

Development of Smart Traffic Light Controller System with Deep Learning Capability in Image Processing

Siu Hong Loh^{1*}, Jia Jia Sim², Chu Shen Ong¹, Kim Ho Yeap¹, Peh Chiong Teh¹ and Kim Hoe Tshai¹

¹Faculty of Engineering and Green Technology, Universiti Tunku Abdul Rahman, 31900 Kampar, Perak, Malaysia

²Faculty of Business and Finance, Universiti Tunku Abdul Rahman, 31900 Kampar, Perak, Malaysia

*Corresponding author: lohsh@utar.edu.my

Submitted 31 January 2021, Revised 17 March 2021, Accepted 22 March 2021.

Copyright © 2021 The Authors.

Abstract: The traffic congestion at the junction is becoming one of the major issues for many cities all around the world. One of the reasons causing this issue is due to the inefficient of the existing traffic light system at the traffic junction. This paper proposes a Smart Traffic Light Controller System (STLCS) with deep learning capability in image processing. The developed STLCS is comprised of Altera DE2 board, personal computer and Intel Neural Compute Stick 2 (NCS2). The personal computer is used as the vehicle detection system of the STLCS by performing various computer vision tasks and inference. The tasks include image acquisition, processing, and vehicle detection and counting. The smart feature of the system can detect the vehicles by using deep learning model and compute a flexible green time for each lane according to the density of traffic in each lane. The vehicle detection emphasizes the image processing by using the deep learning algorithm from the pre-trained model to increase the efficiency and computing time of the system. The efficiency of the vehicle detection system is about 94.73%.

Keywords: Deep learning; Image processing; Inference engine; Traffic congestion; Vehicle detection.

1. INTRODUCTION

In this era of globalization, the vehicle is one of the popular ways for Malaysians to travel to another place. According to the statistics provided by the Malaysian Automotive Association (MAA), for the past ten years, there is an average of six hundred thousand vehicles registered every year [1]. Traffic congestion happens in the urban area when the overwhelming number of the vehicle overloads the street capacity. Several factors are causing traffic congestion; the rapid growth of the population in the urban area, the rapid increase in the number of vehicles, and the ineffective traffic light system. In order to solve traffic congestion, one of the ways is developing a Smart Traffic Light Controller System (STLCS), which aims to provide smooth motion of vehicles in the routes.

The design of STLCS is built up by various systems, such as image capturing systems, adaptive control systems and communication systems. Those systems are needed to create an adaptive traffic light that can adjust the time slots for each of the routes to achieve the highest efficiency. Several researchers have worked on STLCS previously using various approaches. Traffic light control system using microcontroller and infrared transmitter and receiver as the traffic sensing device were quite common [2-3]. In this system, the IR transmitters and receivers were placed on either side of the road. Whenever any object passes the IR transmitter and receiver, the sensors were triggered and send the signals to the microcontroller. The microcontroller can make a comprehensive timing decision based on the number of the object passes the sensors. Several groups of researchers have developed image processing-based system for the traffic light controller [4-6]. In this type of system, Arduino controller board, cameras and image processing toolkit in MATLAB are used. Various techniques in image processing were used to extract useful data from the images. From the perspective of controller, there were groups of researchers preferred Field Programmable Gate Array (FPGA) over conventional microcontroller [7-10]. FPGA is advantageous in terms of coding flexibility and performance compared to microcontroller. In these FPGA-based systems, four-way traffic junction was used to model the smart traffic light controller system. The sensors were placed at every lane to act as the vehicle detection system. The system was able to perform vehicle detection and real-time control of the traffic by using the Finite State Machine (FSM) approach. Researchers also utilized inductive loop structure in their proposed traffic control system [11]. In this design, the detector can provide information on vehicle type, count, length and speed.

As compared to previous systems, the STLCS presented in this paper enhances the performance of the system in two major aspects, namely the image processing task and the main controller system. The image processing-based traffic light

systems [4-6] used MATLAB to process images. Recently, Intel Neural Compute Stick 2 (NCS2) were widely used for various visual applications, such as predicting traffic density [12], real-time movement detection system [13] and autonomous vehicles computing platform [14]. By integrating Intel NCS2 into the developed STLCS, the image processing tasks are highly accelerated as compared to usage of MATLAB alone. From the aspect of the main controller, using FPGA-based controller further increase the response of the entire system. This is due to the use of FSM approach that is flexible enough to alter the level of control states, which is not possible in conventional microcontroller-based system. Previous FPGA-based systems [7-10] focus mainly on the algorithm, which lacks the adaptive capability in response to various traffic conditions. The STLCS presented in this paper combined the hardware-accelerated image processing capability with FPGA-based flexibility so that the overall system is relatively more adaptive and responsive than the previous systems.

