ICIC Express Letters
Volume **15**, Number **3**, March **2021**

ICIC International ©2021 ISSN 1881-803X
pp. **297–305**

SMART TRAFFIC LIGHT MODEL USING DEEP LEARNING AND COMPUTER VISION

Edward Pratama Putra, Stefanus Michael, Taurean Orlin Wingardi
Raffael Lucas Tatulus and Widodo Budiharto*

Computer Science Department
School of Computer Science
Bina Nusantara University
Jl. K. H. Syahdan No. 9, Kemanggisan, Palmerah, Jakarta 11480, Indonesia
*Corresponding author: wbudiharto@binus.edu

Received June 2020; accepted September 2020

Abstract. *Traffic light duration management is still generally done manually, creating a duration imbalance in intersections with differing number of vehicles in each direction, thereby contributing to traffic congestions. This research was conducted to create a smart traffic light model that can dynamically adjust duration based on the current and predicted number of vehicles in each direction. The method offered in the model involves the usage of Arduino as the traffic light and incorporates a calculation function to adjust the traffic lights duration and a deep learning algorithm to predict the vehicle amount of each direction in 30 minutes from the current time stamp. The experiment is conducted using model cars placed in front of a traffic light model and comparing the calculated number of vehicles passing and remaining of the dynamically adjusted duration, to the estimated amount of a set duration. From the test data set, the total number of vehicles that is calculated to pass on a static duration is 167, 175 on the computer vision determined duration, and 172 on the predictive deep learning determined duration. The comparison of vehicles remaining nets a result of 28 for a static duration, 20 for the computer vision determined duration, and 23 for the predictive deep learning determined duration. The results indicate that by using computer vision, the traffic congestion rate can be decreased by 28.57%, while deep learning prediction can reduce it by 17.86% preemptively.*
Keywords: Smart traffic light, Deep learning, Computer vision, Automated traffic management

1. **Introduction.** Daily commutes often occur on public roads with multiple lanes and directions. These different directions can cause traffic to pile up and become convoluted, especially in intersections. To manage the traffic on intersections, traffic lights are normally used. Traffic lights or stop lights can be defined as a set of lights to control traffic on two or more intersecting road [1]. The lights can come in a set of two to three and in the colors red (to signify stop), yellow (to signify preparation for a change of lights), and green (to signify go). Using this system, vehicles need to wait for the traffic light to turn green before proceeding forward, which gives time for vehicles from the other directions to move. Alternatively, when vehicles from a certain direction are moving forward in traffic, the traffic light(s) on the other directions will simultaneously turn red to avoid traffic collision.

Traffic lights have become a widely accepted solution to traffic collision avoidance in bustling intersections. The current system of traffic light operates on a static timing system that is set by the local traffic administrator. This system can directly contribute to the occurrence of traffic jams by imposing the same pre-adjusted wait period (signified by the red light) to all directions, regardless of the number of vehicles in the lanes, which can

DOI: 10.24507/icicel.15.03.297

cause directions that are already jammed to further be filled with vehicles, while directions with a smaller number of vehicles are unable to efficiently utilize the given duration of green light, causing the allotted time to be rendered redundant. Deep learning is a method used in neural networks to correctly assign credits in each transformation stage, so that neurons in a neural network can be transformed to form a desired output/behavior [2].

Besides the utilization of deep learning to find the optimal traffic light duration, computer vision is also used in this research model, in order to reduce the reliance of manual data input in the deep learning process. Computer vision is a field of artificial intelligence that aims to process signals, in order to extract suitable features that can be processed to recognize and classify objects [3]. In this research, the academic contribution is a traffic light model constructed in a model intersection. The traffic light model will then enter the initial learning phase where example historical data will be inputted for the model vehicles. Afterwards, the traffic light will be tested using model vehicles. Computer vision is used to identify the amount of model vehicles in each direction, for the continuous adjustments via deep learning to occur. The results of the dynamic timing traffic light system will then be measured and compared with the calculation result of a model traffic light with static timing.

