AN INTELLIGENT DENSITY-BASED TRAFFIC SYSTEM WITH LIBYAN VEHICLES LICENSE PLATE RECOGNITION ON FPGA

Lamia Atif Ali and Mohamed Muftah Eljhani

Department of Computer Engineering, Faculty of Engineering, University of Tripoli

Email: M.Eljhani@uot.edu.ly

Received 6 June 2023; revised 2 August 2023; accepted 20 August 2023

الملخص

الهدف الرئيسي من هذا البحث هو تطوير واختبار وتطبيق نظام إشارات المرور في الوقت الحقيقي بناءً على كثافة حركة المرور، بما في ذلك الكشف عن مخالفات الضوء الأحمر والتعرف التلقائي على لوحة الترخيص الليبية. لتقدير كثافة حركة المرور، يتم استخدام مجموعة من أجهزة استشعار الأشعة تحت الحمراء المثبتة على كل الطرق المتقاطعة. فهو يعطي الأولوية لمركبات الطوارئ التي تستخدم تقنية تحديد الترددات الراديوية. وباستخدام أجهزة استشعار الأشعة تحت الحمراء، يقوم النظام بالكشف عن مخالفات الضوء الأحمر وتسجيل صورة السيارة المخالفة، والتي يتم تمريرها بعد ذلك إلى نظام التعرف الآلي على لوحة الترخيص، والذي يقوم بتحليل الصورة لتحديد منطقة لوحة الترخيص والتعرف على الأرقام والحروف. تم إنشاء نظام إشارات المرور المقترح ومحاكاته واختباره على لوحة Cyclone IV GX باستخدام لغة وصف أجهزة Verilog.

ABSTRACT

The main objective of this research is to develop, test, and deploy a real-time traffic light system based on traffic density, including red light violation detection and automatic Libyan license plate identification. To estimate traffic density, a set of infrared sensors mounted on each intersecting roadway is utilized. It prioritizes emergency vehicles using radio frequency identification technology. Using passive infrared sensors, the system detects red light violations and records the violating vehicle's picture, which is then passed to the automated license plate recognition system, which analyzes the image to localize the license plate region and recognize the numbers and characters. The proposed traffic light system is built, simulated, and tested on Cyclone IV GX field-programmable gate array using Verilog Hardware Description Language.

KEYWORDS: Density-Based Traffic Light System; Automated License Plate Recognition; Digital Image Processing; Machine Learning; Verilog Hardware Description Language; Field-Programmable Gate Array Design.

INTRODUCTION

Due to the ever-increasing number of cars, vehicle traffic is now a significant economic component in both urban and rural regions, and it requires suitable administration and surveillance to ensure that this enormous number of vehicles does not conflict with one another [1]. Because of the large number of cars on the road worldwide, various levels of traffic rule-breaking occur, mostly red-light offenses [1]. To apprehend offenders and overcome the faults and failings of human traffic supervisors who are unable to be everywhere at once, effective traffic violation detection and number plate recognition methods are necessary [1]. Furthermore, the massive rise in vehicle numbers might easily result in traffic congestion. A traffic congestion occurs when a vehicle is still for a significant amount of time [2]. Traditional traffic signals have a predetermined time period independent of traffic congestion and do not take emergency vehicles into account. This inspired us to think about finding a better and more effective solution to those difficulties. Intelligent traffic light controllers (TLCs) based on field-programmable gate arrays (FPGAs) are being investigated as a means of designing more efficient traffic systems. In contrast, traditional traffic light controllers employ a 24-hour delay and assign distinct traffic signal intervals to various time zones [3]. There is a way of prioritizing routes in which the waiting time for each is manually modified [4]. Four routes have been developed, and traffic flow on them has been monitored by a 24-hour TLC [5]. In [6] TLC based on real-time density handles junctions with varying delays for routes in the intersection and includes an emergency management capability [7]. A dynamic TLC, in which traffic density is computed based on the number of active infrared (IR) sensors, has been proposed [8]. To recognize the violating vehicle license plate number, a red light violation detection system employs infrared sensors and image processing algorithms [1]. Image processing and an artificial neural network are used to develop a system for recognizing Egypt's license plate [9]. A support vector machine algorithm is used to implement a Libyan license plate recognition system [10]. A license plate recognition system that uses a hierarchy of four Region-Based Convolutional Neural Networks (RCNNs) one to detect the vehicle, another one to detect the license plate region, a smaller RCNN to detect individual letters from the detected license plate, and finally, an RCNN classified is employed to recognize the individual letter [11]. The suggested system uses radio frequency identification (RFID) technology and infrared sensors to control and supervise traffic, along with machine learning and digital image processing are employed to detect the license plate number of the infringing vehicle. FPGA, as a system on a chip (SOPC), regulates and controls traffic based on traffic density and automatically prioritizes emergency vehicles. A novel feature proposed in our system is that the traffic signal shifts based on data received from sensors mounted on the roadways to prevent blocking at the junction. Another important feature in our system is red light violation detection through infrared sensors and license plate identification. License plate identification is accomplished in our system utilizing machine learning and image processing techniques. The key benefit of this license plate recognition system over previous license plate recognition systems is that it considers the image's lighting circumstances, as well as the angles and distances from which the image was taken.