2. METHODOLOGY

In this project, the developed STLCS consists of two major sections, namely the vehicle detection system and the traffic light controller system, as shown in Figure 1. Firstly, a high-resolution webcam is used to capture the image of traffic lanes at the four-way junction. Next, an inference engine in OpenVINO Toolkit was used to process the images. Then the number of vehicles detected is counted, and a series of dedicated timing algorithm was used to calculate the time for green time to ensure maximum flow across the junction. The results of the timing calculation are then sent via UART serial communication to Altera DE2 board which act as the traffic light controller. Altera DE2 board controls the green light according to the received time value. The yellow light will be turned on, followed by the green light, and a signal will be sent by the Altera DE2 board to the PC to obtain the value of the next green light. This process is repeated to provide continuous traffic control to maximize traffic flow.

2.1 Image Acquisition

Image acquisition is performed by using Logitech C310 HD Webcam to capture the real-time image of the four-way traffic intersection. It is a very crucial step to ensure the image can be properly captured and processed. There are few precautionary steps that need to be taken to get the best possible image. Firstly, the camera must be placed at a place with wide vision angle to cover the whole view of traffic junction. Since the camera is a fixed focus camera and lacks the ability to autofocus on the object, the distance of the camera is very important to ensure a clear image can be obtained. This will prevent unnecessary noise that is very difficult to be eliminated in the later stage. Therefore, many trial and error attempts are required to find the best position for the installation of the camera.

2.2 Inference Engine

The inference engine is a set of libraries to provide the common Application Programming Interface (API) to deliver the inference solutions on any platform, such as Graphics Processing Unit (GPU), Central Processing Unit (CPU), Vision Processing Unit (VPU) and FPGA. In this project, the Intel NCS2 is used as the platform for the inference application. Intel Movidius VPUs are included in the Intel NCS2 which enables the acceleration of the computer vision and edge AI workloads. In the system, the inference engine is used to perform deep learning on vehicle detection by using the pre-trained model.

Inferencing is a stage to use a trained model to predict the testing samples and compromise a similar forward pass as the training of the model to give the predicted value. Inference does not include the backward pass to compute the error and update the weight. Inferencing is the deployment stage of the inference engine to predict real-world data. In the developed system, some preparations must be made before performing inference in the vehicle detection system. Firstly, the Intermediate Representation (IR) files (.xml and .bin files) must be loaded into the system, and the target device of the system has to be specified. The target device in the proposed system is the Intel Neural Compute Stick 2. Besides, the input image must be resized to 672 x 386 pixels according to the pre-trained model. Then, the image can be read into the input blob to perform inferencing in the OpenVINO inference engine. The results of the inferencing are obtained from the output blob of the inference engine. The output blob of the inference engine contains several parameters such as image ID, label, the confidence of the prediction, and the coordinates of the bounding box. These data can be used in the vehicle detection system to count the number of vehicles detected and to draw the bounding box in the output image. After the inference stage, the vehicles of the foreground objects can be clearly determined and detected. Figure 2 depicts the application of inference engine in the proposed STLCS.

The pre-trained model selected for the proposed system is one of the pre-trained models included in the OpenVINO toolkit. The model used in the proposed system is vehicle-detection-adas-0002, which is one of the vehicle detection networks based on a single-shot detector (SSD) framework. This model is selected because it has a very good average precision (AP) which is about 90.6%. Besides that, the model is capable of detecting a maximum of 200 objects. The model was trained with 3000 images and a total of 12585 vehicles according to the documentation.

