Computing (2022) 104:1203–1217
https://doi.org/10.1007/s00607-021-01038-1

REGULAR PAPER

Machine learning driven intelligent and self adaptive system for traffic management in smart cities

Hameed Khan[1] · Kamal K. Kushwah[1] · Muni Raj Maurya[2,3] · Saurabh Singh[1] · Prashant Jha[1] · Sujeet K. Mahobia[1] · Sanjay Soni[1] · Subham Sahu[1] · Kishor Kumar Sadasivuni[2]

Received: 13 July 2021 / Accepted: 25 November 2021 / Published online: 16 January 2022

Abstract
Traffic congestion is becoming a serious problem with the large number of vehicle on the roads. In the traditional traffic control system, the timing of the green light is adjusted regardless of the average traffic rate at the junction. Many strategies have been introduced to solve and improve vehicle management. However, in order to handle road traffic issues, an intelligent traffic management solution is required. This article represents a self adaptive real-time traffic light control algorithm based on the traffic flow. We present a machine learning approach coupled with image processing to manage the traffic clearance at the signal junction. The proposed system utilizes single image processing via neural network and You Only Look Once (YOLOv3) framework to establish traffic clearance at the signal. We employed YOLO architectures because it is accurate in terms of mean average precision (mAP), interaction over union (IOU) values and fast in object detection tasks as well. It runs significantly faster than other detection methods with comparable performance. The average processing time of single image was estimated to be 1.3 s. Further based on the input from YOLO we estimated the 'on' time period green light for effective traffic clearance. Several real time parameters like number of vehicles (two wheelers, four wheelers), road width and junction crossing time are considered to estimate the 'on'time of green light. Moreover, we used the real traffic images to test the performance and trained the system with different dataset. Our experiments investigation reveals that the predicted

✉ Kamal K. Kushwah
kamal_kushwah2005@yahoo.com

✉ Saurabh Singh
ssingh@jecjabalpur.ac.in

✉ Kishor Kumar Sadasivuni
kishorkumars@qu.edu.qa

1 Jabalpur Engineering College, Jabalpur, Madhya Pradesh, India

2 Center for Advanced Materials, Qatar University, Doha, Qatar

3 Mechanical and Industrial Engineering, Qatar University, Doha, Qatar

vehicle counts were well matched with the actual vehicle count and proposed method apprehended an average accuracy of 81.1%. The reported strategy is self adaptive, highly accurate, fast and has the potential to be implemented in the traffic clearance at the junctions.

Keywords Traffic management · Machine learning YOLO · Image processing · Open CV

Mathematics Subject Classification 68T05 Learning and adaptive system in AI · 68T45 Machine vision and scene understanding · 68-11 Research data for problem pertaining to computer science

1 Introduction

Traffic congestion is a major problem in many cities. Roads are witnessing a large number of vehicles running each day and have made the task of managing the traffic more tedious. Extensive research works are going on to make traffic management system more adaptive, smart, and intelligent. Cameras have been installed on the roads and at junctions for surveillance purposes, for imposing autonomous penalty and person identification upon violating the traffic rules. In terms of traffic management, majority of the traffic junctions use a fixed time green light cyclic system for managing the traffic. The incorporation of fixed cyclic time period operation in traffic light management system imposes certain limitations and has proven to be significantly inefficient in regulating traffic congestions. The traditional time-limited robot system works best when the traffic flow is almost the same in all four directions. However, during a day, we witness a situation when traffic is more from one compared to other directions. Moreover, the traditional system lacks smart management; as a result it makes people wait, irrespective of no vehicle from the other side. This unavoidable waiting time sometime makes person restless, which often ends up in violation of rules and accidents. Further, it promotes more consumption of fuel and adds pollutants in surrounding environment. Consistent research work has been carried out to manage traffic congestion and automate the traffic management process. The idea of making the traffic light adaptive to real-time traffic stream is not new, and different strategies have been reported to achieve it. Researcher proposed the design; which consists of three basic components: a parking management center, a traffic management center, and world facts and management core [1]. The facility depends on the functioning of sensory networks to accumulate traffic congestion and the range of vehicles traveling at every intersection to make a decision when estimating the length of time the traffic light can stay green. Several other framework of the intelligent traffic management has been proposed in this regard [2,3]. These structured frameworks offers the STMS site visitors manipulate machine as the main module and has small modules such as video manipulate system, traffic control system, computer control system, and peripheral devices. The traffic control system controls heavy visitors at a predetermined time on the road. It uses a video surveillance device to discover extra site visitors with a video camera and when the range of automobiles on a precise road exceeds the pre-determined number, it notifies traffic controllers with an alarm indicating that "traffic has been reached" and prevents any other car from entering that route. The following cars would consequently be diverted to every other certain route, which will lead to controlling of traffic jams. This site visitors manipulate machine includes high-quality transmission and uninterrupted conversation by sending and receiving the appropriate signal at the proper time. Peripheral smart devices (IoT) deal with the right configuration of the input sensors and output actuators to capture and receive events and send remarks as needed details for managing points [4]. Similarly, the deployment of the CCTV camera to visualize areas of the smart road covering the scope of the situation is also an important function of this module [5].