2. Literature Review.

2.1. Deep learning.

Deep learning is a type of machine learning algorithm used in neural networks. The character of deep learning is self-learning by making more than one model and teaching it with big quantity of data, while the approach used is likeness-training approach with numerous standards of likeness, acquired by creating uncomplicated, but non-consecutive unit that it alters, so that the likeliness of the unit to a single standard can become closer to a greater and more abstract standard [4]. The training is done using a common intention training rule [5]. Usually, algorithm of deep learning is formed on a developmental algorithm of stochastic gradient descent [6]. Using this approach and algorithm, traditional learning drawbacks such as the requirement of detailed descriptions for a large number of data can be alleviated by supporting computational units created from numerous processing levels that do not require manual input of information about the data [7].

A method to use deep learning to create and predict a traffic model was proposed by Zhao et al. [8]. This method is quite complex as it built a model that directly reflects real traffic scenarios, as opposed to the proposed experimental model of this research. The model can be seen in Figure 1, where X is the input variable, W and U represent the coefficient matrixes, O indicates outputs, V signifies the vector, and S to indicate temporal variables.

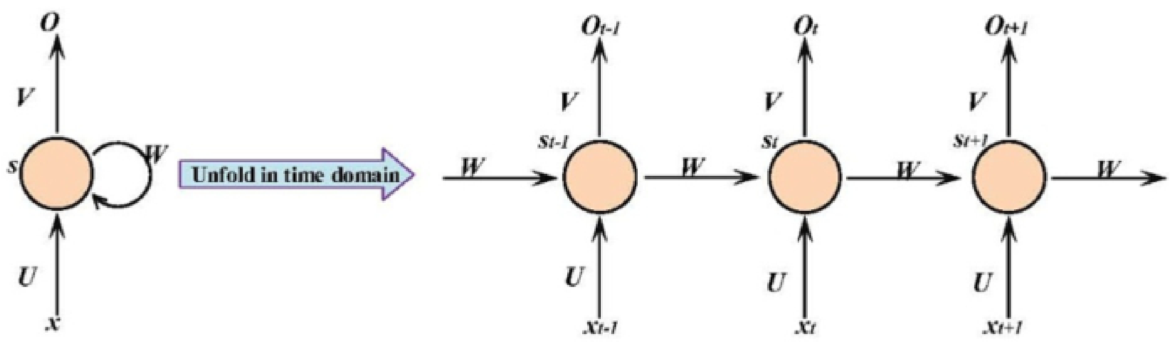

FIGURE 1. RNN structure in deep learning for traffic prediction [8]

Another usage of deep learning comes in the form of vehicle detection, in a method proposed by Zhang and Lin [9] with the method of preprocessing the images before training ImageNet before the ROI model. The convolutional layer will then add an ROI pooling layer to locate the ROI to the image.

2.2. **Computer vision.** Computer vision is a field of science and engineering that aims to make automated systems that can give useful data from real physical scenes based on image or video data [10]. To get information from an image, the computer vision algorithm will do object detection and classify the image with a model that has already been trained with some of the data set. Object detection in computer vision is done by preprocessing the raw image with denoising methods, to reduce unwanted noise. Afterwards, the algorithm will segment the image before extracting the image feature using a system called scale invariant feature transform (SIFT). The extracted features will then be classified to a model that has already been trained to do the classification using methods such as K nearest neighbor, decision tree, neural network, or support vector machine [11].

Previous research has been done by Seenouvong et al. [12] to identify and count vehicles from a road security camera footage by using computer vision algorithm. The method used is to first find the image of foreground objects by using background subtraction technique. The object found is then enhanced and noise reduced by using techniques such as adaptive morphological operations. Then, the centroid of the object is calculated to find the position of the vehicle. Afterwards, vehicles can be counted in a predefined zone. The example result of this algorithm can be seen in Figure 2.