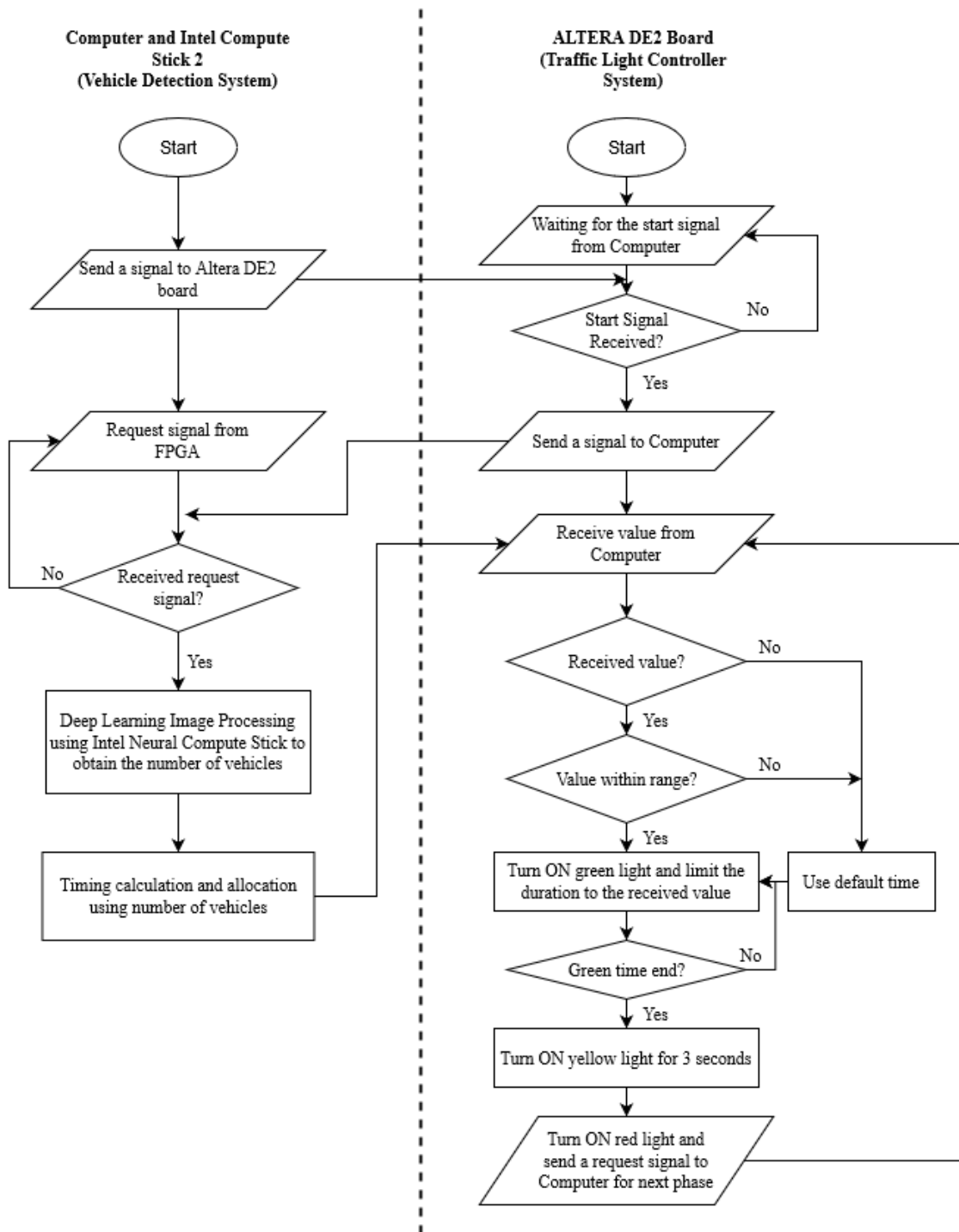


Figure 1. Flowchart of the developed STLCS

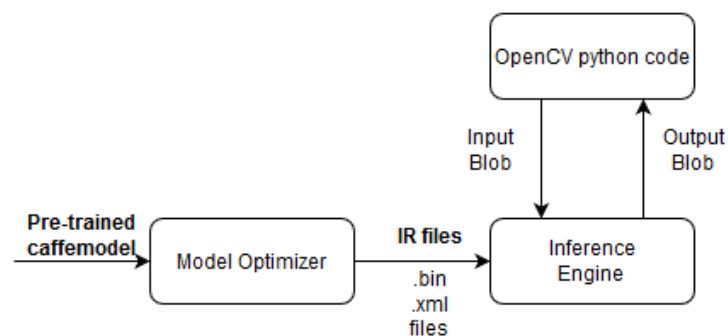


Figure 2. Application of inference engine in the proposed STLCS

2.3 Timing Algorithms for Traffic Light Signal

In order to create an effective and efficient traffic light controller, the developed STLCS comes with a timing algorithm to minimize the queuing time on each lane at the traffic junction to maximize the traffic flow of junction. The time needed for the vehicle to cross the stop line from its initial position is called headway. According to the theory [15], the first headway is relatively higher than the subsequent headway because it includes longer start-up loss time, e_n (reaction time of the driver + the acceleration of the vehicle). The second headway is relatively much shorter than the first headway because the start-up loss time is overlapping with the first headway. Additionally, the time needed to perceive the change of the signal is overlapping with the first headway. After a few vehicles, the headway tends to be saturated and kept on a constant value because the driver queuing behind have more time to accelerate to high speed. Although the differences between headways after fifth or more headways are almost negligible, and it gives about 2.5 s/vehicle for a saturated headway. To ensure that the algorithm is more practical and efficient, the start-up loss time S_{loss} must be added to the algorithm. Because of the start-up loss time, e_n is showing an exponential decay slope, therefore, the total start-up loss time, S_{loss} can be calculated by using the equation below:

$$S_{loss} = \sum_{i=1}^n sloss[f^{i-1}] \quad (1)$$

where S_{loss} is the total start-up loss time, $sloss$ is the standard start-up loss time (assume 2.5 s), f is the decrease factor (assume 0.75) and n is the number of detected vehicles. The green time required for each lane can be estimated as:

$$T = S_{loss} + (h * n) \quad (2)$$

