iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** IX **Month of publication:** September 2021

DOI: https://doi.org/10.22214/ijraset.2021.38053

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue IX Sep 2021- Available at www.ijraset.com

Smart Traffic Light System

Rashi Maheshwari[1], Kritin Mathur[2], Rohan Jagtap[3]
[1, 2, 3]Department of Computer Science, SVKM's NMIMS, MPSTME, Shirpur

***Abstract:** Traffic signal control frameworks are generally used to monitor and control the progression of cars through the intersection of roads. Moreover, a portable controller device is designed to solve the issue of emergency vehicles stuck in overcrowded roads. The main objective of this paper is to design and implement a suitable algorithm and its simulation for an intelligent traffic signal simulator. The framework created can detect the presence or nonappearance of vehicles within a specific reach by setting appropriate duration for traffic signals to react accordingly. By employing mathematical functions and algorithms to ascertain the suitable timing for the green signal to illuminate, the framework can assist with tackling the issue of traffic congestion. The explanation relies on recent fixed programming time.*
***Keywords:** Smart Traffic Light System, Smart City, Traffic Monitoring.*

I. INTRODUCTION

Transportation collisions continue to be one of the leading causes of death and injuries worldwide. To avoid these adverse scenarios and incidents, various laws and systems have been put in place. It has been used on occasion, but eradicating and thoroughly resolving this problem remains a challenging task.

The standard traffic scheme lacks a suitable surveillance tool, which often necessitates manual traffic intersection handling. This not only causes social discomfort for commuters, but it also drains a lot of fuel due to traffic junction delays. Congestion is a big concern nowadays. Megacities are the hardest hit, despite the fact that it seems to be a normal occurrence.

Considering a road cross-section as mentioned in figure 1 below without any traffic light, just imagine the number of accidents on a daily basis. Because without any traffic lights the person driving from east to west wouldn't know what the person in the car going from north to south is planning to do, whether he or she is planning to wait and let him go, or go on with his speed. This can cause a lot of confusion hence leading to a lot of accidents. Hence the traffic lights are installed at such cross-sections, so as to prevent any mishappenings.

Figure 1. Road Intersection

But now the issue faced by the drivers is that when one cross-section road traffic light turns green it remains the same for a fixed time interval, let's say 60 seconds, even though there are only 2 cars on that particular cross-section. And the same happens with another cross-section, where the green light is turned on for 60 seconds, even though there are like 20 cars. Now, this creates a huge problem for all the people waiting at these cross-sections.

Hence, causing a delay in reaching their destinations and also a lot of wastage of fuel. And considering this particular situation in a city like Delhi or Mumbai or any other megacity, such cross-sections lie in an average gap of 5 mins. Just imagine the situation where you have to wait after every 5 mins at such a cross-section, the amount of delay caused is more than 1-2 hours, which further leads to other problems.

So in order to overcome all the traffic problems, the long waiting time, the wastage of fuel, etc, we came across a solution. According to our solution, we can have cameras installed at all these intersections and operate them

in order to decide the vehicular density and hence deciding the time of green light accordingly. And hence we have trained an ML model on supervised learning in order to get a proper result. For example, if the cross-section has one vehicle then it will turn green for maybe only 2 seconds.

II. METHODOLOGY

A. System Overview

Our system works with the help of cameras installed at each cross-section which are connected further to the internet cloud from where they can access the data for the light turning green accordingly. This majorly depends on 2 parameters,

1) The number of vehicles on that particular road.
2) The number of lanes on that road.

The more the number of vehicles the more time will be provided for the traffic to pass through, which means the green light will be on for more time as compared to when fewer vehicles are present. For example, if the cross-section has one vehicle then it will turn green for maybe only 2 seconds.

Also if the number of lanes are more than the time provided for the traffic to pass through will be less as compared to when the number of lanes are less in number. For example, if the cross-section has 5 lanes then the traffic light will turn green for maybe only 2 seconds and if there are only 2 lanes than maybe the green light time would be more than 5 seconds.