METHODOLOGY

Traffic Light System

System overview

This system is in charge of controlling traffic flow at a three-way junction. Five infrared (IR) sensors are deployed on each route to monitor the quantity of cars. Each road also has three RFID readers, which utilize radio waves to detect and identify emergency vehicles by connecting with the RFID tags attached to them. To avoid interference that might impair the system's performance, proper configuration and evaluation of the frequency spectrums used by RFIDs coexisting in the same environment are critical. Figure (1) shows an overview of the system structure.

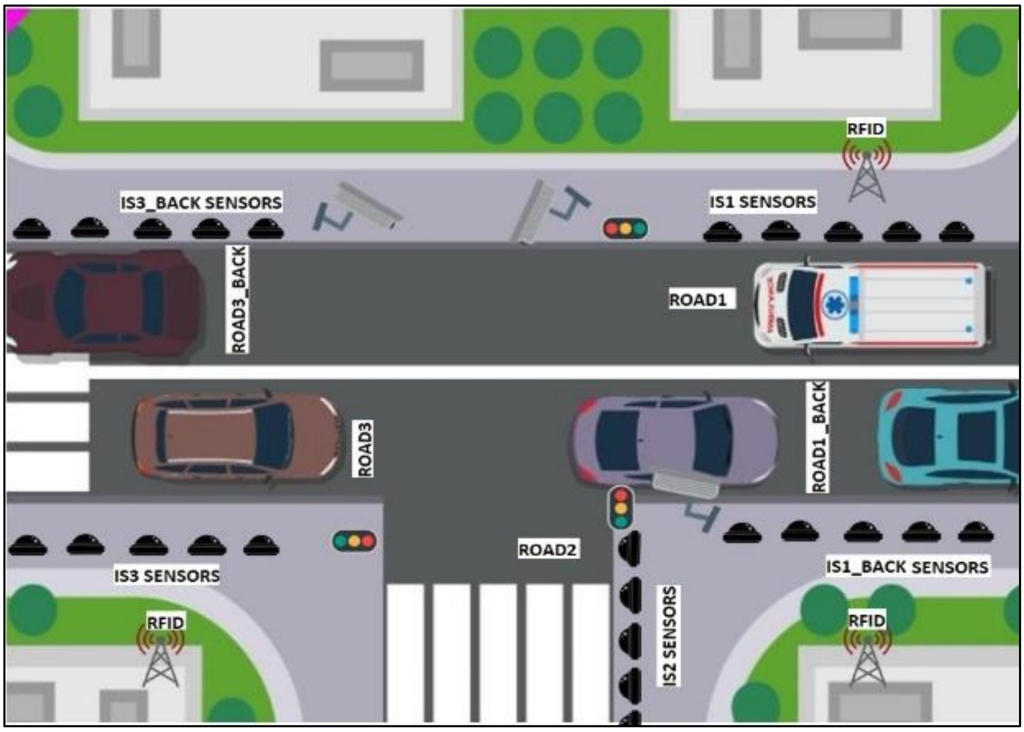

Figure 1: The system's structure.

System algorithm

The algorithm operates by allocating two states, green and yellow, to each road, as well as a shared state in which all roads are red. Each state is illustrated in Table (1).

