ARPN Journal of Engineering and Applied Sciences

A CNN-BASED MODEL FOR DYNAMIC TRAFFIC SIGNAL TIMING ESTIMATION AT SIMPLE URBAN INTERSECTIONS

Camilo A. Laiton-Bonadiez, German Sánchez-Torres and Carlos Henriquez-Miranda
Grupo de Investigación y Desarrollo en Sistemas y Computación, Facultad de ingenierías, Universidad del Magdalena, Santa Marta, Colombia
E-Mail: gsanchez@unimagdalena.edu.co

ABSTRACT

The pattern of change of conventional traffic lights does not consider the density of traffic in real time, thus hampering the efficient flow of traffic. Therefore, it is necessary to create and implement a more efficient control system that could maximize the flow of vehicular traffic. This paper proposes a method to enhance the regulation control of vehicles' density at an intersection by means of the dynamic estimation of the traffic light cycle using a deep convolutional based method. The proposed algorithm is oriented to estimate traffic lights' waiting time at simple intersections in real time. Once the video is processed, this approach can estimate a traffic light cycle based on an estimated traffic volume at a given time. All of the essential aspects of the methodology and materials used for the investigation are described. Algorithm improved the average queue length at intersections by 38% and improved the average waiting time by more than 60% compared with a traditional fixed-time cycle approach. Our proposal combines multiple ideas, image preprocessing, convolutional neural networks for object detection, and a traffic time estimation method based on Webster's formulas. The proposed method, namely the dynamic estimation of the traffic signal cycle, showed a decrease in waiting times, the level of polluting emissions, and noise levels.

Keywords: convolutional neural networks, object detection, lighting control, traffic control.

INTRODUCTION

The number of vehicles in cities worldwide is continuously rising due to the increase in the purchasing power of the middle-income socioeconomic classes in developing countries. This increase also stems from the greater availability of credit, a relative reduction in prices, and a greater supply of vehicles. This has led to many problems for society, such as traffic congestion, delays, accidents, and environmental problems [1]. This indiscriminate increase in vehicles in the streets affects mobility because traffic lights are not capable of efficiently regulating traffic flow, and this contributes to traffic congestion.

The indiscriminate increase in vehicles has also caused problems with the quality of human life. An example of this is the large consumption of fossil fuels, which contaminate the environment. Furthermore, the noise generated in urban areas impacts human health, causing stress and anxiety, among other issues.

Traffic congestion requires control efforts that have been implemented in cities. The most common method is the enforcement of measures that limit mobility during periods of time (such as *pico y placa* policy in Colombia). Nevertheless, although these measures decrease traffic congestion during peak hours, thus improving travel times and speed, they tend to distribute traffic congestion along the day. This increases travel times and decreases travel speed during off-peak hours [2].

Consequently, a different approach is needed to help to mitigate these effects. This research is aimed at proposing a different approach to these mobility-limiting measures supported by new technologies, such as deep learning. However, the literature reports traditional approaches to address the mobility problem through technology. Generally, technological proposals are framed within Intelligent Traffic Systems (ITS) that use the available surveillance cameras infrastructure to detect vehicles and make measurements of characteristics related to vehicular traffic. ITS applications have different steps ranging from low-level processes including pre-processing each video frame and object detection techniques, to high-level processes such as traffic estimation or prediction scenarios.

Image pre-processing techniques are required due to most camera surveillance videos have problems that affect image quality, such as illumination, flashes of lights, low image contrast, specular reflection on cars' surfaces, and low image resolution, among others. Therefore, it is necessary to implement computer vision techniques to improve image quality. In this way, the implementation of these techniques allows for improved deep learning model predictions [3].

The image enhancement techniques applied in this study are mainly aimed at preparing the images sent to the convolutional neural network and thus improving the classification task. This stage consisted of two main steps: mean subtraction and image scaling. Mean subtraction is a technique that enhances image illumination, whereas image scaling is a normalization technique used for deep neural network performance [4]. Value normalization in images is important because it guarantees that each input parameter (pixel values) that goes into the neural network has a similar value distribution. It has several advantages, such as faster convergence and optimization. Presently, two techniques are widely used in mean subtraction. The first is per-pixel mean subtraction, whereas the other is per-image mean subtraction [4].

Object detection is a computer vision field that is oriented to identifying and locating objects in an image. Various object detection algorithms have been developed because detectors are more efficient with a specific object detection purpose defined. The history of object detection algorithms can be divided into two time periods. The first time period is before 2014, where traditional object detection algorithms-such as Viola-Jones (VJ) detector [5], the Histogram of Oriented Gradients (HOG) detector [6], and the Deformable Part-based Model (DPM) detector [7], were widely used. The second time period is after 2014, when deep learning object detection models based on convolutional neural network (CNN) architectures began to perform better than traditional methods have done [8].

