International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 08 | Aug 2020 www.irjet.net p-ISSN: 2395-0072

INTELLIGENT LIGHTS: AN EFFECTIVE AUTOMATED TRAFFIC MANAGEMENT SYSTEM

Manoj Kumar V[1], Jeevith R[2], M Thanushree[3]

[1, 2, 3]Ramaiah University of Applied Sciences, Bengaluru, Karnataka, India

---

Abstract - *In the world full of rapid growing technologies, automated smart systems are designed to play significant role in management and prevention of traffic congestion. The current traffic management system in many urban cities are not at their best version for efficient management of traffic. Although with the current technologies, we can build advanced systems with superlative techniques, they might require expensive equipment with complex operation procedures. Hence, we propose a smart automated traffic management system which reuses the existing roadside cameras to capture traffic scenes, automatically examines the traffic and aids in controlling traffic congestion. Our approach begins with capturing of image sequence and analyzing them using digital image processing techniques. With the vehicles detected, traffic density is computed which is then fed to our designed algorithm to obtain optimized traffic signal time. By focusing on traffic density of all the roads at each instant and the mean waiting time of a vehicle at the traffic junction, our algorithm computes the green light time to be set at the traffic signal to control and manage traffic congestion. With this proposed approach of determining traffic light time, the management of traffic in urban cities can be vastly improved.*

Key Words: **Traffic Congestion, Current Traffic, Automated Traffic, Detection methods, Traffic Density etc.**

1. INTRODUCTION

Traffic congestion is one of the most vexing urban issues. Population growth together with a great demand of transportation indirectly contributes to the increasing number of vehicles on the road that will eventually cause traffic congestion. Traffic congestion has been causing many critical problems and challenges in the major and most populated cities. Increased traffic has led to more waiting time, fuel wastages, and environmental pollution. Due to congestion problems, people lose time, miss opportunities, and get frustrated. A smart automated traffic management system is essential for smooth and efficient management of traffic, which will lead us in the direction of proper investigation of traffic congestion, developing ideas to prevent congestion and alteration of traffic control management techniques. There are a number of approaches that speak about prevention of traffic congestion by alternate ways of transportation, rerouting of vehicles to prevent congestion [1], using many hardware devices with costly operations [2] and many more but none speak about changing the traffic management architecture as a whole to sense the congestion of traffic and its various ways of management.

This paper deliberates an approach on sensing the traffic using accurate detection technologies which detects vehicles under challenging circumstances and applying creative techniques and logical algorithms to manage traffic congestion. Surveillance cameras that continuously monitor traffic can be used to capture the images of the roads at the signal. At specific intervals of time the captured frames are analyzed using image processing, traffic signal times are computed and optimized using the designed optimization algorithms. The computed green light paves the way for efficient management of traffic congestion saving people's time, combating pollution and wastage of fuel caused by excessive road traffic.

1.1 Motivation:

Traffic congestion is a significant issue in urban cities of India and around the world. It is having a considerable effect on the economy of the country and mental health of the community. The magnitude of economic and public health impacts of congestion would be expected to sum up significantly across urban areas. Apart from the effects to economy from wastage of fuel, wastage of person's time and harm to the environment, people stuck in traffic are likely to become more frustrated as it will disrupt their daily routines and lifestyle. People might be delayed from reaching their destinations including work place, meeting and activities etc. Traffic congestion is seen as a threat to the productivity of community [3].

One of the root causes of traffic congestion lies in the current traffic architecture which resembles an alarm clock - timer set signal time on all roads at all conditions. It neither considers the density of the traffic on each side of the road nor compares the traffic density with other roads before setting the green light timer. The increasing negative impact of traffic congestion calls for the need to develop a new type of traffic management architecture, which works based on traffic density with logical thinking, to update the current one that lacks decision making capabilities and works on a routine.

1.2 Approach:

Density of traffic is an important decision factor for the amount of green light time to be allocated to a certain road. Using the density of traffic, we can not only sense the side of the road which can cause traffic congestion but also compare the density of traffic on different roads and vary the green light timing according to that instance of traffic. The side of the road with less vehicles can be allotted less green light time while the road with more vehicles can be allotted more time, and thus, control uneven traffic congestion. Since density of traffic plays a significant role in preventing and controlling traffic, it has been extensively used in designing our green light time optimization algorithm.