Table 1: The system states

States	Meaning
R1G	ROAD1's traffic light is green.
R1Y	ROAD1's traffic light is yellow.
R2G	ROAD2's traffic light is green.
R2Y	ROAD2's traffic light is yellow.
R3G	ROAD3's traffic light is green.
R3Y	ROAD3's traffic light is yellow.
ALLRED	The traffic lights are red for All roads.

The system also features three separate state transition flows dependent on the traffic volume of each route and the emergency lights. A three-way junction is managed by the traffic system. It always begins by switching on the yellow light for ROAD1 for predetermined time period, meanwhile looks for any emergency or road blockages signals. If no emergency or road blockage signals are detected, it decides the length of the green light on ROAD2 by constantly watching the active sensor count on ROAD2, which reflects the traffic density level. After the yellow light time period is over, the green light for ROAD2 is switched on, and the system flow keeps on going. Figure (2) illustrates the normal state transition in the absence of an emergency or traffic blockage signal.

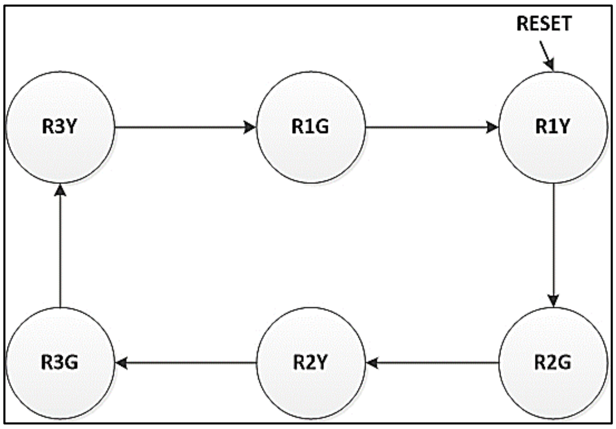

Figure 2: The normal flow of the system's states.

If there is an emergency vehicle on any road the traffic system gives priority to that road to enable the emergency vehicle to pass. Figure (3) shows the state's transition in case of an emergency signal on ROAD3.

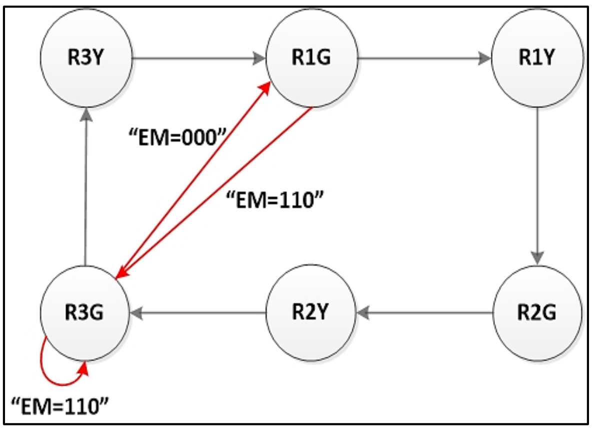

Figure 3: The system's states flow with an emergency vehicle on ROAD3.

When a road blockage signal is received, the system goes into ALLRED state. The road blockage signal depends on the number of active IR sensors on the two main roads from which vehicles leave the intersection. If the active sensor count on them is five, the road has been blocked, and the system switches to ALLRED state (where the traffic light for all roadways is red) giving enough time for the blocked road to become less congested. Figure (4) illustrates the state's flow in road blockage case

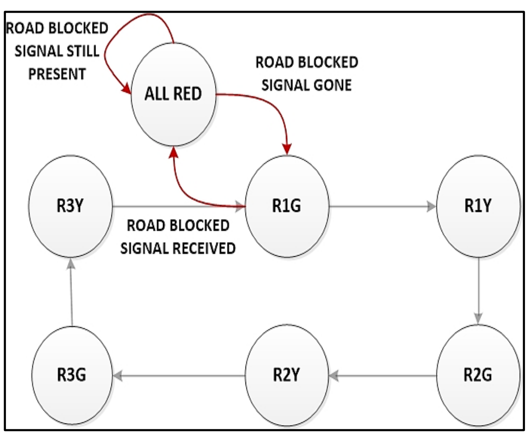

Figure 4: The state transition with road blockage signal.

Figure (5) shows ROAD2's green state flowchart. In ROAD2's green state flow chart illustration of what occurs during this state. The green light is on at ROAD2 and the system checks for emergency vehicles or road blockage and based on that the system's flow changes.