Traffic time estimation techniques have received heavy attention from many researchers over the years. The main goal has been to optimize the flow of traffic, reduce pollution, and enhance the user's experience on the roads. A large variety of methods and techniques have been tested to solve this problem, each with its own strengths and weaknesses. However, they all include at least two principal characteristics: the number of cars present in a line, and the traffic light cycle.

The researchers in [8] developed a traffic control system based on wireless sensor networks and queue theory. They deployed two sensors per line and adopted a scheme with a simple scheduling algorithm that minimizes the time needed for collecting data from the components. They suggested two algorithms: one for a single traffic intersection, and the other for multiple intersections. Indeed, the second algorithm is an extension of the first one due to the indeterministic traffic flow they found in multiple intersections.

The use of fuzzy logic for traffic signal control has also been widely used given the uncertainty of a car reaching a street intersection. In [9] and [10], some fuzzy logic approaches are presented. The main objective is to determine the traffic light's green light time according to the number of vehicles present in the line. In some fuzzy logic approaches, the use of rules, queue theory, and genetic algorithms are common. It is important to clarify that these works are not just for simple intersections. They are also oriented to more complex scenarios.

Additionally, many traffic light control systems have been developed using artificial intelligence algorithms. Foy et al. [11] proposed a traffic signal timing estimation method using genetic algorithms. Their approach was geared toward finding an optimal traffic signal timing configuration for each intersection using the number of cars at each line of the intersection and the external arrival volumes that the researchers set. In addition, [12] proposed an artificial neural network to predict vehicle flow at signalized intersections. They did this to contribute to future studies related to traffic optimization and the impact of traffic flow on arterial networks (arterial roads).

This work was aimed at designing and developing a model based on computer vision techniques, digital image processing, and deep learning models that manage traffic signal timings at simple urban intersections based on measuring the volume of traffic flow during a certain period of time.

The article is organized as follows: section II describes the employed methodology. Section III describes computational experiments and obtained results. Finally, section IV discusses the conclusions and suggestions for future research.

Methodology

The employed methodology for developing this article was divided into three main phases. The first phase was focused on frame preparation (pre-processing) from a video supplied by stoplight cameras at intersections. The second phase targeted vehicle categorization and the vehicle count. The vehicle classes defined in this study are cars, trucks, and buses. The last phase was oriented toward the dynamic estimation of the traffic light cycle based on the previous phase data.

Image pre-processing

We used per-pixel mean subtraction, which performed better given the number of images we worked with. This method involves calculating the average pixel intensity of each channel and later subtracting it from its respective pixel channel across the image(s). Based on this, an RGB image can be modeled as the composition of three-color channels: red (R), green (G), and blue (B). This can be denoted as:

$$I = R \times G \times B \tag{1}$$

Where I is the RGB image, R is the red channel, G is the green channel, and B is the blue channel. Let r, g, and b be the average pixel intensity of the channels of RGB, respectively. Then, the average pixel intensities are subtracted from each channel. In other words, for every pixel in each channel, we subtract its corresponding average pixel intensity. This can be defined as:

$$\bar{R} = R - r; \bar{G} = G - g; \bar{B} = B - b \tag{2}$$

Where $\bar{R}$, $\bar{G}$, and $\bar{B}$are the channels after subtracting the average pixel intensities. Then, the final RGB image of $\bar{I}$ after the mean subtraction is the composition of $\bar{R}$, $\bar{G}$, and $\bar{B}$.

$$\bar{I} = \bar{R} \times \bar{G} \times \bar{B} \tag{3}$$

Image Scaling Normalization

Image scaling is an extra step that consists of dividing each channel pixel by the standard deviation calculated across the training set. Nevertheless, we can also scale an image into the desired range of values using a given value. Scaling factor k was set to 1/256 (each pixel has 256 values [0, 255] for RGB images). This can be defined as:

$$\begin{aligned} \bar{R} &= (R - r) / k \\ \bar{G} &= (G - g) / k \\ \bar{B} &= (B - b) / k \end{aligned} \tag{4}$$

VOL. 17, NO. 1, JANUARY 2022 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

Object Detection and Vehicle Numeration

Selecting the correct CNN is crucial to this study due to three relevant factors in this problem, such as object detection accuracy, processing speed, and computational cost. Because the proposed system is oriented to work in real time, the selected architecture must be paired with low-cost hardware located at each traffic light camera at simple intersections.

Traffic Time Estimation Method

Our proposed traffic time estimation method is based on the rational Webster's method approach [13]. This method is oriented to estimating the optimal times of fixed-time traffic lights at isolated intersections. It was based on the study of 100 intersections located in London, and that led to the development of various formulas to determine the minimum functioning cycle for the traffic lights.

It is worth noting that this method has been modified depending on the country in which it is applied as mentioned in [13]. For example, Colombia and Cuba add an extra yellow time between red and green phases to decrease the amount of green light time that drivers waste, unlike other countries.

