO provides a pedestrian simulation platform as well to depict the real-world scenario. It has an inbuilt Google map on which the user can select the location needed and develop the traffic network per location selected. Traffic light inspection, re-routing vehicles, and traffic forecast are a few characteristics available in SUMO that are useful in our project.

In addition to the proposed model in this paper, as an extension, we will be integrating the model with a traffic light control system simulated using SUMO to effectively control the signal duration, replacing the fixed-time system currently available with a smart adaptive one.

REFERENCES

[1] A. Vajeeran, G. L. D. I. De Silva, "Delay Analysis in a Signalized T Intersection," 2017 MERCon, pp. 325- 330.

[2] R. A. P. U. S. Perera, H. M. O. K. Herath, B. Ayesha, T. Sivakumar, A. S. Kumarage, A. S. Perera, "Evaluation of Manual and Video-Based Automated Classified Vehicle Counting Methods for Heterogeneous Traffic flow," Proceedings of the Eastern Asia Society for Transportation Studies, Vol.13, 2021.

[3] J. G. Ibanez, S. Zeadally, J. C. Castilla, "Sensor Technologies for Intelligent Transportation Systems," Sensors 2018, 18, 1212, April , 2018.

[4] Ki, Y. K.;Baik, D. K.; Vehicle Classification Algorithm for Single Loop Detectors Using Neural Networks; IEEE Transactions on Vehicular Technology, vol. 55, no. 6, November, 2006.

[5] T. Sarapirom and S. Poochaya, "Detection and Classification of Incoming Ambulance Using Artificial Intelligence," in ECTI - CON - 2021 - Smart Electrical Systems and Technology, 2021.

[6] M. Bugdol, M. Krecichwost, Z. Segiet, P. Kasperek "Vehicle Detection using Magnetic Sensors," Transport Problems/ Problemy Transportu, vol. 09, issue 1, 2014.

[7] H. Dong, X. Wang, C. Zhang, R. He, L. Jia and Y. Qin, "Improved Robust Vehicle Detection and Identification Based on Single Magnetic Sensor," in IEEE Access, vol. 6, pp. 5247-5255, 2018, doi: 10.1109/ACCESS.2018.2791446.

[8]] A. Frank, Y. S. K. Al Aamri and A. Zayegh, "IoT based Smart Traffic density Control using Image Processing," IEEE, 2019.

[9] M. Wiering, J. v. Veenen, J. Vreeken and A. Koopman, "Intelligent Traffic Light Control," Intelligent Systems Group, Netherlands, 2004.

[10] K. T. Y. Mahima, R. A. B. Abeygunawardana, T. N. D. S. Ginige, "Dynamic Traffic Light Controlling System Using Google Maps and IoT," Dec. 2020 [Online] Available at: https://www.researchgate.net/publication/347944809_Dynamic_Traffic_Light_Controlling_System_Using_Google_Maps_and_IoT.

[11] H. M. O. K. Herath, R. A. P. U. S. Perera, B. Ayesha, T. Sivakumar, A. S. Kumarage, A. S. Perera, "A comparison of Speed Data by Different Speed Detection Techniques," Proceedings of the Eastern Asia Society for Transportation Studies, Vol.13, 2021.

[12] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," arXiv preprint arXiv:1804.02767, 2018. 2.

[13] M. M. Gandhi, D. S. Solanki, R. S. Daptardar and N. S. Baloorkar, "Smart Control of Traffic Light Using Artificial Intelligence," in 5th IEEE International Conference on Recent Advances and Innovations in Engineering- ICRAIE 2020 (IEEE Record#51050), 2020.

[14] P. V., S. K. Vasudevan, K. M. Siva, B. Akshay and M. Surya, "Smart Control of Traffic Signal System using Image Processing," Indian Journal of Science and Technology, vol. 8, 2015.

[15] J. Visshwak J, P. Saravanakumar, R. I. Minu, "On-The-Fly Traffic Sign Image Labeling," in International Conference on Communication and Signal Processing, India, Jul. 2020, doi: 10.1109/ICCSP48568.2020.9182075.

[16] A. K. Dubey, V. Jain, "Comparative Study of Convolutional Neural Network's Relu and Leaky-Relu Activation Functions," Applications of Computing, Automation and Wireless Systems in Electrical Engineering (pp.873-880), Jan. 2019, doi: 10.1007/978-981-13-6772-4_76.

[17] Y. Liu, Y. Gao, W. Yin, "An Improved Analysis of Stochastic Gradient Descent with Momentum," 34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

[18] B. Zhou, J. Cao, X. Zeng, H. Wu, "Adaptive Traffic Light Control in Wireless Sensor Network-based Intelligent Transportation Systems ," IEEE,2010.

[19] P. A. Lopez, et al(2018) Microscopic Traffic Simulation using SUMO. In: 2019 IEEE Intelligent Transportation Systems Conference (ITSC), pp. 2575-2582. IEEE. The 21st IEEE International Conference on Intelligent Transportation Systems, 4.-7. Nov. 2018, Maui, USA.