Figure (6) shows the flow chart of ROAD1's yellow state. In ROAD1's yellow state, the green light period for ROAD2 is determined and it takes into account the emergency vehicles and road blockage.

Figure (7) views the ALLRED state flow chart. We illustrated ROAD2's green state and ROAD1's yellow state flow charts only as the remaining roads' green and yellow states are similar to them. Note that if the system goes to an unused state it returns to the initial state.

The traffic lights for the three roads are represented by R1light, R2light, and R3light, where 100, 010, and 001 refer to green, yellow, and red, respectively. The inputs IS1, IS2, IS3, IS1_BACK, and IS3_BACK reflect the IR sensors for ROAD1, ROAD2, ROAD3, ROAD1_BACK, and ROAD3_BACK, with values ranging from 00000 to 11111 reflecting the varied traffic levels on every road.

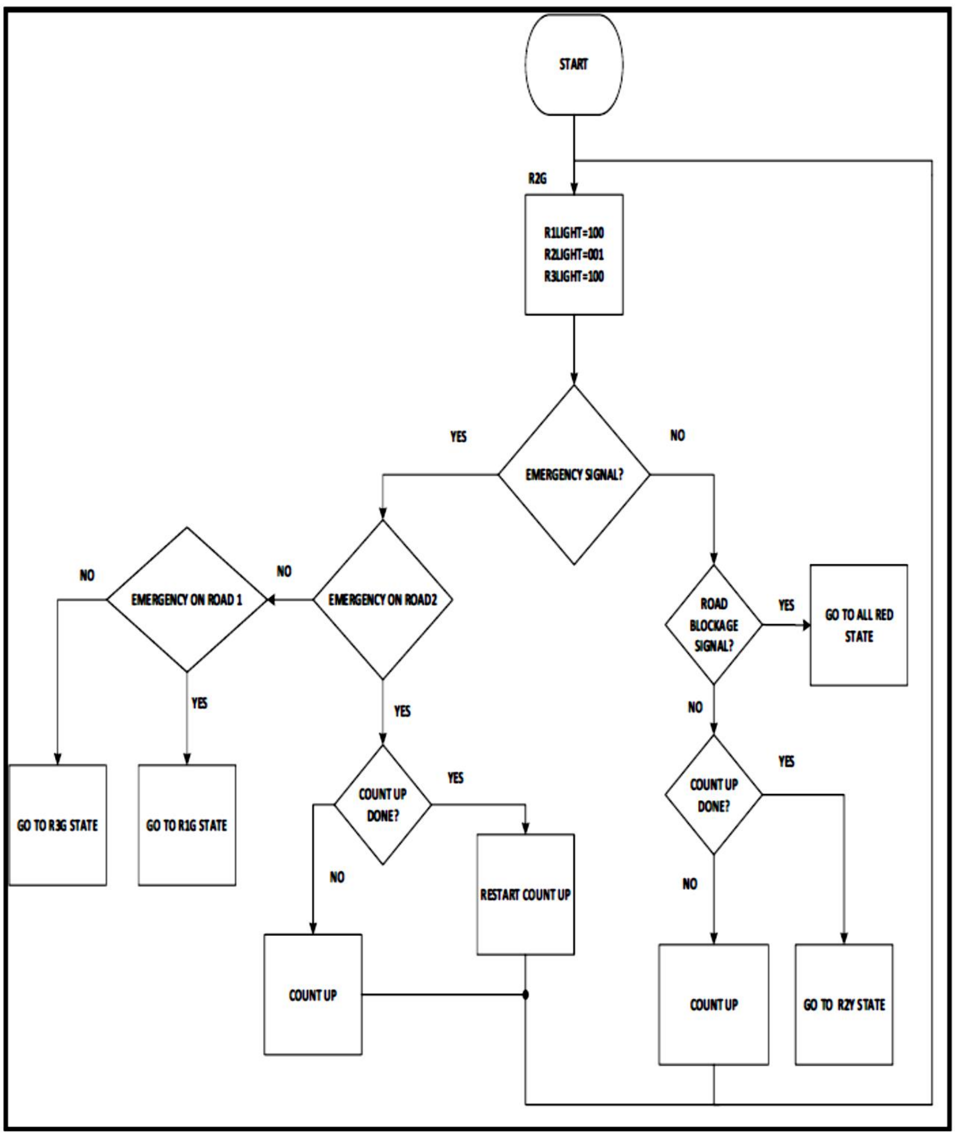

Figure 5: Flow chart of ROAD2 green state code (001).