Accordingly, the yellow time is defined by Webster's method as follows:

$$\mathrm{YT} = \mathrm{RT} + \frac{v}{2d + [2gi]} \tag{5}$$

Where RT is the driver's reaction time set to 1 second, v is the vehicle's flow speed without the traffic light's influence, g is the gravity acceleration (9.8 m/s^2), d is the deceleration rate set to 2.5 m/s^2 for all generated vehicles at each road, and i is the longitudinal pending's percentage of the road, where if the pending is negative, it is descendent, and otherwise, it is ascendent.

On the other side, Webster's method defines the green light time distribution in the traffic light cycle as follows:

$$g_i = \frac{y_i}{Y}(C_0 - L) \tag{6}$$

Where y_i refers to the charge factor for that specific traffic light. The charge factor is nothing else than the flow rate that exists at that moment on the road. Hence, it can be calculated as follows:

$$y_i = \frac{q_i}{s_i} \tag{7}$$

Where q_i represents the arrival flow rate per time unit (usually seen as vehicles per hour), and where s_i represents the saturation flow related to the number of vehicles per green light unit time for the traffic light. The saturation flow is a crucial element that tells us the ideal flow for the road if the traffic light were always green so as to allow for a constant flow of vehicles.

The saturation flow used was $1650\,\mathrm{v}\,/\,\mathrm{g.h}$, which was the representative value [13] given in 1993 for Medelling roads featuring lane widths between 3 and 5 meters.

On the other hand, Y can be calculated as follows:

$$Y = \sum_{i=1}^{n} y_i \tag{8}$$

Where n is the number of movements existing at the intersection.

Equally important is the optimal cycle length for the traffic signal light. This value is considered because it provides a duration cycle time that minimizes the delay time for all vehicles using the intersection. It could be calculated as follows:

$$C_o = \frac{kL + 5}{1 - y};\ 0.75 \leq k \leq 1.5 \tag{9}$$

Here, we use k equal to 1.5 as recommended in the literature, but it could be modified depending on the intersection.

Finally, at every intersection, we usually have wasted time that should be calculated to better determine the optimal green light time at the intersection to allow the largest number of vehicles to pass. This time can be calculated as follows:

$$L = \sum_{\varphi=1}^{n} (I + \lambda_1 + \lambda_2) \tag{10}$$

Where L is the total delay time lost per cycle at the intersection. λ_1 refers to the lost time required for the vehicle to start, and λ_2 is the time saved in the yellow phase between red and green. This last value is used in countries such as Colombia and Cuba, where this policy was applied.

Where $I = YT + AR + (R/A)$. We avoided the use of R/A considering that we do not use the yellow time between red and green phases. YT refers to the yellow time mentioned before, and AR (all red) is an instant where all traffic lights are red. Indeed, all red is an important measure because it is the time required for a vehicle to stop before it collides with the vehicle that gains the right of way. This last one can be calculated as:

$$RR = \frac{d_i + l_i}{v_i} - \frac{d_{i+1}}{v_{i+1}} \tag{11}$$

Where d_i is the distance in meters from the stop line of the movement that loses the right of way to the conflict point of the movement that gains the right of way. l_i is the vehicle length, and v_i is the speed of the vehicle that loses the right of way in meters per second $\left(\frac{m}{s}\right)$, v_{i+1} is the speed of the vehicle that gains the right of way. Finally, d_{i+1} is the distance from the stop line of the movement that gains the right of way to the conflict point of the movement that loses the right of way.

Based on the formulas mentioned above, our objective was directed toward the dynamic estimation of

VOL. 17, NO. 1, JANUARY 2022 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

the optimal cycle length using arrival flow traffic q at the intersection for each time interval using a CNN.

RESULTS

Presently, many databases provide videos and images with vehicular traffic information. Unfortunately, these videos and images do not obey the necessary camera configuration to implement this kind of software correctly. Hence, faced with the difficulty of acquiring these types of videos, it was necessary to utilize graphic video data whose quality and configuration were comparable to the desired quality and configuration. These videos can be found on the internet for public use.

Image Preprocessing and Selection of a CNN Architecture

After the image preprocessing step, we can notice an improvement in the image illumination, which helps the CNN to achieve better results. Figure-1 shows an example of the preprocessed image after per-pixel mean subtraction and image scaling.

To select a CNN architecture, we conducted a literature review to select the most suitable CNN architecture. The CNN that best suits our three relevant factors-object detection accuracy, processing speed, and computational cost-is “You Only Look Once V3” (YOLOv3) due to its unique characteristics (see Table-1).

It is also important to mention that this CNN has a smaller variation called Mixed YOLOv3-LITE, which allows us to use it in embedded devices [14]. However, the use of this variation has one disadvantage due to its smaller representation. It loses accuracy but increases its speed and achieves fewer floating-point operations per second (FLOPS).

For the object detection purpose, we used a pre-trained YOLOv3 model with the Microsoft COCO dataset [15]. This dataset contains 80 object categories where our targeted objects were included. The YOLOv3 model has problems with directly aligning the box with the detected object as shown in Figure-2. We were concerned only with detecting the vehicles passing in front of the traffic light for our study, rather than identifying the entire object perfectly.