In order to monitor traffic, every traffic signal has surveillance cameras which are used to capture the frames of current traffic before the green light time is computed and set. The frames or images are used to calculate the traffic density by specific detection methods.

With an idea of the time required for the vehicles to cross the signal and density of the traffic calculated from the captured images, the amount of green light time required to clear all the vehicles of the road can be computed. But that's not enough to prevent or control the traffic congestion, the calculated green light time should be varied with respect to other roads of the signal. Comparison of the density of traffic in the current road with the traffic density in other roads, will provide the necessary decision making and logical thinking capabilities for our algorithm to vary the calculated green light time. The time allotted will be increased if the current road's density is comparably more than the other roads at the signal, it would also be decreased if the current road's density is comparably less than other roads at the traffic junction, and if all the roads have comparably equal density, and then the calculated green light time will not be varied. Here, it is to be noted that by altering the green light time for the road with huge traffic, the waiting time on less traffic road might be increased. Hence, a third optimization is required to obliterate this issue.

After varying the green light time as per traffic density, in the third step a check is done with regards to the mean waiting time of a vehicle at the traffic signal, if the waiting time exceeds the default mean waiting time then the calculated green light time is reduced to match the maximum limit of the mean waiting time.

With this three-step approach of calculating the green light time and optimizing them to instant traffic conditions, the traffic congestion can be prevented and/or controlled effectively.

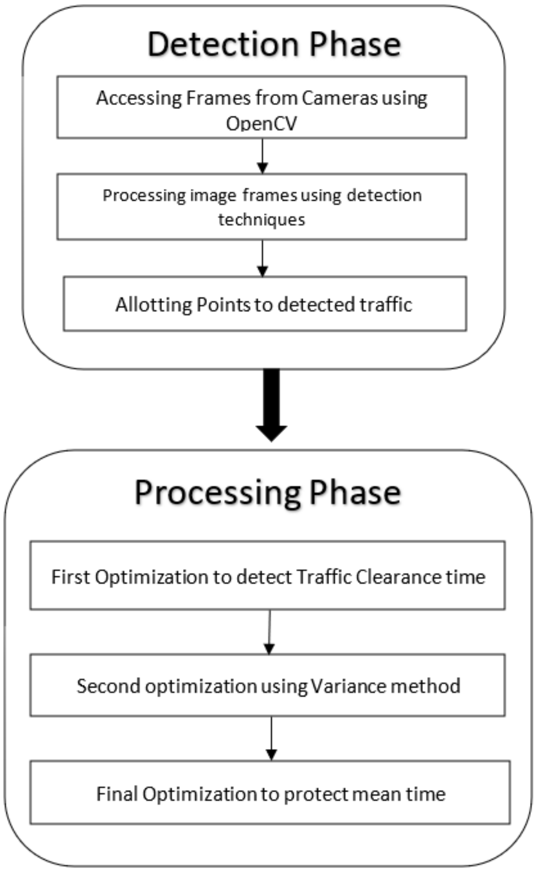


Fig -1: Flow Diagram of Main Process

2. DETECTION

Object detection is a critical capability of autonomous vehicle technology. It's an area of computer vision that's exploding and working much better than just a few years ago. Traffic surveillance cameras are effective in current traffic systems for observation and controlling, so we decided to use the available cameras adjusting at desirable heights making the cost of project quite less. Our algorithm execution starts with detecting vehicles from surveillance cameras and allocating respective points to them.

The detection starts from accessing surveillance camera footage and importing frames using OpenCV. OpenCV is a real time library focusing on computer vision. This python library is completely open source. When the frames are inputted, the next step is to process them. A high-level application programming interface called Cvlib is used to perform image processing and detection. Cvlib is trained under YOLO algorithm on COCO dataset. YOLO stands for "you only look once" is a clever convolutional neural network (CNN) for doing object detection in real time. YOLO applies single CNN over an image and simultaneously predicts multiple bounding boxes and class probabilities for those boxes [4]. The recent version of YOLO, YOLOv3 performs extraction consisting of 53 convolutional layers using binary-entropy loss for the class predictions during training makes it run significantly faster than other detection methods with comparable performance [5]. The training set comes from a large-scale object detection, segmentation and captioning data set called COCO which stands for "Common Objects in Context". COCO was an initiative to collect natural images, the images that reflect everyday scene and provides contextual information [6].