The potential to predict the temporary emergence of the contemporary traffic state is an essential basis for persisted use in traffic management and control. Some studies focused on Advanced Traffic Management Systems (ATMS)/Advanced Traveler Information Systems (ATIS) routes, a community of traffic sensors that grant real-time traffic information [6]. Other significant cost effective approach is vehicle counting and tracking with YOLO using real-time video data; the counting method process a traffic video captured using handheld cameras in a video-based vehicle [7]. Based on this a real-time traffic light control algorithm has been reported which takes input as the real-time traffic video and optimizes the green light time accordingly [8]. Very similar methods have been proposed to use video data to make self-adaptive traffic light and replacing a traffic police officer with an intelligent system [9,10]. In Table 1 we provide a comparative analysis of the different types of work performed by many researchers on different platforms. One thing is common in all the work done; all traffic management systems take input as video data and process the video to get insights about the traffic condition.

In the above reported studies, usually traffic stream video is processed as a standard input data to manage the traffic clearance [2,6–10,22]. Video is a collection of images, thus to process a video and get the insights of traffic, all the images need to be processed in real-time, which is a computationally expensive and time consuming. Moreover, the continuous processing of images extracted from the video affects the durability of the system and is expensive for long term usage. Due to these constrains there is a high need of an intelligent traffic management that is fast, accurate and cost effective. In this paper, a solution to above mentioned problems has been attempted with of machine learning tool. The YOLO object detection algorithm is used to detect cars in real-time images captured by a camera mounted on the top of the signal and allocate green light time according to the number of vehicles. Compared to complete video processing the proposed self adaptive utilizes a single image processing to calculate the optimal green light time in view of optimized memory requirements [23]. The proposed model offer self adaptive robust facility to makes traffic management more efficient and effective.

Table 1 Summary of existing relevant studies and applications

Study	Focus and application	Approach
Makaba et al. [11]	Bayesian network-based framework	Simulation framework
Khan et al. [12]	Aerial vehicle-based traffic analysis	Empirical study
Takano et al. [13]	High-resolution image data collection	Numerical simulation
Fielbaum [14]	Impact of traffic congestion	Numerical simulations
Lykov and Asakura [15]	Tensor-based abnormal pattern detection for traffic	Numerical simulation
Contreras and Gamess [16]	Algorithm to count vehicles at a traffic signal	Numerous simulations
Hu et al. [17]	Recognition of traffic density	Traffic Simulator
Peque et al. [18]	Adaptive learning algorithms for dynamic traffic	Mathematical formulation
Hamidi and Kamankesh [19]	Intelligent traffic management system	Multi-agent system
García et al. [20]	Intelligent method for traffic light scheduling	IOCA-PSO method
Jain [21]	Automatic traffic signal controller	Membership function based

2 Proposed methodology

2.1 System architecture

The system requires four cameras, each at the top of the road that meets on a junction (see Fig. 1). These cameras are used to send the image to an embedded controller. After that, when the controller receives the real-time image, it will count the number of vehicles present in the image and light up the green signal according to the count of vehicles. Most of the traffic junctions have such kind of cameras already installed on top of the signals. We need to connect them with a controller. The controller would be able to perform the operations listed below:

1. Instruct the camera to click the picture of the traffic situation.
2. Process the input image from the camera and count the number of vehicles in the image.
3. Calculate the time duration, as to how long the green light should remain in on state based on the vehicle count.
4. Light up the green light of the corresponding signal and the red light of the other signals.

As shown in Fig. 2, cameras are covering only one side of the road. This site is the driving side from where the vehicles are arriving on the signal. The cameras have to be installed such that vehicles should not overlap each other in the image; otherwise, it can lead us to false detection results. Figure 3 shows the different type of images taken