where T is the required green time and h is the saturation headway (assume 1.2 s/vehicle). The minimum and maximum green time are fixed to 5 seconds and 25 seconds respectively to ensure that the time computed is not too short nor too long, and to act as the precaution measure for vehicle detection error.

2.4 Serial Communication on Altera DE2 Board

The UART serial communication system is implemented in Altera DE2 board, and the system is divided into three parts which are the receiver module, the transmitter module and the baud-rate generator. Altera DE2 board is equipped with the RS-232 port, which allows the realization of the serial communication on board. The UART serial communication consists of three sub-modules. First, the baud rate generator is used to generate a clock signal to determine the rate of the data being transmitted and synchronize the data transmission and reception. Then, the receiver module will receive the data and convert the data into parallel data for the data processing. Lastly, the transmitter converts the parallel data into serial data with the parity bit and transmits the serial data.

2.5 Traffic Light Controller System on Altera DE2 Board

The developed STLCS is targeted to the common four-way traffic junction which consists of north, east, south, and west directions, shown in Figure 3. Each of the directions is equipped with the traffic light with a set of four light: one red light, one yellow, and two green lights that indicate the straight and right turn discharges. In order to achieve the maximum flow of vehicles, the phases of the traffic light controller must be determined. In this system, there are four phases for the traffic light to work properly. Each of the phases will be related to one lane. The different phases of the traffic light controller are shown in the Figure 4. FSM approach is utilised for the controller system. Table 1 shows the state table of the controller system.