To detect our targeted objects, we used a minimum confidence value set to 0.7 by default. Then, once YOLOv3 detected a possible object, we ensured that the classification accuracy was above 70%. Afterward, we counted the number of cars, trucks, and buses visible in our video. As we can see in Figure-2, the YOLOv3 model can detect our targeted objects in our example image with high accuracy (>90%), even if they are far away from the installed camera. However, this accuracy highly depends on image camera quality, weather conditions, and the right camera angle.

Computational Experiments

The mathematical equations above were evaluated with a traffic simulator program called SUMO (Simulation of Urban MObility) [16], which is widely used to facilitate the evaluation of infrastructure, policies, control algorithms, and other parameters before they are implemented in the real world. This software is an open-source, continuous road traffic simulator coded in C++ to model travel systems. This simulator was chosen due to the many applications it offers, as well as its compenetration with the Python programming language. This allows us to control the ongoing simulation depending on what is happening on the simulated roads.

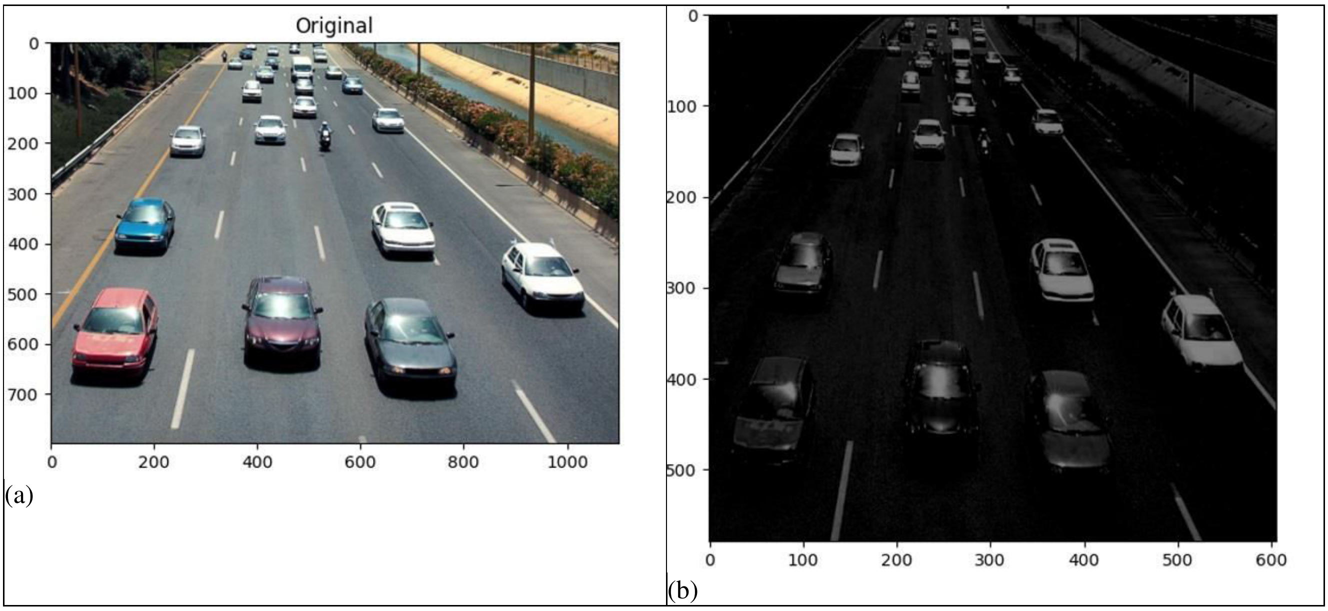

Figure-1. Image after mean subtraction and scaling. (Source: Author).

VOL. 17, NO. 1, JANUARY 2022 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

Table-1. Performance of YOLOv3 models on the COCO Dataset.

Models for YOLOv3	mAP	FLOPS	FPS
YOLOv3-320	51.5	38.97 Bn	45
YOLOv3-416	55.3	65.86 Bn	35
YOLOv3-608	57.9	140.69 Bn	20
YOLOv3-tiny	33.1	5.56 Bn	220
YOLOv3-spp	60.6	141.45 Bn	20
mAP: Mean Average Precision FLOPS: Floating-Point Operations Per Second FPS: Frames Per Second			

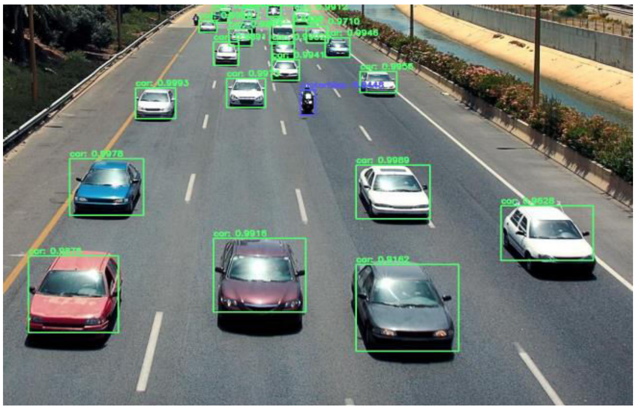

Figure-2. Detecting objects with YOLOv3 model. (Source: Author).