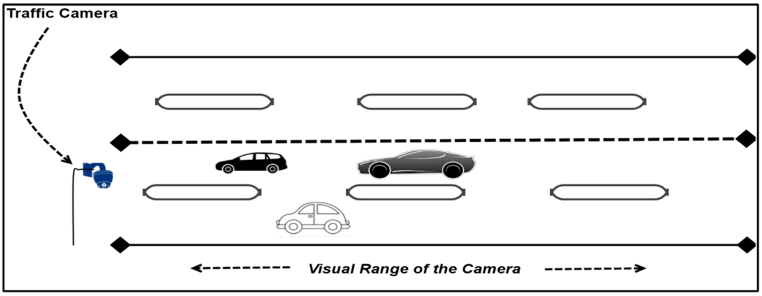


Fig. 1 The side view of the camera placement

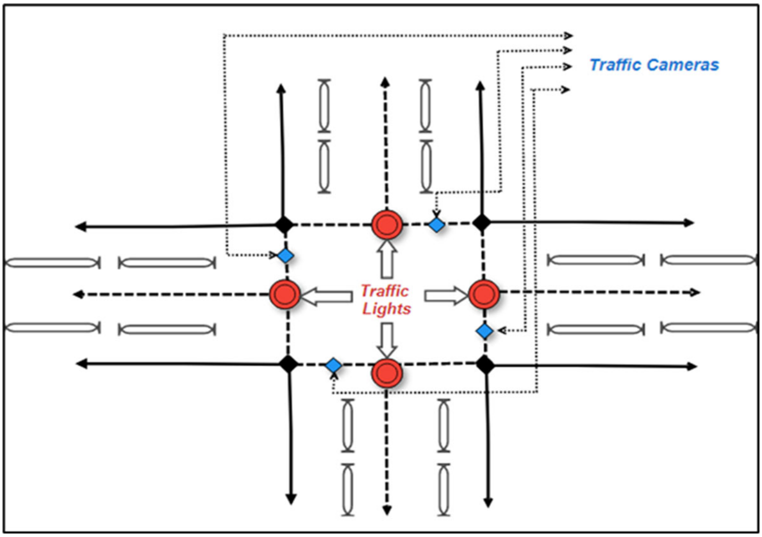


Fig. 2 Top view of the camera placement

on the traffic signal. From figure it's clear that images on the left side are not suitable for processing, as the percentage of vehicle overlapping is significantly high. On the other hand, the two images on the right portion of Fig. 3, gives a clear distribution of the vehicle with minimum overlapping. Thus, these types of images will be very helpful for the present proposed method.

2.2 System architecture

In this work, OpenCV is used for counting the vehicles from the input image. OpenCV is an image processing library written in C++ and Python. OpenCV [5] provides modules to run object detection with various methods, and here in this work, OpenCV.cdn [24] module is used to count the number of vehicles. The input image is provided to the YOLO v3 model through OpenCV [25,26]. YOLO is a smart convolution neural network (CNN) for object detection in real-time. The algorithm applies a single neu-

Fig. 3 The sample images for training purposes

ral network to the full image, and then divides the image into regions and predicts bounding boxes and probabilities for each region. These bounding boxes are weighted by the predicted probabilities. It returns a list of bounding box coordinates for each object detected in the image. The number of vehicles would be the number of objects in the returned list. Majority of the study consider equal time lane clearance time for two-wheelers (TW) and four-wheelers (FW) vehicles (see Fig. 4) [27,28]. However, considering the real time scenario the FW take more time than TW vehicles in starting and leaving the signal. Thus, it's important to consider the time taken to leave the signal junction also. In this work, we have considered that TW vehicles and FW vehicles contribute differently to the time required to clear the lane. Based on the number of vehicles and their respective time to clear the lane the the actual time period is estimated and finally the controller allocate the on time green light which is represented as 't'. The time 't' is determined by Eq. (1).

$$t = \lceil TW/a \rceil \times n + \lceil FW/b \rceil \times m \tag{1}$$

where TW = Number of two-wheelers/bikes in the image; FW = Number of four-wheelers/cars in the image; n = Time to clear single two-wheeler; m = Time to clear single four-wheeler; a = maximum no. of two-wheeler covers the whole width of the road; b = maximum no. of four-wheeler covers the whole width of the road; $\lceil x \rceil$ Denotes ceil of x.

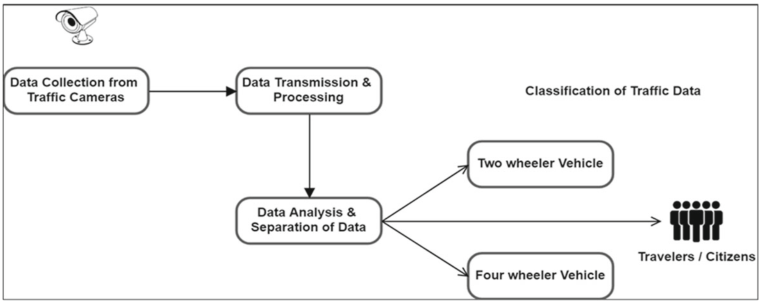


Fig. 4 TCS stages and flow of communication

Clearance time of the individual vehicle is considered by considering the time taken by the vehicle to cross the stop line, when it is standing just before the stop line. Thus constants n, m, are determined by the real-time observation of the traffic junction. The measured value for n and m are 3.6 and 6 s respectively. Since the reaction time of a driver in situations is generally greater than 1 s, thus we have rounded the value of n to 4 s. To calculate constants a, b, we need to consider more parameters like, road width (Wroad), width of a TW (W2), width of a FW (W4), and the gap between any two vehicles (G). Road width may vary for different traffic junctions. Here we are considering the common case. Wroad $= 30 \pm 5$ feet, W2 $= 2$ feet, W4 $= 6$ feet, G $= 1$ feet, then a and b are calculated using Eqs. (2) and (3), respectively.