The work flow of object detection will be usually carried out in 3 main steps:

1. Object Localisation: A deep learning model or algorithm (YOLO in our case) is used to generate a large set of bounding boxes spanning full image. The size of the spanning box will depend on the CNN applied. Since Cvlib uses YOLOv3, a 53-layer CNN will be applied for Localisation.

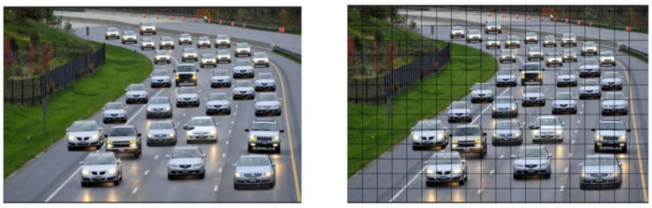


Fig -2: Applying bounding boxes to input image

2. Object Classification: Visual features are extracted for each of the bounding boxes. They are evaluated and it is determined whether and which objects are present in the boxes based on visual features. This is determined using edge detection techniques. In YOLO architecture, a midpoint for all of the determined objects is located and only the box containing the midpoints are focused.

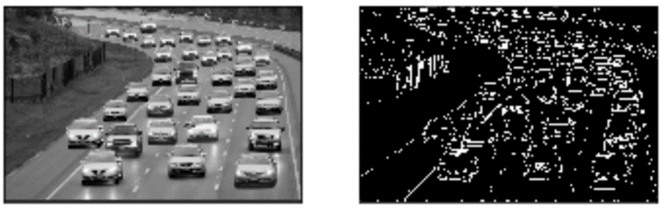


Fig - 3: Applying Edge detection to input image

3. The final step removes all the overlapped boxes using Non maximum suppression. Bounding boxes with less probability of "objectness" are removed hence called Non maximum suppression meaning supressing less maximums.

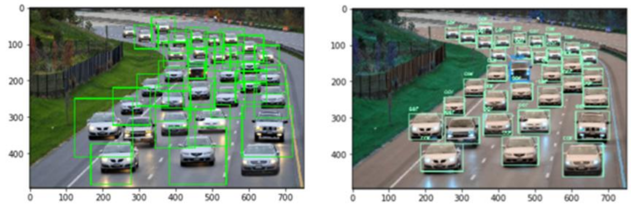


Fig -4: Applying Non-Maximum Suppression

After the vehicles being detected and classified, the points are allotted to each classified stream. The vehicles are divided under several groups as far as accuracy is concerned since not all the vehicles occupy same space and not all of them passes the signal in same time. So, keeping this in mind, each group of vehicles gets some amount of points. The grading system starts with motorcycles since they require least of both time and space. The points are then increased to cars followed by buses and trucks. Both buses and trucks get same amount of points.

3. PROCESSING PHASE

After the detection of vehicles is completed, next task is to process the information gathered in detection phase. The output of detection will be number of vehicles identified with their respective points being allotted.

3.1 Pre-Processing Phase

Before the main execution begins, some of pre-processing activities should be performed so that main processing will be carried out without waiting for certain information. These activities are achieved even before detection but they are used before main execution so they are listed as pre-processing activities.

Initializing:

Even though the algorithm doesn't require explicit manual intervention, some factors need to be initialized before. The initialization begins with number of lanes. The lanes in each road are manually coded to lane variables. These lanes are further used to calculate the lane points. The second thing to initialize is default green light. As stated earlier, we are trying to work with available resources. The current green light time in many areas are effective but not efficient in real time situations. So, the effective green light time is considered and made efficient for real time situations. Each road has different default green light times, so there will be different green light variables.

Point based system:

This approach is a new experiment and it worked out really well. The point-based system allocates unique range of points to specific phases of processing. There are two types of point-based system, one for length of traffic and other for variance method. The points were calculated and listed by performing many trails and errors, referring huge resources and carrying out many numbers of back tracking steps. The points are tested on real time data to achieve exact result and the outcomes were promising.

Creating Classes and Objects:
Object oriented approach is applied by creating a Superior class called road and every road will be its object. Every object carries certain information from its parent class such as default time, number of lanes which are initialized in starting steps. The results of detection are also stored in objects which makes it easier for further calculations and makes memory management and execution speed a bit better. Every time a calculation is performed involving these objects, the recent results will be updated to parent class to help future operations.

3.2 First Optimization Point

The first optimization is to find the length of the traffic and to predict the time required for the traffic to pass the signal. The length of the traffic gives a picture of severity of the traffic. The first optimization point is obtained in two steps, first by finding the length factor followed by calculating green light factor and finally multiplying this green light factor to default green light.