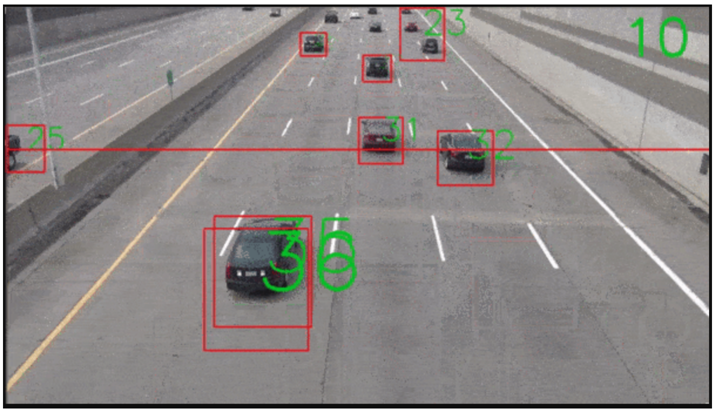

FIGURE 2. Example of vehicle counting result using computer vision algorithm

2.3. **Arduino.** Arduino is an open source platform that can be used to create prototypes. Arduino is equipped with both hardware, such as microcontrollers, and software that can be used to develop programs for the prototype [13].

The process of writing and executing a code in Arduino consists of writing the code in the provided IDE, verifying the syntax for mistakes, connecting the hardware component (controller) to the computer, and pressing upload for the controller to run the code [15].

2.4. **Traffic light duration management.** The duration of traffic lights is generally set manually by officials from the public transportation department. The duration is set based on experience, prior traffic data, or parameters such as average queue length and cumulative waiting time [16]. Manually setting traffic lights duration can be non-optimal, as traffic predictions can be inaccurate and officials are not always on standby during traffic jams to provide appropriate adjustments. To combat this problem, methods of vision-based systems [17] and infra-red based systems [18] have been used to create current generation of smart traffic lights to reduce traffic congestions.

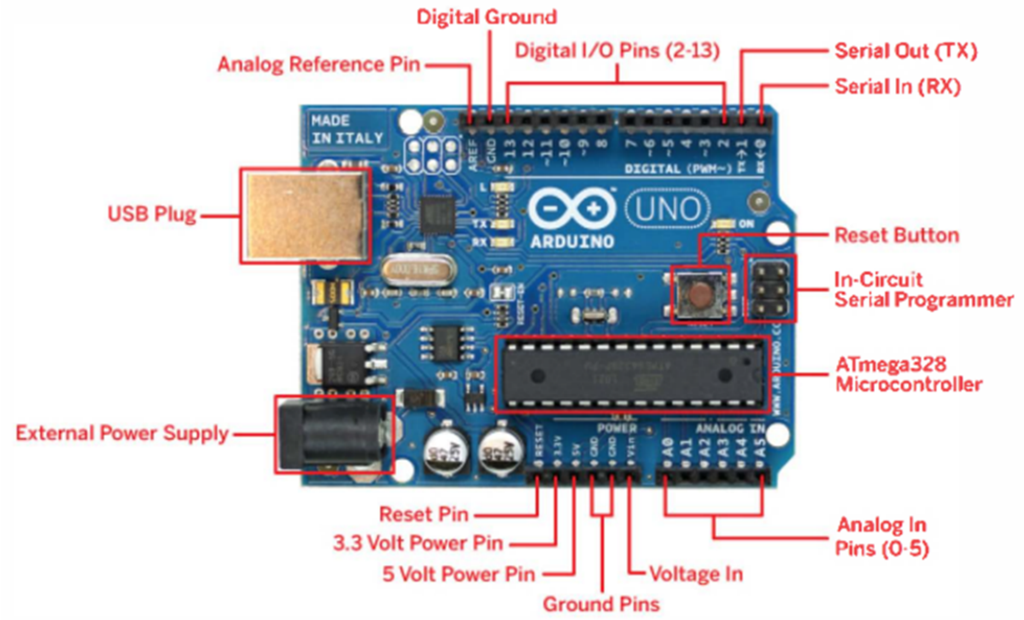

Figure 3. Arduino Uno [14]