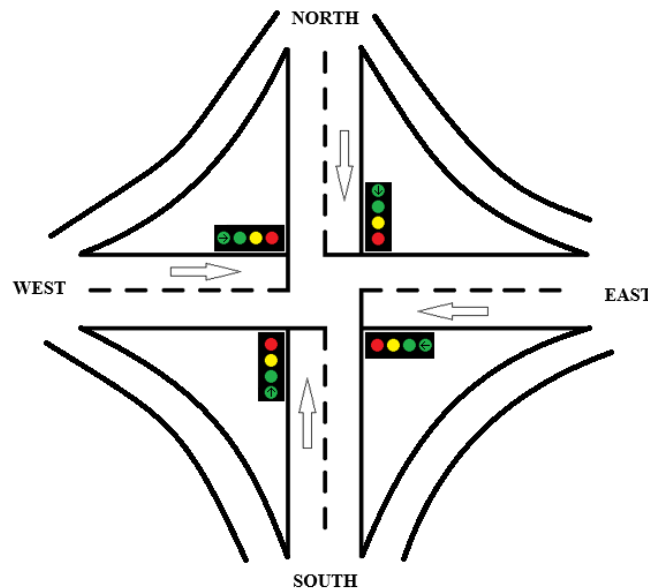


Figure 3. Four-way traffic junction

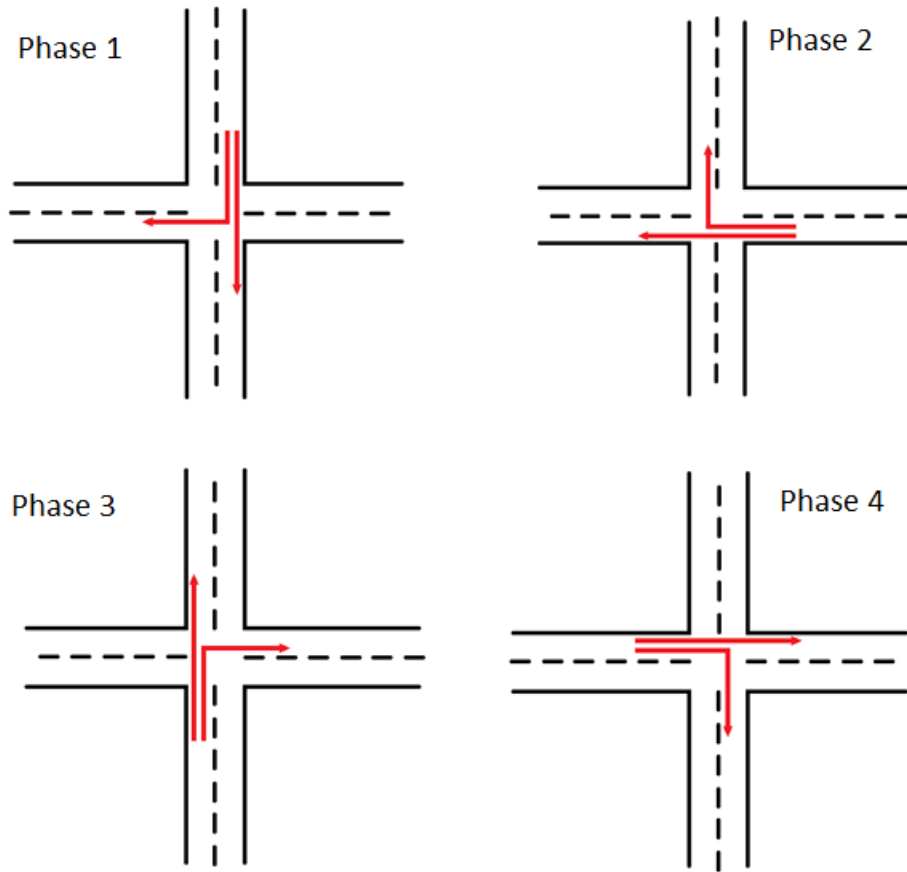


Figure 4. Four phases of traffic light signals

Table 1. State table of the traffic light controller system

States				Direction of lanes			
Current State	Store State	Next Stage		North	East	South	West
		Clr = 0	Clr = 1				
Idle(D)	A	I	Idle	Red	Yellow	Red	Red
A	C	B	Idle	Green	Red	Red	Red
B		I	Idle	Yellow	Red	Red	Red
C	E	D	Idle	Red	Green	Red	Red
D		I	Idle	Red	Yellow	Red	Red
E	G	F	Idle	Red	Red	Green	Red
F		I	Idle	Red	Red	Yellow	Red
G	A	H	Idle	Red	Red	Red	Green
H		I	Idle	Red	Red	Red	Yellow
I		Stored State	Idle	Red	Red	Red	Red