The length factor defines the spread of the traffic. It tells how long the traffic is, based on calculated vehicle count. If the traffic is considered as a matrix, length factor tells about the number of columns in the matrix. The total count of vehicles divided by the number of lanes gives the length factor. It is further used to calculate green light factor.

$$length\ factor \ = \frac{Vehicle\ count}{Number\ of\ Lanes}$$

Green light factor is basically a fractional number which is multiplied to default green light time. It either increases or decreases the default green light in an effective way based on the length factor. Green light factor is obtained by dividing the length factor by lane point associated with respective lane.

$$Green\ light\ factor(GLF) = \frac{length\ factor}{Lane\ point}$$

This obtained green light factor is directly multiplied to default green light time and the result obtained is first optimization point.

$$First\ optimization\ point \ = \ Def.Green\ Light * GLF$$

The green light factor explains up to what extent the default time should be varied so that detected traffic can pass completely. But this is not enough since if the detected traffic on some road is high and on other is low, then vehicles on low traffic road should wait for a long time until huge traffic road gets cleared. So, the obtained optimization point is further tuned in following phases.

3.3 Second Optimization Point

As the first optimization point only tells about varying default time considering traffic on single road making it not much efficient, so the second optimization point comes into picture. This method is based on comparing vehicle densities on all other roads with the current road by finding variance between them. The main goal is to find another tuning factor called variance factor which is multiplied with first optimization and the product is added to first optimization point to increase the accuracy for real time situations.

The phase starts by developing a comparative list which is just a list or array containing vehicle counts of other roads. This list gets updated and changed for every iteration as the source or current road will be changed. After updating the list, the comparison phase begins. Vehicle count of current road which is getting green light in near future is compared with list elements one by one in iterative manner.

The variance factor is initialized to 0 in beginning of the loop and as the iteration proceeds, the variance factor gets updated. The comparison begins between the current vehicle count and first element of list. If the vehicle density in the current road is more than the element in the list, then the variance factor is incremented by variance point else the variance factor is decremented by variance point. A current set consists of current road count and one element from comparative list. The variance of current set is calculated using standard variance function imported from statistics module of python. This variance is used to find the respective variance point by which the variance factor gets modified. This loop continues with all the roads in the comparative list and final variance factor is generated which is modified several times with each iteration.

$$variance = statistics.variance(current\ set)$$

$$if(current\_count > comparitive\_count1):$$

$$variance\ factor = variance\ factor + varaince\ point$$

$$else:$$

$$variance\ factor = variance\ factor - variance\ point$$

The variance factor is multiplied to first optimization point and the product is again added to first optimization point. This couple of arithmetic process confirms that first optimizations is improved itself by some external factor.

$$Second\ Optimization\ point \ = \ first\ opt.point + (firsr\ opt.point * variance\ factor)$$

3.4 Final Optimization Point

So far in all the proposed methods we observed, the road with huge traffic gets its time to clear and road with minimal traffic should wait until the other becomes empty. The problem associated with this is not providing a big time for large traffic but making the traffic on another road wait so long. So, we arrived at a concept called mean time and this phase extremely focuses on maintaining associated mean time.

The mean time is the maximum waiting time a vehicle in first row of the road should wait for. No matter what optimization time you provide, this mean time policy should not be violated. In order to proceed, we calculated mean time based on default green light time as people are used to these timings. For some road A, the mean time is sum of default green light times of other roads. Before announcing the green light time for a particular road i.e., after second optimization, the time calculated is added to recent final optimization times of other roads and checked if mean time is not exceeded. If the condition satisfies, then the calculated time is declared as final optimization time else, the time is altered by finding maximum time less than calculated time which do not exceed the mean time is declared as final optimization time.