$$a = \left\lfloor \frac{W_{road}}{W_2 + G} \right\rfloor \tag{2}$$

$$b = \left\lfloor \frac{W_{road}}{W_4 + G} \right\rfloor \tag{3}$$

For, road width = 25 ft.

$$\mathrm{a} = \left\lfloor \frac{25}{2+1} \right\rfloor = \lfloor 8.33 \rfloor = 8;\ \mathrm{b} = \left\lfloor \frac{25}{6+1} \right\rfloor = \lfloor 3.57 \rfloor = 3$$

For, road width = 30 ft.

$$\mathrm{a} = \left\lfloor \frac{30}{2+1} \right\rfloor = \lfloor 10 \rfloor = 10;\ \mathrm{b} = \left\lfloor \frac{30}{6+1} \right\rfloor = \lfloor 4.28 \rfloor = 4$$

For, road width = 35 ft.

$$\mathrm{a} = \left\lfloor \frac{35}{2+1} \right\rfloor = \lfloor 11.67 \rfloor = 11;\ \mathrm{b} = \left\lfloor \frac{35}{6+1} \right\rfloor = \lfloor 5 \rfloor = 5$$

where $\lfloor x \rfloor$ denotes floor of x

There is a huge possibility that there are too many vehicles waiting at the junction and will take long time for them to pass, which will lead to starvation for vehicles on the another side of road junction. Thus to overcome this need to set a maximum threshold time and choose the minimum value between the time estimated from the equations mentioned above. Considering this situation the actual time to calculate the on time for green light is given by Eq. 4

$$t_{actual} = min(t, MAX\_GREEN\_TIME) \tag{4}$$

To calculate MAX_GREEN_TIME, we need to consider the worst case of Eq. (1) that is when all vehicles are FW. Assuming that on any side of the junction we need to clear maximum 20 FW and putting TW = 0, FW = 20 in Eq. (1) and considering different road widths the estimated time is

For road width = 25 ft.

$$\mathrm{t} = \left\lceil \frac{0}{8} \right\rceil \times 4 + \left\lceil \frac{20}{3} \right\rceil \times 6 = 42\ \mathrm{s}$$

For road width = 30 ft.

$$\mathrm{t} = \left\lceil \frac{0}{10} \right\rceil \times 4 + \left\lceil \frac{20}{4} \right\rceil \times 6 = 30\ \mathrm{s}$$

For road width = 35 ft.

$$\mathrm{t} = \left\lceil \frac{0}{11} \right\rceil \times 4 + \left\lceil \frac{20}{5} \right\rceil \times 6 = 24\ \mathrm{s}$$

For further calculation we have considered the worst-case scenario i.e. the minimum road width of 25 ft. that correspond to the MAX_GREEN_TIME of 42 s. Figure 5 shows the behavior of Eq. (1), with respect to number of vehicles, for three different scenarios; (1) when traffic only has TW, (2) when traffic has only FW, and (3) when traffic has both TW and FW in equal ratio. From Fig. 5 it is observed that irrespective of the above three situations the curve reaches a flat for MAX_GREEN_TIME of 42 s.

To further investigate on the worst-case scenario we have only considered the FW vehicles and estimated the clearance time, as shown in Fig. 6. From the figure it can be observed that the calculated total clearance time of 100 FW vehicles in traffic is 546 s. The round time mentioned in the plot is the time when a vehicle get second clearance chance after MAX_GREEN_TIME threshold time is reached. Figure 7 represent the flow chart of the algorithm for a four way junctions clearance framework. For each of the four ways meeting on the intersection:

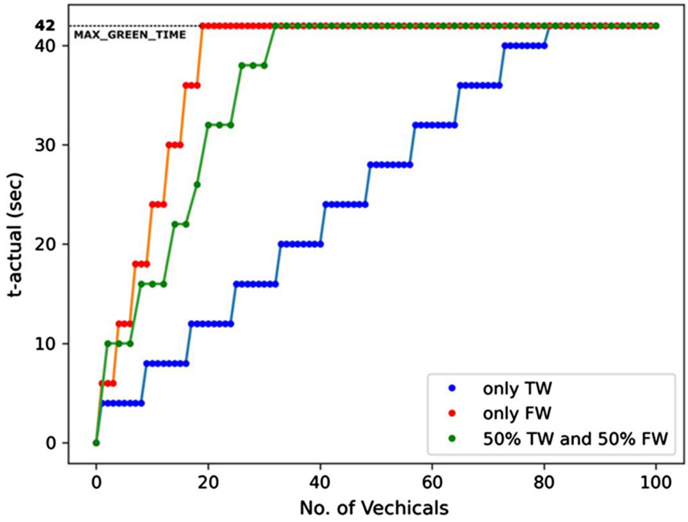


Fig. 5 Behavior of formula derived to calculate time for green light

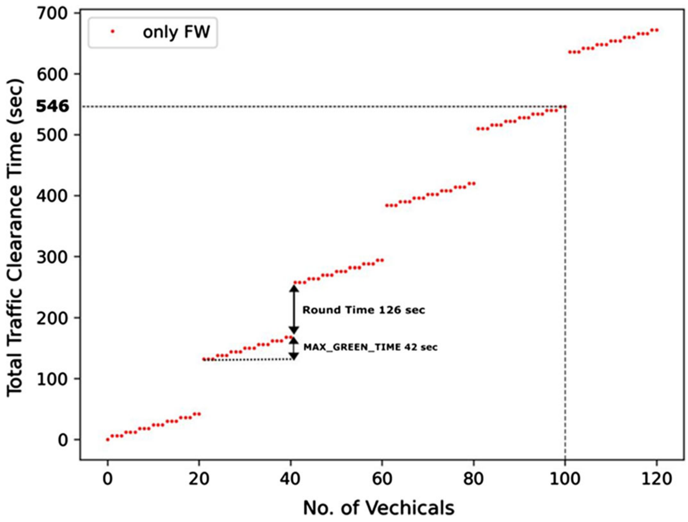
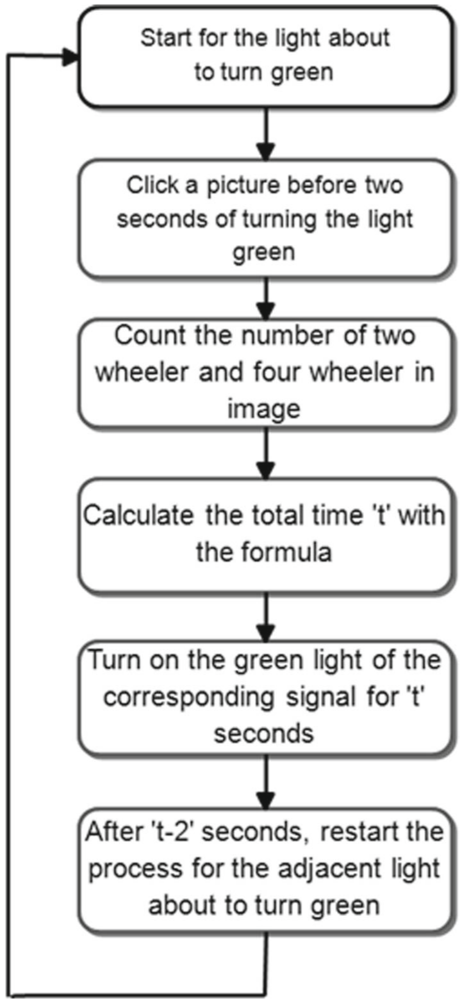


Fig. 6 Total clearance time for traffic with respect to number of vehicles in worst case

1. Take input as the image before 2 s of turning ON the green light.
2. Preprocess the image for vehicle detection.
3. Perform object detection and count the number of (a) TW (b) and FW in the image.
4. Calculate the total time t by putting values in Eq. (1).
5. Turn the green light for the total duration of time t
6. Repeat for the adjacent camera from step-1.

The extraction of features, convolution layers are there in the model to extract them. The data input is actually an image which is provided in the form of a numpy array

Fig. 7 Flowchart of the process



to the model and the model outputs the objects detected in the image with the coordinates of the corresponding object. YOLOv3 is the state-of-the-art model for real-time object detection. YOLOv3 predicts 4 coordinates for each bounding box around an object. Training is performed with sum of squared error loss. YOLOv3 predicts an objectness score for each bounding box using logistic regression and the class probabilities using independent logistic classifiers. It uses binary cross-entropy loss for the class predictions. YOLOv3 extracts features from 3 different scales using a concept similar to feature pyramid networks. From the darknet-53 feature extraction backbone, YOLOv3 adds several convolutional layers, the last of which predicts a 3-d tensor encoding bounding box, objectness, and class predictions. In YOLOv3's original COCO experiments, they predict 3 bounding boxes at each scale, so the tensor is N N $[3(4 + 1 + 80)]$ for the 4 bounding box offsets, 1 objectness prediction, and 80 class predictions. For the second scale, YOLOv3 takes the feature map from the 2