For our experiment, we created a simple intersection with four stoplights and 750-meter roads with four lanes each, where we randomly generated cars. In this work, we are using only standard cars. However, SUMO allows for other types of vehicles, such as bicycles, trucks, taxis, and trailers.

The modeled intersection has four directions and eight possible movements where all vehicles can go straight. It is important to mention that SUMO uses steps as the time counter, where one step represents one second but provides us with the possibility of speeding up the simulation steps to achieve faster results. For the generated cars, we set a maximum speed of 12 m/s (meters per second), a maximum acceleration of 1 m/s^2, a maximum deceleration of 2.5 m/s^2, and a minimum gap between cars of 2.5 meters (see Table-2).

To simulate a real traffic flow environment in the modeled intersection, traffic generation is essential [17]. We set a bundle of 24 episodes, which refers to the 24 hours in a day. In each episode, we used a specific distribution function based on that specific hour's desired behavior.

Accordingly, we used the Weibull distribution function to simulate incremental traffic flow in our peak hours for our modeled intersection and a constant distribution for most of the other hours. Also, we used a Laplace function to simulate a sudden traffic jam to evaluate our proposed traffic time estimation method in this type of situation. The last distribution function should be considered due to the traffic flow present when a flow of cars is released from a previous traffic jam. It was implemented just for hour number 20.

Similarly, we simulated the CNN task on the simulator by gathering information about the cars within a range of vision set to 100 meters, starting from each traffic light.

Inside this range of vision, we get the road's actual arrival vehicle volume q during a t time interval, which we refer to as the analysis time. We set t to 300 seconds (five minutes) for our experiments. This means our optimal cycle length will change each 300 seconds depending on the arrival volume for the time interval used in the formulas above to estimate the new optimal cycle length.

It should be pointed out that the analysis time can be changed depending on the behavior we want for the traffic light. Indeed, the analysis time is crucial because it determines how many changes per hour the cycle length will have. Therefore, it will affect the average waiting time per vehicle at the intersection, the average queue length, and the CO2 emissions on the road, among others factors. Consequently, if this value is too high, almost no change will occur regarding a fixed-time estimation approach due to an extended period of time. This happens because the shorter the analysis time is, the greater the chances that the arrival flow rate will remain with the previously estimated cycle length or at least change slightly. On the other hand, if this value is too small, the intersection's cycle length would be highly dynamic, thus leading to an undesired behavior due to the lack of information. For instance, this value should be in the middle of the two sides.

After the analysis time ends and we correctly gather the arrival flow rate, we proceed to change the cycle length to the new optimal cycle length for that time interval. To do so, and considering that we have only straight movement on the roads, we compare the arrival flow rates between the traffic lights that share the same time circuit. This means we compare arrival flow rates between north and south, then west and east, to later select the highest in each comparison.

With these two arrival flow rates, we then adjust the optimal cycle length for each road, assuming that north and south will have the same cycle length, and the same for west and east. It is important to clarify that a minimum and a maximum value limit the calculated green light time distribution within the traffic light cycle. They are set to 10 and 60 seconds, respectively.

In regards to the use of Webster's formulas in our experiments, we used a longitudinal pending's percentage of 0.00 (0%) for all roads to model the ideal traffic light intersection. However, this could be changed depending on the studied road. With our approach, the yellow time can easily be calculated given the fact that we need to know some of the specific characteristics of the road on which this method will be implemented.

Also, we avoided the use of $\lambda_1$and λ_2, as they do not approach reality when these values are taken into a

ARPN Journal of Engineering and Applied Sciences

traffic simulator. For the value of λ_1, we noticed that vehicles do not take much time to start because of their own simulators. We excluded λ_2, since we are not using the yellow phase between red and green. Accordingly, Eq. 10 was reduced to:

$$L = \sum_{\phi=1}^{n} I \tag{12}$$

To test our approach, we created two scenarios: one for our approach and another for the traditional fixed-time signal timing. For both scenarios, we gathered information from the simulations (10 simulations per scenario), such as the average waiting time per vehicle, the average queue length, the CO2 emissions, and the generated traffic distribution. In Figure-3, we can see the traffic distribution for both scenarios at hour 12.

Average queue length at the intersection

Figure-4(a) shows an average queue length comparison between the two aforementioned scenarios. This metric, which was measured for an interval of 300 seconds, was calculated by identifying the largest queue length per traffic light when the light was red. Then, to graph the queue length behavior throughout the entire intersection, we took the average queue length value for that time interval. Finally, when all 10 simulations were completed, we calculated the average of all of the results per hour (this was done for all metrics) due to the simulations' stochastic behavior.