3. **Proposed Method.** The traffic light model consists of a model three-way intersection made from cardboard, 9 LED lights consisting of 3 red, 3 yellow, and 3 green colored lights, to simulate the traffic lights. The lights are controlled using an Arduino Uno R3 controller connected to a computer. OV7670 Camera modules for the Arduino controller are also placed on each side of the model intersection. The controller is also equipped with a DS3231 real time clock module, to keep track of the time. Model vehicles will also be added to the intersection in the form of model cars sizing 6-8 cm. A block diagram depicting the components can be seen in Figure 4.

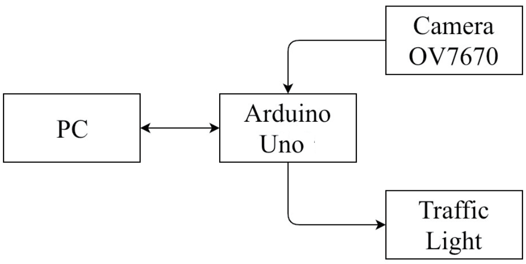

Figure 4. Block diagram for the smart traffic light model

Arduino is used to obtain image data for the computer vision algorithm alongside the image time stamp and to control the traffic lights model by turning individual LEDs (representing red, yellow, and green lights) according to a specified or calculated duration. A program to control the LED lights can be seen in Algorithm 1 for controlling a traffic light.

Algorithm 1. Pseudocode to control a traffic light

Begin
Turn on the red light
Set the yellow light duration
Set the green light duration
For each of the traffic light do

Turn on the yellow light
Delay for yellow duration
Turn off red light
Turn off yellow light
Turn on green light
Delay for green duration
Turn off green light
Turn on yellow light
Delay for yellow light
Turn off yellow light
Turn on red light
End

Computer vision is used to identify and count the number of vehicles queueing in each direction of the intersection. The camera modules are used to provide video data to the computer vision algorithm. The algorithm will then identify each vehicle in a lane before processing the gained data to produce several parameters. The parameters gained from the computer vision algorithm include the number of vehicles on each direction and the length of traffic congestion. A diagram outlining the computer vision process can be seen in Figure 5.