Table 2 Vehicle detection results on testing images

Image no.	Actual count	Predicted count	Correct predictions (true positive)	False positives (%)	Accuracy (%)	Time (s)
5	42	39	38	1	92.85	1.76717
10	18	12	11	1	66.66	1.25578
15	13	12	11	1	92.3	1.26183
31	32	32	28	4	100	1.2677
45	91	49	46	3	53.84	1.25728

previous layers and up samples it by 2×. It also concatenates in a feature map from earlier in the network, and then adds a few more convolutional layers to process the combined feature map, and now predict a similar tensor at the second scale. This process is repeated at the third scale. It automatically identifies and draws rectangular bounding boxes around objects of interest at a rate of roughly 15–20+ frames per second. YOLO networks divide the image into regions and predict bounding boxes and probabilities for each region. For a 416×416 input image, YOLOv3 predicts $13 \times 13 \times 3 = 507$ boxes for the first scale, $26 \times 26 \times 3 = 2028$ boxes for the second scale, and $52 \times 52 \times 3 = 8112$ boxes for the third scale, for a total of 10,647 boxes. Non-maxima suppression and IOU thresholds are then used to cut the number of boxes down significantly, often to a few or a couple dozen per image. There are several alternative state-of-the-art object detection models and some of these models are more accurate than YOLOv3 or return more specific results. However, for all of these alternative models, the increased accuracy comes at the cost of slower. In this paper our objective is to detect objects in real time, hence we have to set a trade-off between speed and accuracy.

3 Results and discussion

The YOLO object detection model was trained on 42 authentic traffic images of two main junctions of the city Jabalpur. A total of five images were kept apart from the training set for testing the results. Table 2 illustrates the results obtained on the test images.

From the results in Table 2, it can be concluded that YOLO gives us fair results considering that we have a minimal dataset that consists only of 42 images. The mean time taken for detecting the vehicles per image is 1.36 s. The time taken in object detection mainly depends on the incorporated processing hardware. The above results have been obtained on Intel(R) Xeon(R) CPU @ 2.00GHz processor. In Table 2, the results obtained for image no. 45 shows quite different results from the other four images. It is an outlier. That image was not taken at a right angle, and it has many vehicles overlapping each other. For correct predictions, it is necessary to take an image that shows vehicles as discrete as possible, not overlapped by any other object, as discussed in the system architecture section and shown in Fig. 3 (right portion

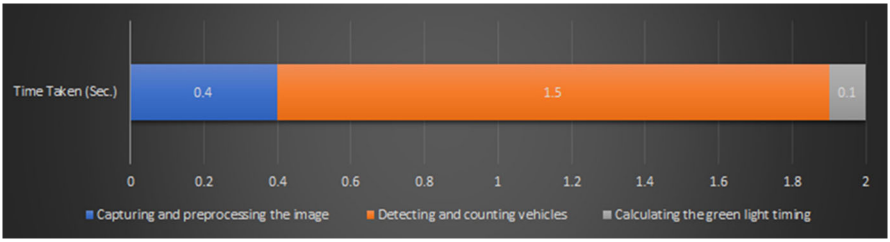
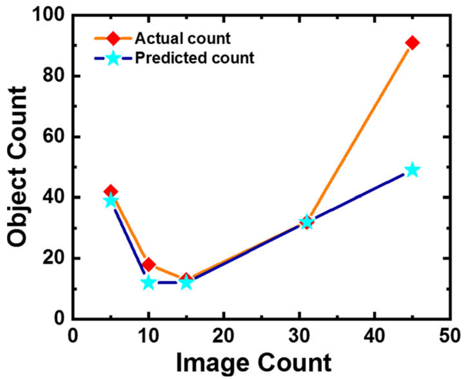


Fig. 8 Processing time for one image

Fig. 9 Actual object count versus predicted object count



images). The time mentioned in Table 2 is calculated programmatically by taking the difference between the time points just after giving the input image and the time point just after getting output. However the time considered for illuminating the green light was estimated to be 2 s. Figure 8 summarizes the reason behind taking 2 s before turning the green lights ON. From the results in Table 2, the meantime for vehicle detection and counting is 1.36 s, which can go up to 1.5 s. In some cases, about 0.4 s has taken in image preprocessing before the vehicle detection process. In the end, the green light time calculation can take a maximum time of 0.1 s. The whole process sums up in 1 s. Figure 9 represents the actual object count versus the predicted count. From the figure it is evident that, the dark blue line curve (predicted) follows the orange cure (actual) except for image number 45, reason behind such behavior has already been described earlier. In Fig. 10, the graph summarizes the accuracy of images based on the object predicted in the image. As it is visible, image 31 has highest accuracy while image 45 has the lowest accuracy. Thus the average estimated accuracy of the adopted method was 81.1%.