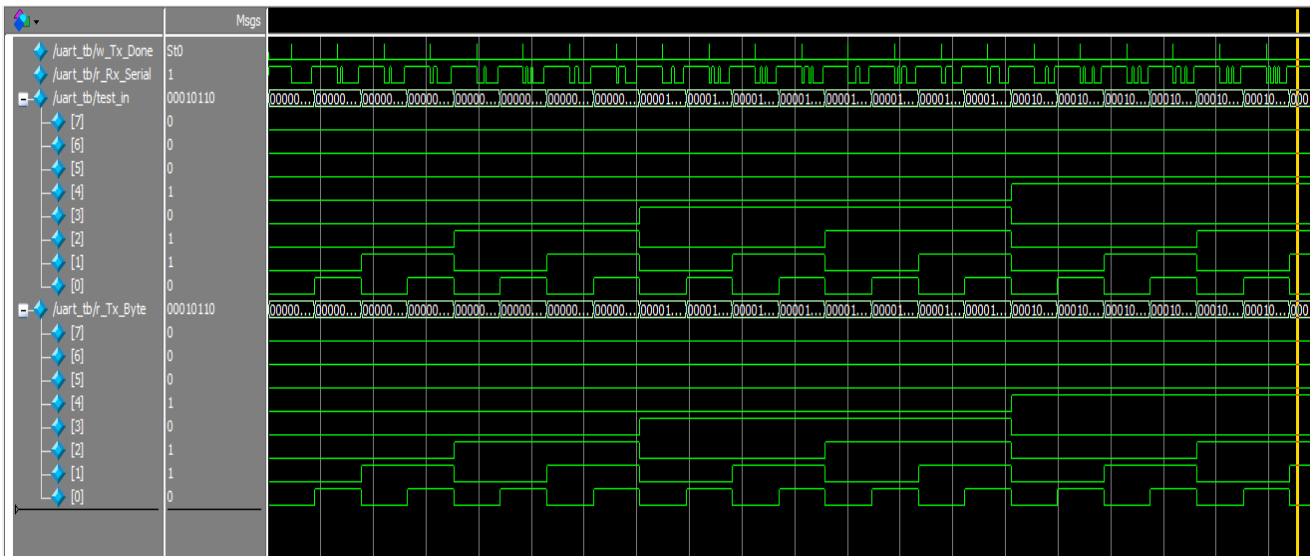


Figure 5. Sample of output waveform for the UART serial communication module

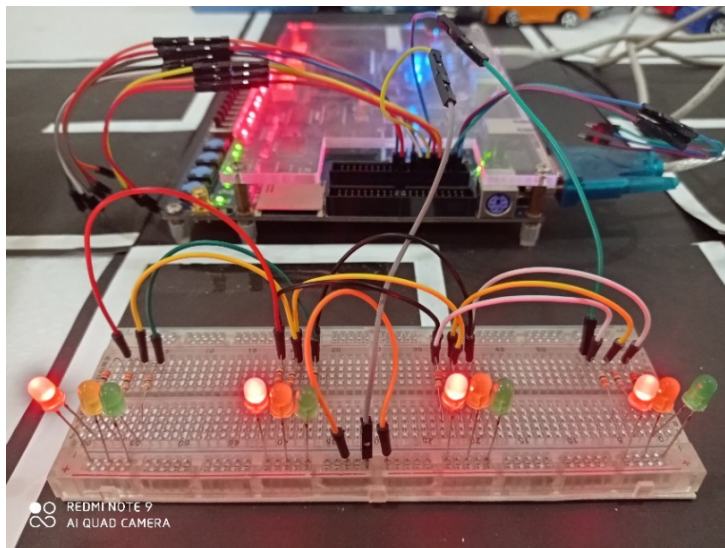


Figure 6. LED circuit to represent the traffic light system

3. RESULTS AND DISCUSSION

In the developed STLCS, several preliminary tests were conducted before implementing the full system test. First, the test benches were written to simulate the UART serial communication section. The modules were tested by giving specific inputs. The sample of output waveform for the module is shown in Figure 5.

Next, the traffic light circuit was tested by connecting the anodes of the LEDs to the built-in 3.3 V GPIO port on Altera DE2 board, and 330 Ω resistors were connected in series to the LEDs to limit the current flow. The cathodes of the LEDs were connected to the ground GPIO port of Altera DE2 board to form a complete circuit. The circuit was checked to ensure that LEDs were illuminated as required. Figure 6 depicts the LED circuit to represent the traffic light system. The computer and Altera DE2 board were connected by using the RS232 to USB converter cable. The UART modules were programmed into both devices, and the required driver was installed on the computer. The data transferred from both devices were checked and observed to ensure that the data reached both devices properly. Table 2 shows the results of the testing by both devices.

The prototype model of the STLCS is shown in Figure 7. The prototype was made of polystyrene and toy cars were used to imitate the traffic light junction in the real world. Every module implemented in the software and hardware were integrated for the complete functional tests. Few experiments were conducted to examine the effectiveness of the system in terms of the accuracy of the deep learning model. Table 3 shows the result of a single functional test. Different combinations and arrangements of the colour and number of vehicles were assigned to four lanes. The actual vehicle number indicates the actual vehicles assigned to the lane, and the detected vehicle number indicates the number of vehicles detected by the deep learning model and computer vision. The accuracy represents the percentage error between the actual and detected number of vehicles.