Another condition can be if there are 4 cars waiting at an intersection road with 4 lanes and another intersection road has 2 lanes. Then, the amount of time the green light will be on at the 4 lane road will be less as compared to the 2 lane road, as the 4 vehicles can easily pass through from the 4 lane road in less time. Hence this decides the amount of time the traffic light would be turned green, thus solving the problems faced by the drivers to wait for a long time at these intersections.

At the intersection, the turning on of the green light rotates in the cycle, which means first the traffic light on the north side road will turn green then the traffic light on the east side road will turn the green, and so on in the clockwise direction.

B. Hardware and Connections

The hardware configuration of the system is as illustrated in figure 2 below. The Raspberry PI 4B is connected to 4 cameras via USB. These 4 cameras are powered by POE and are Gphoto2 compatible. The Raspberry PI 4B is also connected to a GSM module to connect to the internet and access web APIs. The Raspberry PI 4B is powered using a 5V DC power supply. We are using Microsoft Azure ML studio Web API service to access the ML model online from the Raspberry PI 4B.

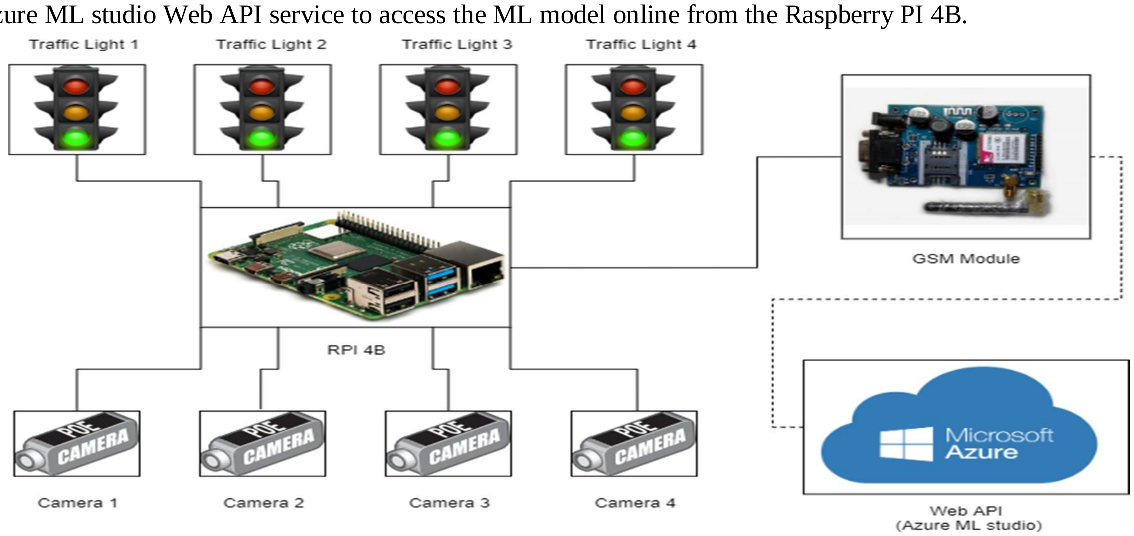

Figure 2. Hardware Configuration

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue IX Sep 2021- Available at www.ijraset.com

1) *Raspberry PI 4B:* The Raspberry Pi is a low-cost, credit-card-sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. It is a capable little device that enables people of all ages to explore computing and to learn how to program in languages like Scratch and Python. Raspberry Pi is a series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation in association with Broadcom. The Raspberry Pi project originally leaned towards the promotion of teaching basic computer science in schools and in developing countries.
2) *POE Camera:* POE-powered traffic cameras are an innovative and extremely functional use of traffic surveillance technology. They're atop traffic signals and placed along busy roads, and at busy intersections of the highway. They are Gphoto2 compatible. gPhoto is a set of software applications and libraries for use in digital photography. gPhoto supports not just retrieving of images from camera devices, but also upload and remote-controlled configuration and capture, depending on whether the camera supports those features.
3) *GSM Module:* A GSM modem or GSM module is a hardware device that uses GSM mobile telephone technology to provide a data link to a remote network. From the view of the mobile phone network, they are essentially identical to an ordinary mobile phone, including the need for a SIM to identify themselves to the network. A Raspberry Pi HAT is an expansion board that's specifically made and tested to work with a certain version of a Pi. They are pretty much the same thing as shields in the Arduino universe. However, unlike modules and other smaller expansions, HATs usually perform more demanding tasks. GSM Module will provide 4G/5G internet access to the RPI4 B using AT commands.