Table-2. Modeled intersection's configuration per road.

Road	M	Car's maximum speed (m/s)	Car's maximum acceleration (m/s^2)	Car's maximum deceleration (m/s^2)	Car's minimum gap (m)
North to South	S	12	1	2.5	2.5
South to North	S	12	1	2.5	2.5
West to East	S	12	1	2.5	2.5
East to West	S	12	1	2.5	2.5
Allowed movements for the road (turn left(L), turn right(R), go straight (S))					

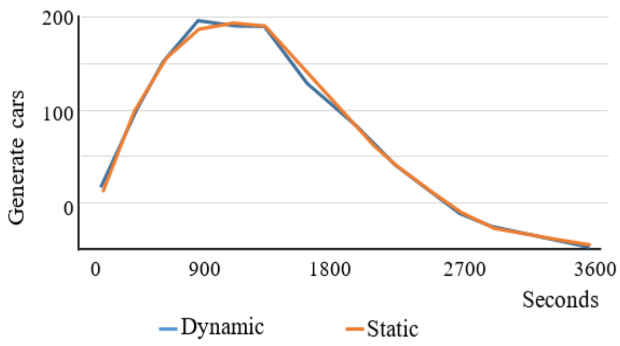

Figure-3. Traffic distribution comparison between scenarios during hour 12. (Source: Author).

The proposed time estimation method showed an improvement in the average queue length at the intersection directly related to the amount of time that vehicles needed to wait to cross the intersection. Many studies use queuing theory to decrease waiting times per vehicle as mentioned in Section 2.3.

Average Waiting Time Per Vehicle

Consequently, the average waiting time per vehicle decreased with our dynamic approach as expected from Figure-4 (a). In this graph, we note that our approach behaved similarly to the traditional one at the beginning of the episode, but after 300 seconds, the average waiting time decreased.

This occurred because our analysis time was set to 300 seconds for this experiment, so the algorithm began gathering information from the road. After it captured the traffic flow for that time interval and determined that the traffic light cycle was too high (caused an increase in the waiting time), the algorithm proceeded to adjust the traffic light cycle with the formulas described in Section 2.3. Consequently, it is highly recommended to start the algorithm operation at a time period when traffic flow is not too heavy so that the traffic light cycle change will not be as drastic.

CO2 Emission and Fuel Consumption

The congestion on the roads due to the incredible growth of traffic increases the amount of pollutants in our environment and the growing health problems associated with this [18]. Therefore, it is imperative to evaluate the environmental impact of our approach. For this purpose, the SUMO simulator provides us with several emission models as follows:

- HBEFA (The Handbook Emission Factors for Road Transport) in versions v2.1 [19] and v3.1 [20].
- PHEMLight (derivation of Passenger Car and Heavy Duty Emission Model, or PHEM).
- Electric vehicle model [21].

We used the HBEFA v3.1, which is the default emission model that SUMO provided. In Fig. 4(c), we can see an improvement in the CO2 emissions from cars. This

VOL. 17, NO. 1, JANUARY 2022 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

is possibly because during traffic congestion, vehicles spend more time on the road accelerating, decelerating, crawling, or even idling along the road (we can see this behavior in Figure-4 (b) in the static series), which increases harmful or CO2 emissions [22].

In Figure-4 (b) and Figure-4 (c), we show that we were able to reduce the amount of time that vehicles spend on the road. Figure-4 (d) shows the fuel consumption and how it is proportional to the CO2 emissions as expected.
For the purpose of generating CO2 emissions, fuel consumption, and noise emission graphs, the SUMO simulator allowed us to gather all of the information per car for each second along all of the roads. It was not limited by our simulated CNN vision range on the simulator. We did this because we wanted to have a general overview of the intersection and how our approach worked.

Noise Emission

As we mentioned before, the increased number of vehicles on the roads is leading to an increase in the number of diseases and their exacerbation, with one of them being hearing problems, especially in people living near congested roads. This is known as noise pollution.

In our experiments, we expected that noise emission would not drastically decrease during the simulations because we generated the same number of cars at the intersection using the same distribution function. However, after the maximum number of cars is generated, the noise emission from all of the roads starts to decrease. This may be because the number of cars that have already crossed the intersection is higher in our approach than in the traditional approach, and therefore, fewer vehicles would be at the intersection and generating noise pollution.

Also, Figure-4(e) shows that noise emission reaches 300 decibels (db). We sum up the noise emission from all of the roads (north, south, east, and west). This would result in a maximum value of 75 db per road, which is high because this is during a peak-hour-traffic simulation.

Thus, the proposed system is able to alleviate the congestion problem associated with fixed-time traffic signal light approaches focused on reducing contamination, fuel consumption, the average waiting time that users spend at traffic lights, and the average queue length, among others. Another contribution of this paper is the flexibility that this method provides by adapting this algorithm to the intersection infrastructure, which influences vehicular flow. After comprehensive consideration, this method proves to be capable of achieving better results compared with traditional approaches.

ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

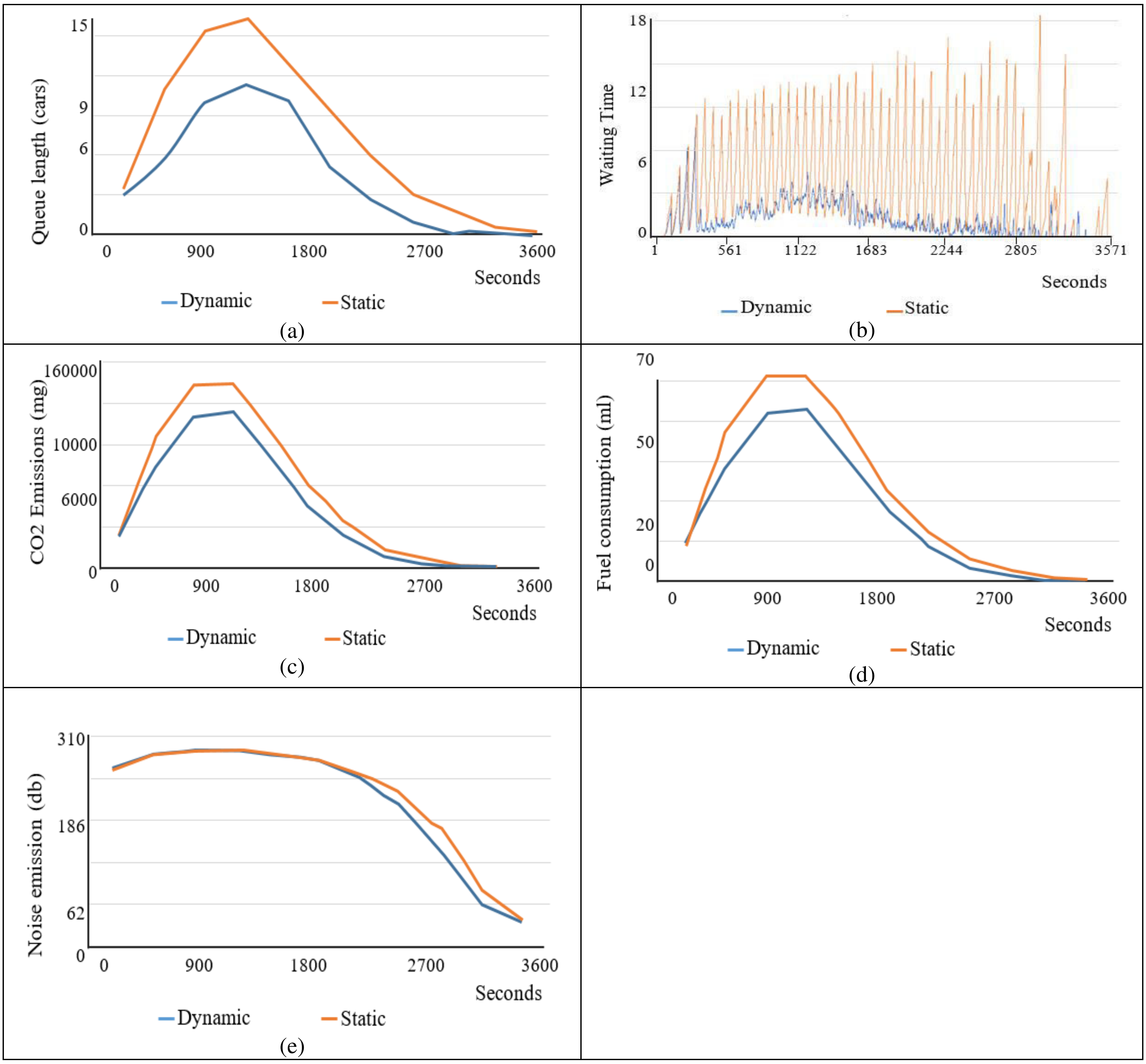

Figure-4. Results of comparison between scenarios during hour 12 a) Queue length, b) Waiting time, c) CO2 emission, d) Fuel consumption and e) Noise emission. (Source: Author)

CONCLUSIONS

In this study, we proposed a model for the dynamic estimation of the traffic signal timing at simple urban intersections. This model uses in its architectural design a CNN oriented to embedded devices. The adaptative traffic signal time is based on the traffic flow at the intersection for a period of time, which allows us to use Webster's formulas to estimate the best traffic signal time cycle that allows as many vehicles as possible to pass without affecting other vehicle flows.

To test our approach, we developed two scenarios in the simulator. The first scenario uses a traditional traffic signal light fixed-time cycle, whereas the other uses our dynamic method. Our approach was able to decrease the average queue length at intersections by 38% and improved the average waiting time by more than 60%.

In the development of this study, we faced some limitations that should be further explored and improved upon. For example, the traffic time estimation depends highly on image quality because of the use of convolutional neural networks. Environmental conditions such as rain, snow, and dark environments will result in poor quality images leading the network to perform poorly. In this case, more advanced methods should be carried out to improve image quality.