The process starts by developing a comparative list consisting of recent final optimization times of other roads. The comparative list gets updated every time as the source road changes. The sum of the list is calculated and added to second optimization time of current road. If the total sum stays within the meantime, the second optimization time is declared as final time with no change. But if it exceeds the meantime then certain tuning is performed to make it stay within the mean time.

$$sum = sum(comparitive\_list)$$

$$if\ (Second\ opt.point + sum) > meanTime:$$

$$Final\ opt.time = meanTime - sum$$

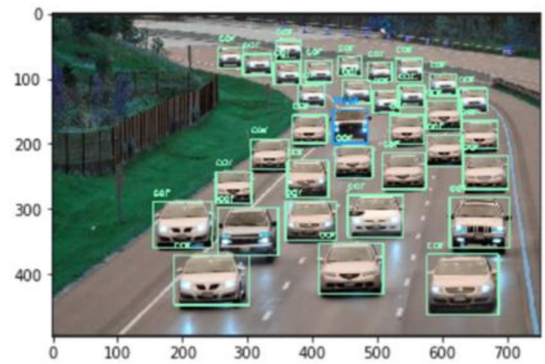
$$else:$$

$$Final\ opt.time = Second\ opt.time$$

4. RESULTS AND OUTCOMES

Experimental results are carried out considering all the aspects of real-time conditions. The algorithm is tested over several kinds of data sets. Every time the testing is performed, the execution speed and deviation from the current architecture are observed and noted.

```
Signal A is opening
```



```
Number of cars identified: 33
Number of bikes identified: 0
Number of buses identified: 0
Number of trucks identified: 1
Total Vehicles detected 34
Total Points Allocated 34.5
Number of Vehicles at B currently: 50
Number of Vehicles at C currently: 15
Number of Vehicles at D currently: 17
First optimization time: 17 seconds
Second Optimization time: 19seconds
Final Optimization time:19 seconds
```

Fig -5: Output for Signal A

Every signal gets its output data in the above format. The program was successful in detecting and classifying vehicles. As planned earlier, different points are allocated to different set of vehicles. In the above figure it is observed that, current vehicle count of all other signals are listed and compared followed by modifying the optimization times.

When the execution of all the signals gets completed, the cycle summary for respective cycle will be obtained. This cycle summary gives us the brief outline of how better or worse our algorithm in effect to traditional one.

```
Total default time for 1 cycle to complete: 100
Total Number of vehicles which can pass through default time: 81
Total optimized time for 1 cycle to complete: 49
Total Number of vehicles which can pass through optimized time: 79
```

Signals:	Signal A	Signal B	Signal C	Signal D
Vehicles Identified:	34	9	20	17
Default Time:	25	25	25	25
Optimized Time:	23	4	13	9

Fig -6: Cycle summary for one particular Cycle

In the above figure we can observe the variation of green light times allotted by both default and optimized methods. The point-based system is effective in predicting the vehicles that can be passed within allotted time. It is shown that even in less time, the algorithm is successful in passing a greater number of vehicles.

Optimized method vs default method under different circumstances:

Excessive traffic congestion on few roads:

Signals:	Signal A	Signal B	Signal C	Signal D
Vehicles Identified:	34	45	15	37
Default Time:	25	25	25	25
Optimized Time:	19	30	7	23

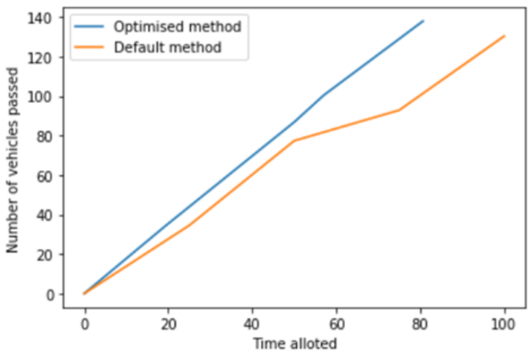


Fig -7: Graph depicting results of congestion on few roads

In the above figure it is proved that during uneven traffic congestion, the optimized method is much better than default method both in total time and vehicles passed.

Excessive traffic congestion on all roads:

Signals:	Signal A	Signal B	Signal C	Signal D
Vehicles Identified:	45	42	45	38
Default Time:	25	25	25	25
Optimized Time:	25	25	25	25

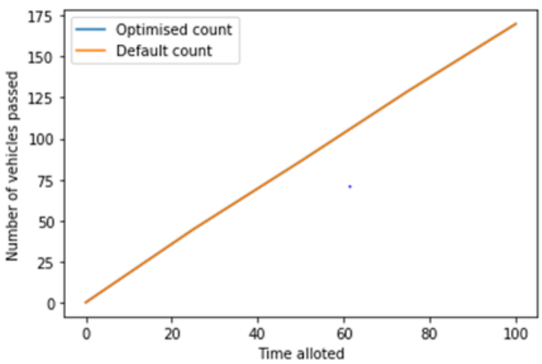


Fig -8: Graph depicting results of congestion on all roads

It is observed that when the traffic is evenly congested, both the methods perform equally same. Since all the signal gets the same amount of green light time, so there will not be a major difference in both optimization and default methods.