C. Vehicle Detection

This system consists of video cameras on the traffic junctions for each side as if it is a four way junction. Therefore four video cameras will be installed over the red lights facing the road. Cameras would be capturing video and broadcasting it to the servers where using video and image processing techniques the vehicle density on every side of the road is calculated and an algorithm is employed to switch the traffic lights accordingly.

There are separate cameras present for each road at the intersection, just below the traffic light, which are responsible to click pictures of the cars at that particular road of the cross-section. This picture is then forwarded to the Raspberry PI that connects all the cameras and also contains a python code that does the calculation of the number of vehicles present with the help of image processing. Here image processing works as an important tool to count the number of vehicles.

Take a four way traffic junction, having sides named N, E, S and W. Cameras are installed over the red light for each side named C1, C2, C3 and C4 respectively. Now we need the image of the lane when it was empty as a reference. This is done manually and required to be done only once since the same background image can be used until the structure of the road is changed.

Then the image is captured every second from the camera installed at the road(fig 3). And this image is subtracted from the image of the background inorder to eliminate the background and to get the vehicular density of the particular road.

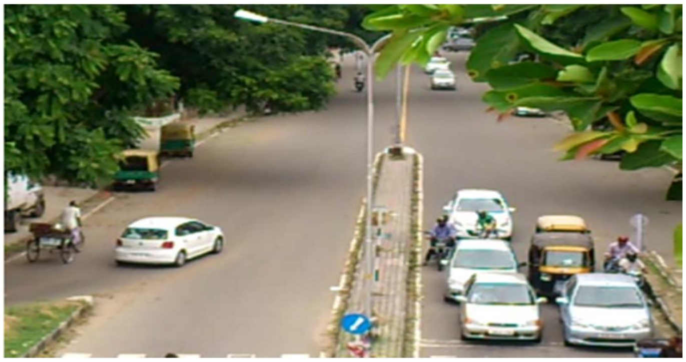

Figure 3. Camera Feed

A constant value is calculated in order to further calculate the number of vehicles is as follows:

x = Height of camera from road * Number of rows in subtracted image matrix * Number of columns in subtracted image * Number of frames per second in video

number = number / x

This provides us with the approximate number of vehicles on road considering that a vehicle larger in size will cover relatively more area and more time to pass the traffic junction. This process is repeated for every second and for all the sides.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue IX Sep 2021- Available at www.ijraset.com

D. Web API

A python script running on the Raspberry PI calls a Web API. This Web API is built using Microsoft Azure ML studio. We have used linear regression in the structuring of this Web API inorder to build our model. The model is trained using a labelled dataset which has 3 columns including Number of vehicles, Number of Lanes and the time in seconds for which the traffic light turns green and the traffic passes through the intersection. The number of vehicles is the column responsible for the Number of Vehicles present of the road captured with the help of the cameras. The Number of Lanes on the particular road is fixed. Time is the optimal time in seconds to clear traffic. Number of Lanes and Number of Vehicles are independent variables and Time is a dependent variable.

Below is the flow diagram of the experiment model on which the Web API is based.(fig 4).