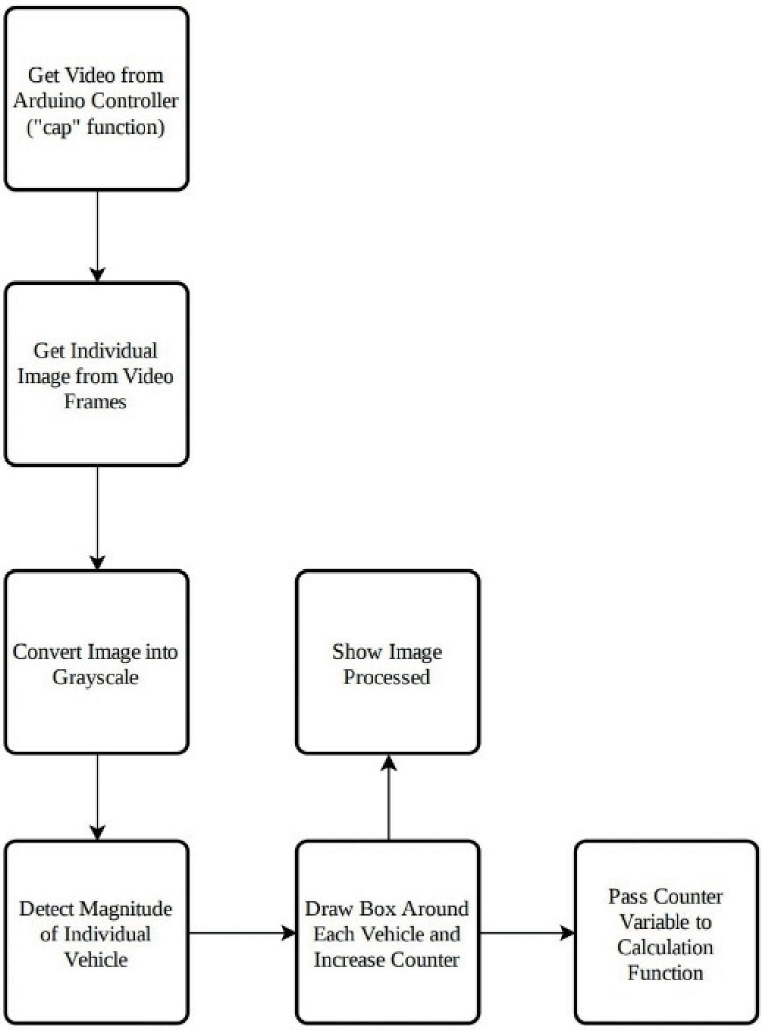

Figure 5. Computer vision activity diagram

The computer vision algorithm is implemented using OpenCV library, specifically the CV2 module [19]. This module is chosen due to both ease of usage and the amount of provided inbuilt functions that are specifically tailored for computer vision appliances. In order to get the video data passed from the camera modules on the Arduino controller, the "cap" function is used.

After getting the video data, vehicle model object classification is conducted using XML. The object classification is done by first getting individual images from the video frames. The images are then converted into grayscale. Afterwards, from the shade of black and white, the magnitude of an individual vehicle can be detected. A box is then drawn around the vehicle and shown to the screen, while the vehicle counter variable is increased by one. This classification process is done repeatedly by the library, in order to detect all the vehicles in the video input. The computer vision pseudocode can be seen in Algorithm 2 for reading and training the dataset, and testing the test data.

Algorithm 2. Computer vision pseudocode for vehicle detection

```
Begin
  Read dataset for training the model
  Train the model using the dataset
  Capture image from traffic camera
  Change captured image color space
  Count detected vehicle
End
```

Deep learning is used to create a prediction model in order to predict the occurrence of traffic congestion, so that the duration of the smart traffic lights model can be adjusted in anticipation for the influx of traffic. The computer vision algorithm provides the number of vehicles in each lane of the intersection, while the real time clock module provides the time stamp for the vehicle amounts. Utilizing the time stamps and vehicle amounts, a pattern can be produced to predict traffic congestion based on the time of the day on each lane. The library used for the deep learning algorithm consists of numpy and keras. Numpy is used to perform required calculations, while keras is used to create the neural network layers for the deep learning process. After the initial learning phase is concluded, the pattern and prediction result will be stored and passed to the smart traffic light duration calculation function. The prediction results will then be compared to the actual vehicle amount on each lane from the computer vision algorithm in order to find the prediction deviation rate. Further training will then be done in order to find a more fitting pattern. The pseudocode for the deep learning process can be seen in Algorithm 3.

Algorithm 3. Deep learning pseudocode for the number of vehicle prediction

```
Begin
  Initialize model for deep learning
  Read train data
  For every 30 minutes
    Read live traffic data
    Train the model with live traffic data
    Get number of car prediction
End
```