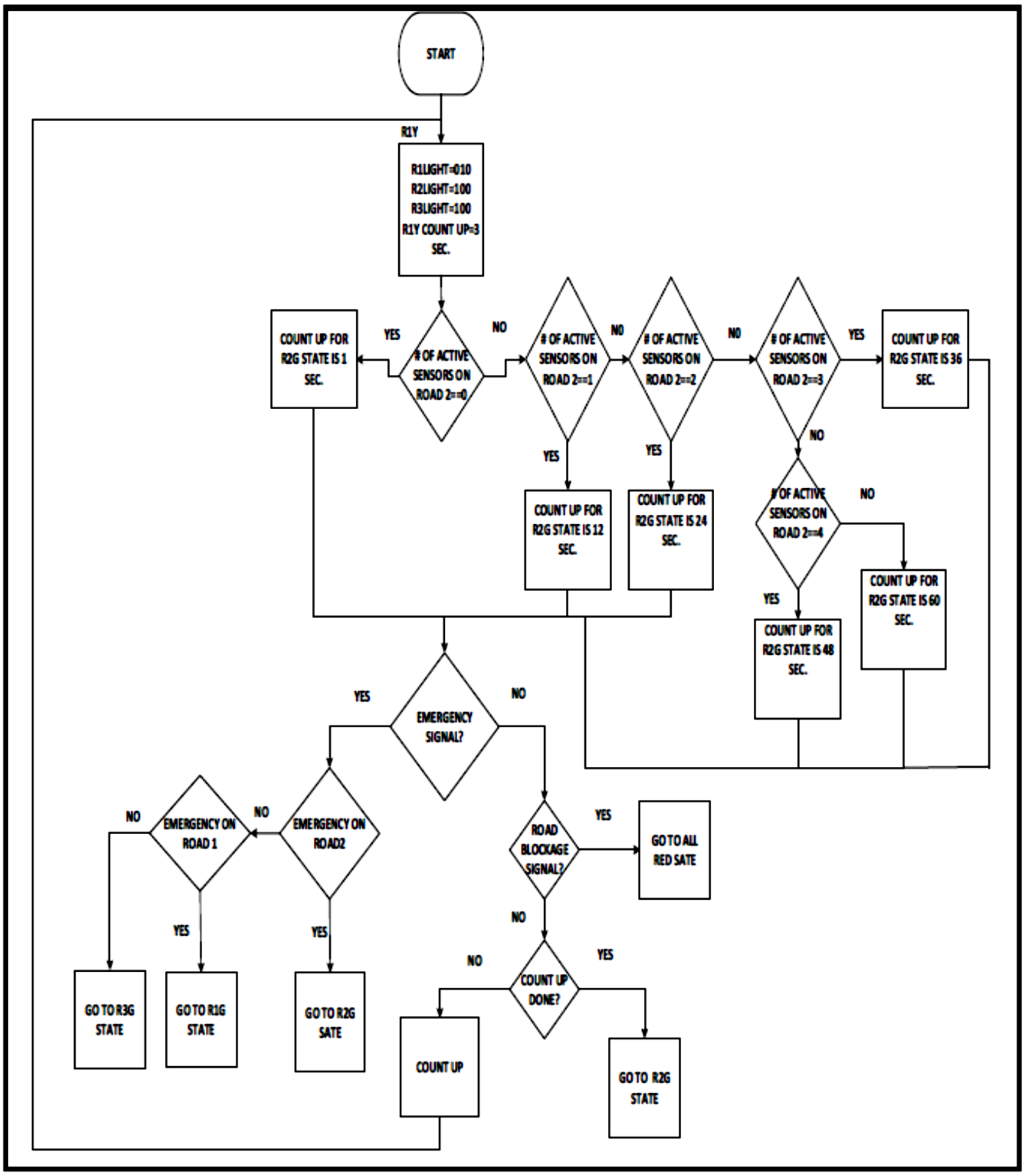

Figure 6: Flow chart of ROAD1 yellow state code (010).