Mixture of high and zero vehicle densities:

Signals:	Signal A	Signal B	Signal C	Signal D
Vehicles Identified:	45	0	20	35
Default Time:	25	25	25	25
Optimized Time:	25	0	19	21

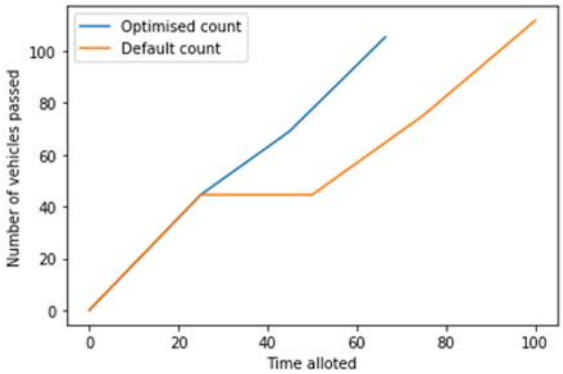


Fig -9: Graph depicting results of mixture of high and low traffic

When there is a difference in traffic density among the roads, again the optimized technique gains the maximum efficiency.

From all the simulations carried, it is noted that the optimization method wins over default method by huge margin when the traffic density is uneven. When the traffic is equal, both perform considerably same.

5. CONCLUSION

The problems faced by the country and its people due to poor traffic management system led to the motivation of building a new traffic management architecture, which aims to prevent and manage the traffic congestion in a better way using the present advanced technologies. In this project we aimed to design and implement an effective architecture than the existing ones using innovative approaches. Collection of the traffic dataset was an important obstacle, but with the available resources, the new system of traffic management is proposed to the community, which is designed, implemented and tested successfully. Also, the implementation is cost effective and overcomes the present system's weakness of preventing and managing the traffic congestion. We were successful in converting the time wasted in current architecture to a useful amount which can increase traffic clearance faster. This implementation will be efficient than the existing ones and make the traffic management simple by making people's day to day life easier.

6. IMPROVEMENTS

So far so good, the program worked very well but there is a scope for improvement all the time. We worked on this project mainly concentrating on time wasted in current or proposed traffic methodologies. But that's not enough, the experimented method should withhold daily changing challenges. The developed optimization techniques are very new so we were adopted for establishing a working content rather than making them absolute best. Our optimization techniques can be further enhanced to compete with big real time data. Lack of specific data sets and constraints on speed and time as execution of program is concerned, limited us to arrive to this point-based method. If the resources are made available, the optimizations can be made advanced by training over a huge dataset which can accommodate the success.

ACKNOWLEDGEMENT

We thank our team members R Pradeep, Rahul V and Rasineni Yatish for their assistance throughout all aspects of our study and we additionally thank our Mentor Harikrishna S M (Assistant Professor, Department of Computer Science, Faculty of Engineering and Technology, Ramaiah University of Applied Sciences) for his continuous support.

REFERENCES

[1]: Cao, Zhiguang & Jiang, Siwei & Zhang, Jie & Guo, Hongliang. (2017). A Unified Framework for Vehicle Rerouting and Traffic Light Control to Reduce Traffic Congestion. IEEE Transactions on Intelligent Transportation Systems. 18. 1958 - 1973. 10.1109/TITS.2016.2613997.

[2]: Saranya J, Jayashwanth J.S, Kiran J, Harish S, Linga Kumar T. (2019). Automated Density Based Traffic Light Control System Using Arduino Platform.

[3]: Jain, Vipin & Sharma, Ashlesh & Subramanian, Lakshminarayanan. (2012). Road traffic congestion in the developing world. 10.1145/2160601.2160616.

[4]: J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 779-788, doi: 10.1109/CVPR.2016.91.

[5]: Redmon, Joseph & Farhadi, Ali. (2018). YOLOv3: An Incremental Improvement.

[6]: Lin TY. et al. (2014) Microsoft COCO: Common Objects in Context. In: Fleet D., Pajdla T., Schiele B., Tuytelaars T. (eds) Computer Vision – ECCV 2014. ECCV 2014. Lecture Notes in Computer Science, vol 8693. Springer, Cham. https://doi.org/10.1007/978-3-319-10602-1_48