The duration of the traffic lights is calculated using simple mathematic division of the total duration of green light in all the traffic lights. For simplicity, the default duration of a green light in the model traffic light is set as 18 seconds, creating a total of 54 seconds of green light duration across a three-way intersection. Then, based on the number of vehicles in the lane compared to the total amount of vehicles (obtained from the either the computer vision algorithm or the deep learning algorithm), a weight is assigned to each lane. Based on the assigned weight, the total green light duration is divided and rounded to the nearest second. The red-light duration is adjusted by subtracting the total green time duration with the lane's green light duration. The result of the automatically adjusted traffic light duration can be measured by calculating the total amount of vehicles that are able to pass the intersection on a cycle of green lights (54 seconds). The number of vehicles passing the intersection from a specific direction is calculated by multiplying the green-light duration with the number of lanes in the direction. The result is then divided by the average time required for a car to pass a traffic light to obtain the vehicle total. The total amount of vehicles passing will then be rounded down to the number of cars in queue on the direction, if the total is higher, to avoid over calculation. In order to simplify the calculation, the average time for a model car to pass the traffic light model is predetermined to be 3 seconds. After obtaining the number of vehicles passing the intersection from every direction as a total, evaluation can be done by comparing the total to the result of the same calculation performed using a fixed green light duration for every direction of the intersection.

4. **Experimental Result.** The smart traffic light model is made by using an Arduino Uno to control connected LED lights signifying the red, yellow, and green light of a traffic light. The lights are turned on and off according to a dynamically calculated duration. The camera modules attached to each traffic light model can capture image data every 30 minutes. The image, alongside the respective time stamp is then sent to the computer vision algorithm coded in Python 3 using the OpenCV library. The algorithm successfully discerned and counted most vehicles in the captured image data as shown in Figure 6. The resulting vehicle count of every direction is stored to be used by the calculation function as shown in Table 1.

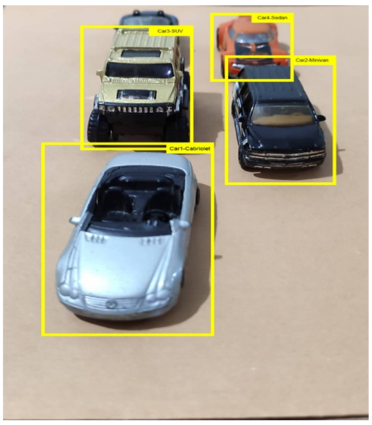

Figure 6. Model vehicle detection result

Table 1. Stored vehicle amounts from test data

Time Stamp	Direction 1	Direction 2	Direction 3	Total
7:00:00	5	3	7	15
7:30:00	8	6	3	17
8:00:00	10	8	5	23
8:30:00	6	4	8	18
9:00:00	3	9	7	19
9:30:00	2	5	5	12
10:00:00	4	1	6	11

The deep learning algorithm can predict the vehicle amount of each direction 30 minutes from a current time stamp, with an accuracy rate of 75.4%. The prediction is trained by using vehicle amounts data gathered by the computer vision algorithm. The predicted value is then passed to the duration calculation function to obtain a predictive green lights duration for every traffic light. The prediction value for the time stamp of the test data can be seen in Table 2.

Table 2. Deep learning vehicle amount prediction result

Day	Time Stamp	Prediction of Vehicle Pass (Deep Learning)			
		Direction 1	Direction 2	Direction 3	Total
Mon	7:00:00	5	4	8	17
Mon	7:30:00	7	6	4	17
Mon	8:00:00	9	6	3	18
Mon	8:30:00	8	4	5	17
Mon	9:00:00	3	8	7	18
Mon	9:30:00	3	7	7	17
Mon	10:00:00	6	3	8	17

5. **Conclusion.** This paper presents an implementation of computer vision and deep learning to create a smart traffic light model that can dynamically adjust the lights duration of each direction in an intersection based on the number of vehicles. By using this method, the traffic light model can predict the vehicle amounts in 30 minutes with an accuracy rate of 75.4%, reducing the traffic congestion rate by 17.86% preemptively, while reducing the current traffic congestion by 28.57%. This study focuses on predicting and reducing the number of vehicles remaining in the traffic light. However, a real-world application would require a consideration of the differing weather conditions and vehicle passing rate in different intersections, as well as the road conditions and number of usable lanes available. For future works, this traffic light model can be improved by adding the amount of considered variables in both prediction and duration adjustments. Furthermore, a full-sized model can also be made to scale with real-world conditions, in order to increase the prediction accuracy.