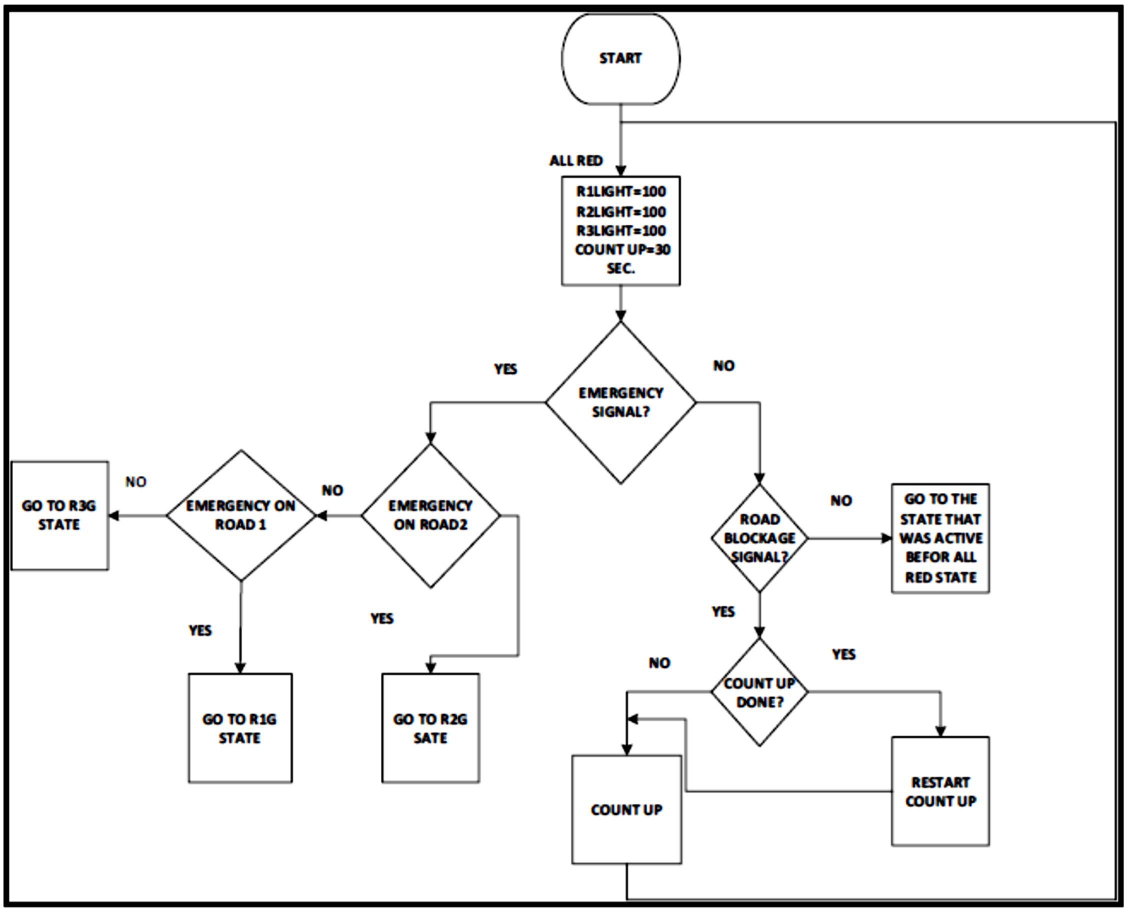

Figure 7: Flow chart of ALLRED state code (100).

Features proposed in the system

Dynamic traffic light period: This intelligent traffic system functions in real-time and schedules traffic lights according to traffic density. Consider that it is ROAD1's time to receive the green light, but ROAD1 is not congested at all, while ROAD2 is highly congested. It is not an effective strategy to assign an identical time period for different traffic densities. This traffic system is able to recognize the situation and provide a time period for ROAD1's green light depending on the traffic density on that specific road, ensuring sufficient traffic management based on the traffic density of each road. Figure (6) illustrates ROAD1's yellow state flow chart, which shows varying time durations based on traffic density.