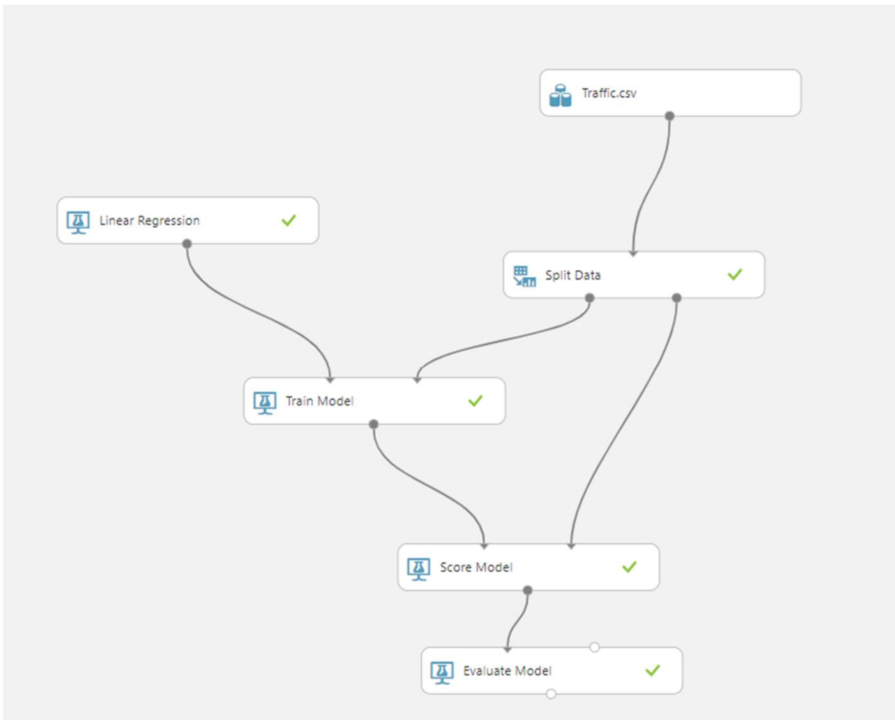

Figure 4. Linear Regression Flow Diagram

Linear Regression is a supervised machine learning algorithm where the predicted output is continuous and has a constant slope. It's used to predict values within a continuous range, (e.g. sales, price) rather than trying to classify them into categories (e.g. cat, dog).According to Figure 4, we have used 80% of the total data set as training data and the remaining 20% as testing data.

The web API is called after the number of vehicles have been detected. The web API is requested by sending the current number of vehicles on road and number of lanes on that road with http POST method. The web API returns the time in seconds for which the traffic signal needs to be green.

```
[2]:  time = ml_req(20,5)
      print(time)

15.7768429238091
```

Figure 5. Endpoint API Response

Fig 5 is an example of the endpoint API test. Here we have functionalized the API in function ml_req(). The first argument is the number of vehicles and the second argument is the number of lanes. The function returns the time in seconds for that lane.

E. Raspberry PI and Software

The Raspberry PI 4B acts as the central HUB for all operations such as Image Capture, Image Processing for Vehicle count, Requesting web API and controlling the traffic Lights. A python script on Raspberry PI 4B allows it to autonomously manage all these tasks. Raspberry PI is connected to the internet using a GSM module HAT. The Cameras are connected to the PI using USB connections. The traffic lights are connected to the PI via GPIO pins.

After a traffic light cycle is complete , the camera on the first lane clicks an image using the Gphoto2 library in python script. The script uses OpenCV library to calculate the number of vehicles. After the number of vehicles have been calculated an API request is made with 2 parameters - number of lanes and number of vehicles. The API responds with optimal time required to clear traffic when light is green. The python script controls the traffic light and turns the first lane green for the time in API response. The python script waits for that exact time and after that the script moves to the second lane and does the same process indefinitely.

III. RESULT

A. Testing Machine Learning Model

Below is the figure that shows the Mean Absolute value i.e., 1.787(fig 6(b)) and also shows the difference in time present in the dataset and the time that is predicted by the trained model(fig 6(a)).