The processing time of five images has been depicted in Fig. 11. Here minimum processing time is 1.25 s, whereas the maximum processing time is 1.76 s. The mean processing time is approximately 1.3 s. From the above investigations it can be inferred that the proposed intelligent traffic management system based on the single image processing is self adaptive, highly accurate, fast and has the potential to be implemented in the traffic clearance at the junctions.

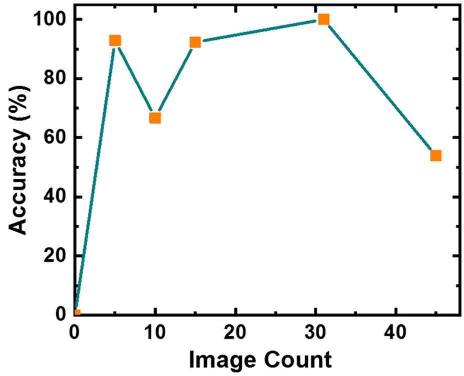


Fig. 10 Accuracy of images

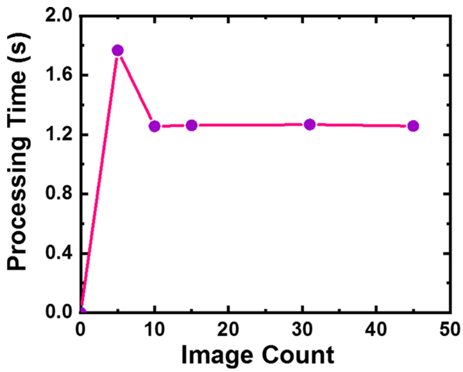


Fig. 11 Image processing time

4 Conclusion and future work

In summary, a traffic management system has been proposed to make the traffic lights adaptive to real-time traffic streams with the help of machine learning driven YOLO method. We considered the time parameters like number of vehicles (two wheelers, four wheelers), road width and juction crossing time to estimate the 'on' time of green light. We incorporated the real traffic images as input to our neural-network and trained the system with different dataset that improved the vehicle detection. The evaluation results showed that the proposed system achieved satisfactory performance with an average accuracy of 81.1%. The reported method is modest in hardware requirements and cost effective compared to the traditional traffic clearance strategies. In addition, it does not need large scale construction or installation work. The proposed model facilitates that the system can be improved by making it learn by itself with the help of reinforcement learning. This can be active by providing regular feedback to the system so that it can learn from it. In this way, the system will become more accurate with time. The time taken for capturing the image and calculating the time can also be reduced from 2 s with the help of a faster object detection system. Functionality to take care of special cases can be added in the future. For example, if a traffic camera detects an ambulance, the system will let it pass as soon as possible. Moreover the future work will consider the test with cameras installed at higher altitude having wide and panoramic coverage to cover larger section of the road that will be further improve the accuracy of the results.