Setting priorities for emergency vehicles: When the system recognizes an emergency vehicle, it allows the emergency vehicle to pass promptly. Figure (3) illustrates the situation in which an emergency vehicle is recognized on ROAD3 and the system's state transition. ROAD1 has the green light switched on. The traffic system detects the emergency vehicle on ROAD3 and then activates the red light for ROAD1 and the green light for ROAD3, enabling the emergency vehicle to continue on its way. The traffic system then returns to ROAD1.

Prevent the junction blockage: This approach prevents junction blockage when one or both of the primary roadways from which cars depart the junction become jammed. If the system grants one of the roads the green light, the junction will be blocked even if the other roadways are not congested. Figure (4) illustrates the state transition that occurs once a road blockage signal is received. The system enters an ALLRED state, in which red lights are activated on all roads to offer extra time for congested roadways to become less congested. Figure (8) illustrates the situation when ROAD1_BACK and ROAD3_BACK are congested and ROAD2 is granted a green light.

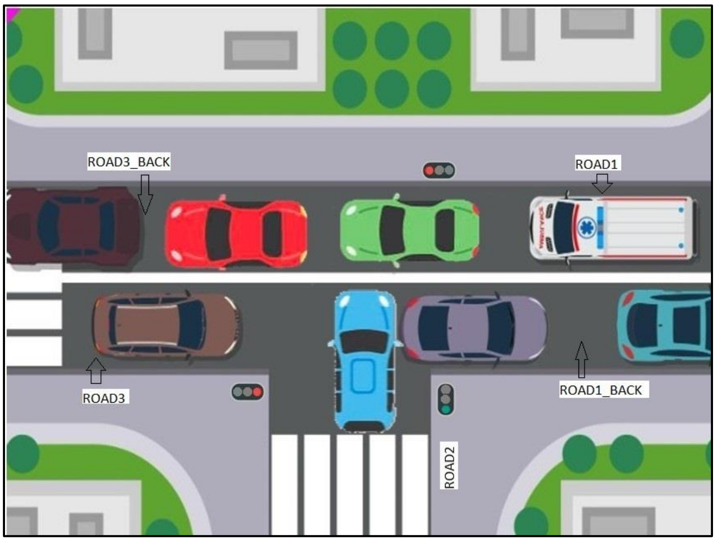

Figure 8: Three-way junction blockage view.

License Plate System

The proposed automatic license plate system consists of five parts:

1. Red-light offences detection and image acquisition.
2. License plate detection.
3. Image pre-processing.
4. License plate number segmentation.
5. License plate number recognition.

Red-light offences detection and image acquisition.

At the junction of a three-way, sensors are set at a distance from each of the junction' entrance points. These sensors work in synchrony with the red-light traffic signal to enable and disable the sensors on each route at the same time. The infrared sensor detects each vehicle that enters the junction when the light is red. When a violating vehicle is detected, the camera assigned to that section of road is activated and a picture of the offending vehicle is taken. At that time, the FPGA gets the vehicle picture. The camera was set up to capture pictures of the front of an automobile that was violating the law, including the license plate. Each camera is located at a distance ahead of the junction's entrance point, facing the road that allocated it.

License plate detection

A machine learning algorithm is implemented to determine the license plate region in the image. aggregate channel features (ACF) object detector from the MATLAB computer vision toolbox was trained to detect license plates.

Aggregate channel features The MATLAB toolbox's object detector use the boosting technique to construct an ensemble of weaker learners. Higher values in the boosting method can be used to increase detection accuracy at the trade-off of slower detection performance rates. The recommended range is 300 to 5000. A form of Ensemble learning approach is the boosting method. Instead of developing a single model for prediction, ensemble learning is used. We will create a number of machine learning models known as weak learners. The strong learner is formed by combining all weak learners. To create multiple models, we will divide the available training data into smaller datasets, which will be utilized to create several weak learns. A strong learner model is formed by combining all weak learners. To create multiple models, we will divide the available training data into smaller datasets, which will be utilized to create several weak learns. The sum of all weak learners results in a strong learner or strong model. The boosting approach is used to develop a stronger learner, and all individual models are built progressively. That is, the outcome of the first model is passed on to the next model, and so on.

The ACF license plate detector was trained using a dataset of Libyan license plates. The detector identifies license plates from a variety of angles and distances, as well as skewed license plates. The license plate detector was trained on 3069 images before being tested on 746 images of vehicles with license plates from various angles and distances. The data set was obtained from streets with varying lighting conditions and labelled manually. The machine learning approach is utilized for license plate localization because it provides the best results that image processing approaches cannot grant under a variety of settings such as variances in lighting, distances, and angles of the obtained image. The license plate detector's performance on the test data yields an average precision of 0.9. Figure (9) shows the detected license plate.