Table 2. Data transferred between the computer and Altera DE2 board


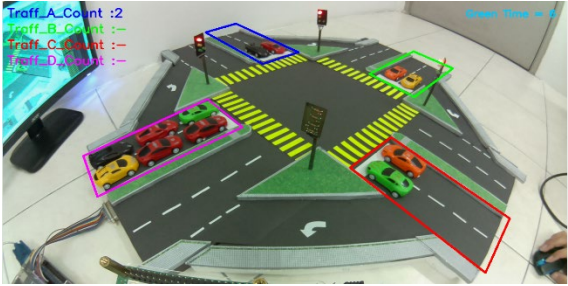
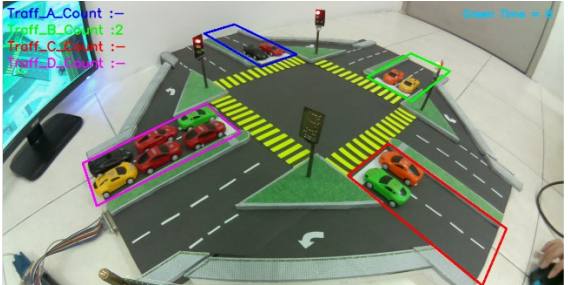

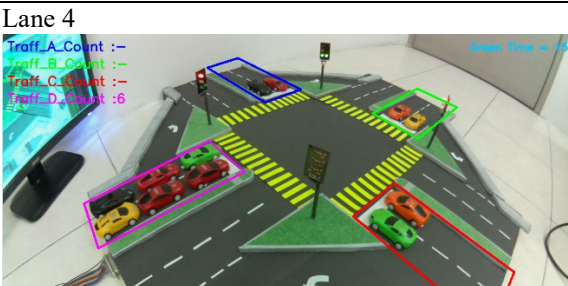
Computer Command Prompt	Altera DE2 board
Received number: 1 Transmitted number: 13	
Computer: Received data is 1, and transmitted data is 13. Altera DE2 board: Received data is 13, and transmitted data is 1.	
Received number: 2 Transmitted number: 3	
Computer: Received data is 2, and transmitted data is 3. Altera DE2 board: Received data is 3, and transmitted data is 2.	
Received number: 3 Transmitted number: 6	
Computer: Received data is 3, and transmitted data is 6. Altera DE2 board: Received data is 6, and transmitted data is 3.	
Received number: 4 Transmitted number: 9	
Computer: Received data is 4, and transmitted data is 9. Altera DE2 board: Received data is 9, and transmitted data is 4.	



Figure 7. Prototype model of the STLCS

In total, nine sets of functional tests were conducted, and the results are shown in Table 4. It is observed that the miscounting often happens in lane 1 and lane 2 (blue and green square boxes in the images in Table 3). The issue is most likely caused by the limited resolution of the input images and the image processing algorithm. The algorithm is unable to detect the vehicle well when the vehicle is too small in the images. Besides, the colour of the car is also affecting car detection. In the testing process, if there are vehicles with similar properties of color are positioned closely to each other, the binary objects tend to be combined together, possibly causing the segmentation algorithms leading the system to count the vehicles more than the actual number (Test 2 and Test 4). In addition, if two vehicles with different properties of color and located in series (far away from the camera), some portion of the vehicle behind is blocked by the vehicle in front. If the area of detection is too small, the system will ignore it as an object in the counting process. In general, the developed system can generate an acceptable result in certain tests. The overall efficiency is 94.74%, demonstrating the effectiveness and the high accuracy of the system.

Table 3. Result of a single functional test

Sample	Number of vehicles		Accuracy (%)
	Actual	Detected	
Lane 1 	2	2	100.00
Lane 2 	2	2	100.00
Lane 3 	2	2	100.00
Lane 4 	6	6	100.00

4. CONCLUSION

In this paper, a comprehensive STLCS with machine learning capability is presented. The developed system aims to improve the efficiency of the existing traffic light controller system by assigning the best queuing time for each lane to achieve the maximum traffic flow. The development of this system is divided into three major parts which are vehicle detection system, UART serial communication system and traffic light controller system. The implementation of the vehicle detection system was done by using a computer with Intel CPU, Intel NCS 2 and python programming. The vehicle detection emphasizes the image processing by using the machine learning algorithm to increase the efficiency and computing time of the system. The efficiency of the vehicle detection system is about 94.73%. Next, a full-duplex UART serial communication system was implemented by using FPGA and Verilog HDL to allow the communication between the Altera FPGA DE2 board and vehicle detection. Besides, the traffic light controller system was constructed on the Altera FPGA DE2 board by using a FSM approach to control the external traffic light that was constructed by using resistors and LEDs. The system can communicate with the vehicle detection system to increase the flexibility of the traffic signal control.