Number_of_cars	Number_of_lanes	Time	Scored Labels
9	3	8	9.273591
6	3	5	6.814212
15	3	12	14.192349
21	1	25	21.625578
24	4	23	20.31325
30	5	25	23.974773
11	1	15	13.427648
38	6	28	29.275882
50	4	45	41.627869
4	2	7	6.431861
21	4	20	17.853871
36	4	30	30.150767

Figure 6(a). Output of the Model

Metrics

Mean Absolute Error	1.787662
Root Mean Squared Error	2.037135
Relative Absolute Error	0.196807
Relative Squared Error	0.033779
Coefficient of Determination	0.966221

Error Histogram

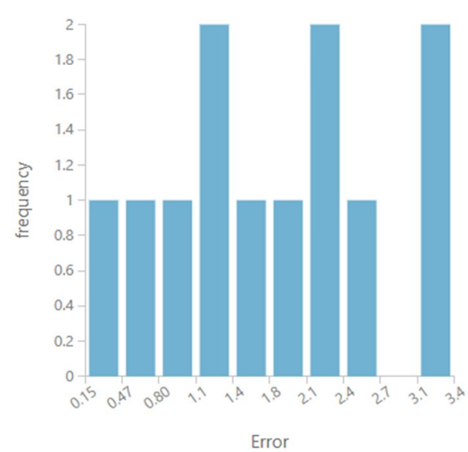

Figure 6(b). Performance of ML Model

Below is the diagram that gives the end point testing of Web API and simulation result(fig. 6(c)).It shows the directions of the roads at the 4-way intersection, i.e., North(N), East(E), South(S), West(W). This is how the traffic lights will work in the real world too, in clockwise direction, first North, East, South, West. Then it shows the number of vehicles, followed by the number of lanes, which further requests the API to give the calculated amount of time of the green signal based on the number of lanes and number of vehicles.

```
--Green lane : N
--Number of vehicle detected on Green lane: 7
--Number of Lanes : 4
--Requesting API
--Time of Green signal : 6.37676924667898

--Green lane : E
--Number of vehicle detected on Green lane: 4
--Number of Lanes : 2
--Requesting API
--Time of Green signal : 6.43186114399765

--Green lane : S
--Number of vehicle detected on Green lane: 2
--Number of Lanes : 4
--Requesting API
--Time of Green signal : 2.27780419198793

--Green lane : W
--Number of vehicle detected on Green lane: 2
--Number of Lanes : 2
--Requesting API
--Time of Green signal : 4.79227512212123
```

Figure 6(c). Simulation Result

B. *Comparison with Conventional Method*

1) Hard Coded: In the conventional traffic lights after every particular time the light switches back to red. Therefore we take a scenario where it is a junction with four sides, so at any side green light remains for 60 seconds and red for 180 seconds i.e. every side gets green light for a fixed time of 60 seconds, one after another. This is the general algorithm of hard-coded traffic systems.
2) Dynamic Coded: In the algorithm proposed by us, consider a side which is currently red. On this side we will add density of vehicles present every second (It will indirectly represent waiting time), so it keeps on getting calculated for all the sides where light is red. Now, just before 5 seconds when the green light of a lane is going to finish, we look into total density values of each lanes having red light, and the one with maximum is provided with the green signal, the density is converted to 0 for the one which was green earlier, density of other two red lights remain the same, and the process of adding up the density repeats.

The time of green signal is calculated using the number of vehicles as per density that can pass in one second. So the total density present divided by the number of vehicles that can pass in one second provides us with the amount of time for which signal is to be kept green. We will also consider that the minimum amount of green light provided to a lane must be 10 sec and maximum is 60 for practical reasons.

IV. CONCLUSION

The traffic management system proposed possesses certain advantages over the existing intelligent traffic control systems:-

A. Distance among Vehicles: One great feature is that traffic always remains at a distance. If we look at the density of vehicles at all lanes in dynamic algorithm mode and compare it to hard coded, we could see a drastic difference which would reduce congestion by almost 4 to 5 times and all vehicles can move spaciously.
B. Installation Cost of our system is very less comparatively because we require the live feed which is easily accessible from the surveillance cameras present at each traffic junction. Most major cities and the traffic junction where traffic remains high have cameras already installed.
C. Maintenance Cost of our system is virtually negligible as our system does not include any additional hardware components as compared to the other traffic monitoring systems which employ pressure mats which normally suffer the problem of wear and tear due to their placement on roads where they are subjected to immense pressure constantly.