References

1. Redmon J, Farhadi A (2018) Yolov3: an incremental improvement. arXiv preprint arXiv:1804.02767
2. Rath M (2018) Smart traffic management system for traffic control using automated mechanical and electronic devices. IOP Conf Ser Mater Sci Eng 377:012201. https://doi.org/10.1088/1757-899X/377/1/012201
3. Shinde MS, Jagtap S (2016) Intelligent traffic management systems. A review. Int J Innov Res Sci Technol 2:293–298
4. Komsiyah S, Desvania E (2021) Traffic lights analysis and simulation using fuzzy inference system of mamdani on three-signaled intersections. Procedia Comput Sci 179:268–280. https://doi.org/10.1016/j.procs.2021.01.006
5. Toh CK, Sanguesa JA, Cano JC, Martinez FJ (2020) Advances in smart roads for future smart cities. Proc R Soc A 476(2233):20190439. https://doi.org/10.1098/rspa.2019.0439
6. Allström A, Barceló J, Ekström J, Grumert E, Gundlegård D, Rydergren C (2017) Traffic management for smart cities. In: Angelakis V, Tragos E, Pöhls HC, Kapovits A, Bassi A (eds) Designing, developing, and facilitating smart cities. Springer, Cham, pp 211–240. https://doi.org/10.1007/978-3-319-44924-1_11
7. Asha CS, Narasimhadhan AV (2018) Vehicle counting for traffic management system using YOLO and correlation filter. In: 2018 IEEE international conference on electronics, computing and communication technologies (CONECCT). IEEE, Bangalore, pp 1–6. https://doi.org/10.1109/CONECCT.2018.8482380
8. Alpatov BA, Babayan PV, Ershov MD (2018) Vehicle detection and counting system for real-time traffic surveillance. In: 2018 7th Mediterranean conference on embedded computing (MECO). IEEE, Budva, pp 1–4. https://doi.org/10.1109/MECO.2018.8406017
9. Basil E, Sawant S (2017) IoT based traffic light control system using Raspberry Pi. In: 2017 International conference on energy, communication, data analytics and soft computing (ICECDS). IEEE, Chennai, pp 1078–1081. https://doi.org/10.1109/ICECDS.2017.8389604. https://ieeexplore.ieee.org/document/8389604/
10. Corovic A, Ilic V, Duric S, Marijan M, Pavkovic B (2018) The real-time detection of traffic participants using YOLO algorithm. In: 2018 26th Telecommunications forum (TELFOR). IEEE, Belgrade, pp 1–4. https://doi.org/10.1109/TELFOR.2018.8611986
11. Makaba T, Doorsamy W, Paul BS (2021) Bayesian network-based framework for cost-implication assessment of road traffic collisions. Int J ITS Res 19(1):240–253. https://doi.org/10.1007/s13177-020-00242-1
12. Khan M, Ectors W, Bellemans T, Janssens D, Wets G (2018) Unmanned aerial vehicle-based traffic analysis: a case study for shockwave identification and flow parameters estimation at signalized intersections. Remote Sens 10(3):458. https://doi.org/10.3390/rs10030458
13. Takano T, Ono S, Kawasaki H, Ikeuchi K (2021) High-resolution image data collection scheme for road sensing using wide-angle cameras on general-use vehicle criteria to include/exclude collected images for super resolution. Int J ITS Res 19(2):299–311. https://doi.org/10.1007/s13177-020-00243-0
14. Fielbaum A (2020) Strategic public transport design using autonomous vehicles and other new technologies. Int J ITS Res 18(2):183–191. https://doi.org/10.1007/s13177-019-00190-5
15. Lykov S, Asakura Y (2020) Anomalous traffic pattern detection in large urban areas: tensor-based approach with continuum modeling of traffic flow. Int J ITS Res 18(1):13–21. https://doi.org/10.1007/s13177-018-0167-5
16. Contreras M, Gamess E (2020) An algorithm based on VANET technology to count vehicles stopped at a traffic light. Int J ITS Res 18(1):122–139. https://doi.org/10.1007/s13177-019-00184-3
17. Hu H, Gao Z, Sheng Y, Zhang C, Zheng R (2019) Traffic density recognition based on image global texture feature. Int J ITS Res 17(3):171–180. https://doi.org/10.1007/s13177-019-00187-0
18. Peque G, Miyagi T, Kurauchi F (2018) Adaptive learning algorithms for simulation-based dynamic traffic user equilibrium. Int J ITS Res 16(3):215–226. https://doi.org/10.1007/s13177-017-0150-6

19. Hamidi H, Kamankesh A (2018) An approach to intelligent traffic management system using a multi-agent system. Int J ITS Res 16(2):112–124. https://doi.org/10.1007/s13177-017-0142-6
20. García-Nieto J, Alba E, Carolina Olivera A (2012) Swarm intelligence for traffic light scheduling: application to real urban areas. Eng Appl Artif Intell 25(2):274–283. https://doi.org/10.1016/j.engappai.2011.04.011
21. Jain P (2011) Automatic traffic signal controller for roads by exploiting fuzzy logic. In: Das VV, Stephen J, Chaba Y (eds) Computer networks and information technologies, vol 142. Springer. Berlin, Heidelberg, pp 273–277. https://doi.org/10.1007/978-3-642-19542-6_46
22. Hilmani A, Maizate A, Hassouni L (2020) Automated real-time intelligent traffic control system for smart cities using wireless sensor networks. Wirel Commun Mob Comput 2020:1–28. https://doi.org/10.1155/2020/8841893
23. Singh S, Vishwakarma S, Kispotta S, Yadav A (2020) A statistical analysis of various technologies to detect and prevent fake news. In: Batra U, Roy NR, Panda B (eds) Data science and analytics, vol 1230. Springer, Singapore, pp 164–172. https://doi.org/10.1007/978-981-15-5830-6_15
24. Pop MD (2018) Traffic lights management using optimization tool. Procedia Soc Behav Sci 238:323–330. https://doi.org/10.1016/j.sbspro.2018.04.008
25. Alexy AB (2021) Github repository for YOLO V3. https://github.com/AlexeyAB/darknet
26. PrashantJha SS, Kispotta S (2019) A lightweight approach for an adaptive traffic signal. Jabalpur, India
27. Liu X, Wang H, Dong C (2021) An improved method of nonmotorized traffic tracking and classification to acquire traffic parameters at intersections. Int J ITS Res 19(2):312–323. https://doi.org/10.1007/s13177-020-00247-w
28. Wang Z, Cui J, Zha H, Kagesawa M, Ono S, Ikeuchi K (2014) Foreground object detection by motion-based grouping of object parts. Int J ITS Res 12(2):70–82. https://doi.org/10.1007/s13177-013-0074-8