Table 4. Summary of functional test results

Test	Lanes	Actual number of vehicles	Detected number of vehicles	Accuracy (%)
1	1	1	1	100.00
	2	1	1	100.00
	3	1	1	100.00
	4	1	1	100.00
2	1	2	2	100.00
	2	4	5	75.00
	3	2	2	100.00
	4	2	2	100.00
3	1	3	3	100.00
	2	3	2	66.67
	3	3	3	100.00
	4	3	3	100.00
4	1	4	3	75.00
	2	4	5	75.00
	3	4	4	100.00
	4	0	0	100.00
5	1	5	5	100.00
	2	3	3	100.00
	3	0	0	100.00
	4	4	4	100.00
6	1	6	5	83.33
	2	5	4	80.00
	3	0	0	100.00
	4	0	0	100.00
7	1	0	0	100.00
	2	6	4	66.67
	3	5	5	100.00
	4	4	4	100.00
8	1	0	0	100.00
	2	0	0	100.00
	3	6	5	83.33
	4	5	5	100.00
9	1	2	2	100.00
	2	2	2	100.00
	3	2	2	100.00
	4	6	6	100.00
Total		95	90	94.73

REFERENCES

- [1] Malaysian Automotive Association, Sales & Production Statistics. <http://www.maa.org.my/statistics.html>, 2021 (accessed 27.01.2021).
- [2] T. E. Somefun, C. O. A. Awosope, A. Abdulkareem, E. Okpon, A. S. Alayande and C. T. Somefun, Design and implementation of density-based traffic management system, *International Journal of Engineering Research and Technology*, 13, 2020, 2157-2164.
- [3] P. Sinhmar, Intelligent traffic light and density control using IR sensors and microcontroller, *International Journal of Advanced Technology & Engineering Research*, 2, 2012, 30-35.
- [4] P. Choudekar, S. Banerjee and M. K. Muju, Implementation of image processing in real time traffic light control, *2011 3rd International Conference on Electronics Computer Technology*, Kanyakumari, India, 2011.
- [5] Kushi, Smart control of traffic light system using image processing, *2017 International Conference on Current Trends in Computer, Electrical, Electronics and Communication (CTCEEC)*, Mysore, India, 2017.
- [6] A. H. M. Almagwani, Design of real time smart traffic light control system, *Proceedings of ISER 108th International Conference*, Mecca, Saudi Arabia, 2018, 51-55.
- [7] A. Osarenomase, F. S. Bala and G. Bakare, Field programmable gate array based intelligent traffic light system, *International Journal of Engineering and Innovative Technology*, 4, 2015, 10-16.

- [8] B. Dilip, Y. Alekhya and P. D. Bharathi, FPGA implementation of an advanced traffic light controller using Verilog HDL, *International Journal of Advanced Research in Computer Engineering & Technology*, 1, 2012, 1-6.
- [9] S. U. Holambe and D. B. Andore, Advance traffic light control system based on FPGA, *International Journal of Scientific Engineering and Research*, 3, 2015, 31-35.
- [10] D. Bhavana, D. R. Tej, P. Jain, G. Mounika and R. Mohini, Traffic light controller using FPGA, *International Journal of Engineering Research and Applications*, 5, 2015, 165-168.
- [11] V. S. Rakesh and V. Shaithya, A traffic control system using inductive loop detector, *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, 4, 2015, 4590-4593.
- [12] R. A. Asmara, B. Syahputro, D. supriyanto and A. N. Handayani, Prediction of traffic density using YOLO object detection and implemented in Raspberry Pi 3b + and Intel NCS 2, *2020 IEEE 4th International Conference on Vocational Education and Training (ICOVET)*, Malang, Indonesia, 2020.
- [13] T. Tsai, C. Hsu and W. Wan, Live demonstration: Vision-based real-time fall detection system on embedded system, *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, Sevilla, Spain, 2020.
- [14] E. Z. Q. Koh, A. Y. Abdalla and H. Nugroho, Visual computing-based perception system for small autonomous vehicles: Development on a lighter computing platform, *2020 IEEE Student Conference on Research and Development (SCORED)*, Johor, Malaysia, 2020.
- [15] L. A. Elefteriadou, *Highway Capacity Manual, 6th Edition: A Guide for Multimodal Mobility Analysis*. Washington, DC: The National Academies Press, 2016.