REFERENCES

[1] Microsoft Azure for ML Model and Web API https://azure.microsoft.com/en-in/

[2] Kaggle for end point testing of Web API and simulation https://www.kaggle.com/

[3] Kumar, A., Krishnamurthi, R., Nayyar, A., Luhach, A. K., Khan, M. S., & Singh, A. (2020). A novel Software-Defined Drone Network (SDDN)-based collision avoidance strategies for on-road traffic monitoring and management. Vehicular Communications, 28, 100313. https://doi.org/10.1016/j.vehcom.2020.100313

[4] Chakraborty, P. S., Tiwari, A., & Sinha, P. R. (2015). Adaptive and Optimized Emergency Vehicle Dispatching Algorithm for Intelligent Traffic Management System. Procedia Computer Science, 57, 1384–1393. https://doi.org/10.1016/j.procs.2015.07.454

[5] Hawi, R., Okeyo, G., & Kimwele, M. (2015). Techniques for Smart Traffic Control: An In-depth Review. International Journal of Computer Applications Technology and Research, 4(7), 566–573. https://doi.org/10.7753/ijcatr0407.1014

[6] Abdoos, M., & Bazzan, A. L. C. (2021). Hierarchical traffic signal optimization using reinforcement learning and traffic prediction with long-short term memory. Expert Systems with Applications, 171(October 2020), 114580. https://doi.org/10.1016/j.eswa.2021.114580

[7] Firdous, A., Indu, & Niranjan, V. (2020). Smart Density Based Traffic Light System. ICRITO 2020 - IEEE 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions), 497–500. https://doi.org/10.1109/ICRITO48877.2020.9197940

[8] Alam, J., & Pandey, P. (Dr. . M. K. (2014). Development of Intelligent Traffic Light System Based On Congestion Estimation Using Fuzzy Logic. IOSR Journal of Computer Engineering, 16(3), 36–44. https://doi.org/10.9790/0661-16363644

[9] Busch, J. V. S., Latzko, V., Reisslein, M., & Fitzek, F. H. P. (2020). Optimised Traffic Light Management Through Reinforcement Learning: Traffic State Agnostic Agent vs. Holistic Agent With Current V2I Traffic State Knowledge. IEEE Open Journal of Intelligent Transportation Systems, 1(November), 201–216. https://doi.org/10.1109/ojits.2020.3027518

[10] Spoorthy.C, N. . (2014). Traffic Light Priority Control For Emergency Vehicle. 7–10.

[11] Ghazal, B., Elkhatib, K., Chahine, K., & Kherfan, M. (2016). Smart traffic light control system. 2016 3rd International Conference on Electrical, Electronics, Computer Engineering and Their Applications, EECEA 2016, October 2017, 140–145. https://doi.org/10.1109/EECEA.2016.7470780

[12] Joo, H., Ahmed, S. H., & Lim, Y. (2020). Traffic signal control for smart cities using reinforcement learning. Computer Communications, 154(March), 324–330. https://doi.org/10.1016/j.comcom.2020.03.005

[13] Turkey, A. M., Ahmad, S., Zaliman, M., & Yusoff, M. (2009). The Use of Genetic Algorithms for Traffic Light and Pedestrian Crossing Control. IJCSNS International Journal of Computer Science and Network Security, 9(2), 88. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.511.5913&rep=rep1&type=pdf

[14] Tomar, A. S., Singh, M., Sharma, G., & Arya, K. V. (2018). Traffic Management using Logistic Regression with Fuzzy Logic. Procedia Computer Science, 132, 451–460. https://doi.org/10.1016/j.procs.2018.05.159