Also, additional studies should be done on this approach to ensure human safety on the roads, even though the SUMO simulator already applies safety-related measures for human safety (Surrogate Safety Measures SSM).

REFERENCES

[1] Alberto Bull. 2003. Traffic Congestion: The Problem and how to Deal with it. United Nations Publications.

[2] Posada J., Farbiarz V. and Gonzalez-Calderon C 2011. Analisys of Pico y Placa as vehicular circulation restriction in Medellín - based on traffic volumes Dyna. 78: 112-121.

[3] A. Krizhevsky, I. Sutskever and G. E. Hinton. 2017. ImageNet Classification with Deep Convolutional Neural Networks. Commun. ACM. 60(6): 84-90, doi: 10.1145/3065386.

[4] K. K. Pal and K. S. Sudeep. 2016. Preprocessing for image classification by convolutional neural networks. 2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), Bangalore, pp. 1778-1781, doi: 10.1109/RTEICT.2016.7808140.

[5] P. Viola and M. Jones. 2001. Rapid object detection using a boosted cascade of simple features. Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, Kauai, HI, USA, pp. I-I, doi: 10.1109/CVPR.2001.990517.

[6] N. Dalal and B. Triggs. 2001. Histograms of oriented gradients for human detection. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 2005, pp. 886-893 vol. 1, doi: 10.1109/CVPR.2005.177.

[7] P. Felzenszwalb, D. McAllester and D. Ramanan. 2008. A discriminatively trained, multiscale, deformable part model. 2008 IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, pp. 1-8, doi: 10.1109/CVPR.2008.4587597.

[8] Ahmad Yousef, Khalil and Al-Karaki, JN & Shatnawi Ali. 2010. Intelligent Traffic Light Flow Control System Using Wireless Sensors Networks. J. Inf. Sci. Eng.. 26. 753-768.

[9] N. Shahsavari Pour, H. Asadi, y M. Pour Kheradmand. 2013. Fuzzy Multiobjective Traffic Light Signal Optimization. Journal of Applied Mathematics, 2013: 249726, doi: 10.1155/2013/249726.

[10] S. Mohanaselvi y B. Shanpriya. 2019. Application of fuzzy logic to control traffic signals. AIP Conference Proceedings, 2112(1): 020045, doi: 10.1063/1.5112230.

[11] Foy, M. D., Benekohal, R. F. and Goldberg D. E. 1992. Signal Timing Determination Using Genetic Algorithms. In Transportation Research Record 1365, TRB, National Research Council, Washington, D.C. pp. 108-115.

[12] S. Lawe y R. Wang. 20163. Optimization of Traffic Signals Using Deep Learning Neural Networks. en AI 2016: Advances in Artificial Intelligence, Cham, 9992: 403-415, doi: 10.1007/978-3-319-50127-7_35.

[13] Valencia Alaix V. 2000. Principios sobre semáforos. Colombia, Medellín: Facultad Nacional de Minas.

[14] H. Zhao, Y. Zhou, L. Zhang, Y. Peng, X. Hu, H. Peng, and X. Cai. 2020. Mixed YOLOv3-LITE: A Lightweight Real-Time Object Detection Method. Sensors, 20(7): 1861, doi: 10.3390/s20071861.

[15] T.-Y. Lin *et al*. 2014. Microsoft COCO: Common Objects in Context, en Computer Vision - ECCV 2014, Cham, 8693: 740-755, doi: 10.1007/978-3-319-10602-1_48.

[16] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz. 2011. SUMO - Simulation of Urban MObility: An Overview. In SIMUL 2011, [Online]. Available: https://elib.dlr.de/71460/.

[17] Vidali A., Crociani L., Vizzari G., Bandini S. 2019. A Deep Reinforcement Learning Approach to Adaptive TrafficLights Management. In Proceedings of the WOA, Parma, Italy.

[18] Krzyzanowski M., Kuna-Dibbert B., Schneider J. 2005. Health effects of transport-related air pollution. WHO, Denmark.

[19] Keller M., de Haan P., Knörr W., Hausberger S, Steven H. 2004. Handbuch Emissions faktoren des Strassenverkehrs 2.1: Dokumentation. Bern, Heidelberg, Graz, Essen, INFRAS

[20] S. Hausberger. 2009. Emission Factors from the Model PHEM for the HBEFA Version 3.

[21] T. Kurczveil, P. Á. López, and E. Schnieder. 2014. Implementation of an Energy Model and a Charging Infrastructure in SUMO. Simulation of Urban Mobility, 8594: 33-43, doi: 10.1007/978-3-662-45079-6_3.

[22] R. Smit, L. Brown, and Y. C. Chan. 2008. Do air pollution emissions and fuel consumption models for roadways include the effects of congestion in the roadway traffic flow?. Environmental Modelling and Software, 23 1262-1270, doi: 10.1016/j.envsoft.2008.03.001.