REFERENCES

[1] *Traffic Light*, Cambridge, https://dictionary.cambridge.org/dictionary/english/traffic-light, Accessed on 7 March, 2020.

[2] J. Schmidhuber, Deep learning in neural networks: An overview, *Neural Networks*, vol.61, pp.85-117, 2015.

[3] D. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*, Prentice Hall, Upper Saddle River, N.J., London, 2003.

[4] M. Wu and L. Chen, Image recognition based on deep learning, *2015 Chinese Automation Congress (CAC)*, Wuhan, China, 2015.

[5] Y. LeCun, Y. Bengio and G. Hinton, Deep learning, *Nature*, vol.521, no.7553, pp.436-444, 2015.

[6] I. Goodfellow, Y. Bengio and A. Courville, *Deep Learning*, The MIT Press, Cambridge, Massachusetts, 2016.

[7] S. Mezgec and B. K. Seljak, NutriNet: A deep learning food and drink image recognition system for dietary assessment, *Nutrients*, vol.9, no.7, p.657, doi: 10.3390/nu9070657, 2017.

[8] Z. Zhao, W. Chen, X. Wu, P. C. Y. Chen and J. Liu, LSTM network: A deep learning approach for short-term traffic forecast, *IET Intelligent Transport Systems*, vol.11, no.2, pp.68-75, 2017.

[9] L. Zhang and W. Lin, *Selective Visual Attention: Computational Models and Applications*, John Wiley & Sons Inc., Singapore, 2013.

[10] K. Shi, H. Bao and N. Ma, Forward vehicle detection based on incremental learning and fast R-CNN, *2017 the 13th International Conference on Computational Intelligence and Security (CIS)*, doi: 10.1109/cis.2017.00024, 2017.

[11] K. Azhar, F. Murtaza, M. H. Yousaf and H. A. Habib, Computer vision based detection and localization of potholes in asphalt pavement images, *2016 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, Vancouver, BC, pp.1-5, 2016.
[12] N. Seenouvong, U. Watchareeruetai, C. Nuthong, K. Khongsomboon and N. Ohnishi, A computer vision based vehicle detection and counting system, *2016 the 8th International Conference on Knowledge and Smart Technology (KST)*, 2016.
[13] M. Margolis, *Arduino Cookbook: Recipes to Begin, Expand, and Enhance Your Projects*, O'Reily Media, Inc., 2011.
[14] Y. A. Badamasi, The working principle of an Arduino, *2014 the 11th International Conference on Electronics, Computer and Computation (ICECCO)*, Abuja, pp.1-4, 2014.
[15] M. Banzi, *Getting Started with Arduino*, O'Reilly Media, Inc., 2009.
[16] S. El-tantawy, B. Abdulhai and H. Abdelgawad, Design of reinforcement learning parameters for seamless application of adaptive traffic signal control, *Journal of Intelligent Transportation Systems*, vol.18, no.3, pp.227-245, doi: 10.1080/15472450.2013.810991, 2014.
[17] M. H. Malhi, M. H. Aslam, F. Saeed, O. Javed and M. Fraz, Vision based intelligent traffic management system, *2011 Frontiers of Information Technology*, pp.137-141, doi: 10.1109/FIT.2011.33, 2011.
[18] B. Ghazal, K. Elkhatib, K. Chahine and M. Kherfan, Smart traffic light control system, *2016 the 3rd International Conference on Electrical, Electronics, Computer Engineering and Their Applications (EECEA)*, pp.140-145, 2016.
[19] OpenCV, *Open Source Computer Vision Library OpenCV*, http://sourceforge.net/projects/opencvlibrary, Accessed on 10 March, 2020.