[15] Rida, N., Ouadoud, M., Hasbi, A., & Chebli, S. (2018). Adaptive Traffic Light Control System Using Wireless Sensors Networks. Colloquium in Information Science and Technology, CIST, 2018-Octob, 552–556. https://doi.org/10.1109/CIST.2018.8596620

[16] Kanungo, A., Sharma, A., & Singla, C. (2014). Smart traffic lights switching and traffic density calculation using video processing. 2014 Recent Advances in Engineering and Computational Sciences, RAECS 2014, 6–8. https://doi.org/10.1109/RAECS.2014.6799542

[17] Yousef, A., Shatnawi, A., & Latayfeh, M. (2019). Intelligent traffic light scheduling technique using calendar-based history information. Future Generation Computer Systems, 91, 124–135. https://doi.org/10.1016/j.future.2018.08.037

[18] Eric, H., & Jean Baptiste, M. (2018). Impact of Intelligent Road Traffic Control in a Technologically Connected Vehicle Environment. 6(3), 265–274.

[19] Li, C., & Shimamoto, S. (2011). Dynamic Traffic Light Control Scheme for Reducing Co 2 Emissions Employing Etc Technology. International Journal of Managing Public Sector Information and Communication Technologies (IJMPICT), 2(1), 1–12. https://doi.org/10.5121/ijmpict.2011.2101

[20] Khan, N. A., Jhanjhi, N. Z., Brohi, S. N., Usmani, R. S. A., & Nayyar, A. (2020). Smart traffic monitoring system using Unmanned Aerial Vehicles (UAVs). Computer Communications, 157(January), 434–443. https://doi.org/10.1016/j.comcom.2020.04.049

[21] Khushi. (2018). Smart Control of Traffic Light System using Image Processing. International Conference on Current Trends in Computer, Electrical, Electronics and Communication, CTSEEC 2017, 99–103. https://doi.org/10.1109/CTCEEC.2017.8454966

[22] Del Vecchio, M., & Tozzi, V. D. (2010). Introduction to research. In Giornale italiano di nefrologia : organo ufficiale della Società italiana di nefrologia (Vol. 27, Issue 2, pp. 122–128). https://doi.org/10.5014/ajot.48.11.1106a

[23] Atta, A., Abbas, S., Khan, M. A., Ahmed, G., & Farooq, U. (2020). An adaptive approach: Smart traffic congestion control system. Journal of King Saud University - Computer and Information Sciences, 32(9), 1012–1019. https://doi.org/10.1016/j.jksuci.2018.10.011

[24] Al-Ani, M. S., & Al-Heeti, K. (2011). Intelligent Traffic Light Control System Based Image Intensity Measurement. September. http://repository.nauss.edu.sa/handle/123456789/55421

[25] Özkul, M., Capuni, I., & Domnori, E. (2018). Context-aware intelligent traffic light control through secure messaging. Journal of Advanced Transportation, 2018. https://doi.org/10.1155/2018/4251701

[26] Kalaiselvi, V. K. G., Sangavi, A., & Dhivya. (2017). Li-Fi technology in traffic lights. Proceedings of the 2017 2nd International Conference on Computing and Communications Technologies, ICCCT 2017, 404–407. https://doi.org/10.1109/ICCCT2.2017.7972307

AUTHORS

A. Rashi Maheshwari,
MBATech CS, SVKM's NMIMS, MPSTME and rashiirm06@gmail.com.
(Researched about 3 different methodologies including Image processing, LiFi technology and Sensor technology. Worked on the Research Paper and API)

B. Kritin Mathur
MBATech CS, SVKM's NMIMS, MPSTME and kritinmathur75@gmail.com.
(Researched about 3 different methodologies including Reinforcement Learning, UAV technology and Fuzzy Logic. Worked on the Research Paper and API)

C. Rohan Jagtap
MBA Tech CS, SVKM's NMIMS, MPSTME and rohanjagtap2000@gmail.com.
(Researched about 3 different methodologies including Calendar Based Technology, Fuzzy Logic and Video processing. Worked on the Research Paper and API)