Figure 9: The detected license plate.

Image pre-processing

Following the determination of the license plate region, the license region goes through image processing steps to prepare it for digit segmentation and recognition.

Image processing: is a method to perform some operations on an image, in order to enhance the image or to extract some useful information from it. In this algorithm to prepare the license plate image for number extraction and recognition we used built-in functions from the MATLAB image processing toolbox. We used image processing functions to crop the license plate and convert it into a binary image, resize it, remove small objects from the binary image, and label connected components in the 2-D binary image, etc.

Extract the license plate region.

The license plate detection using machine learning results in a rectangular bounding box that describes the spatial location of an object the (x,y)-axis coordinates of the bounding box, and the width and height of the box. Then the license plate region is cropped using the Matlab function “imcrop”, which takes the car image and the bounding box as parameters and returns the cropped license plate image as shown in Figure (10).

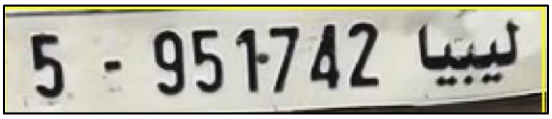

Figure 10: The cropped license plate.

Convert the license plate to a binary image.

The Matlab built–in function “imbinarize” is used to convert the RGB image of the license plate into a binary image as shown in Figure (11). This step is needed to prepare the license plate’s number for segmentation.

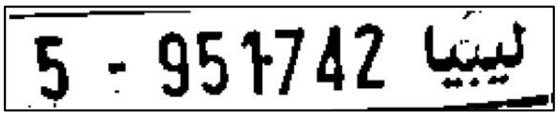

Figure 11: Binary image of license plate.

Skew Detection and Correction.

Skewed license plates reduce the accuracy of the license plate number recognition, hence fixing skewed license plates is critical. In this method, the Hough transformation is applied. Hough transform is essentially a feature extraction approach that is used to identify lines and locate arbitrary shapes in a given image. The steps below outline the skew detection and correction method:

a) Convert the image to grayscale.
b) Apply Canny or Sobel filter.
c) Locate Hough lines with angles ranging from 0.1 to 180 degrees.
d) Round the angles between the line peaks to two decimal places.
e) Determine the angle with the highest frequency of occurrence.
f) Rotate the image by that angle.

Figure (12) shows the result of the skew correction.

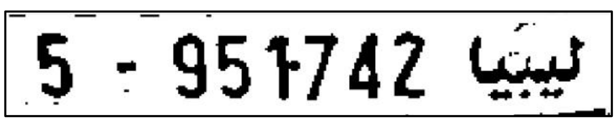

Figure 12: Skew corrected license plate.

Remove the black border.

Because the black border might create issues while extracting the license number, it is deleted as follows:

a) Apply two-dimensional median filtering to the picture of the license plate.
b) Fill up the holes.
c) NOT (XOR (median filtering result, hole filling result)).

Figure (13) shows the resulting image after removing the black border.

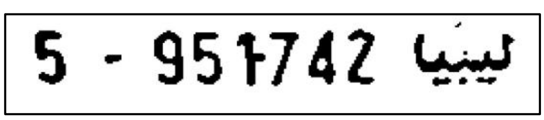

Figure 13: License image after removing black border.

Invert the image colour.

The color of the license plate must be inverted because the numerals and characters must be white in order to be segmented using connected-component labeling.Figure (14) shows the license plate image after inverting its color.

Figure 14: Inverted license plate image color.

Remove small items.

The license plate is scaled to a set size, in this example [136 513], and then all items with a size less than 400 pixels that are neither numerals or the word "ليبيا" are eliminated. Figure (15) illustrates the license plate image after small items have been removed.

Figure 15: License plate image after removing small objects.

The license plate number segmentation.

A license plate number is segmented using connected-component labelling. From top to bottom and left to right, each connected component labelling extracts items from an image. Given that Libyan license plates are rectangular in shape and have all of their numbers on a single line, the connected-component segmentation approach is applicable. Figure (16) illustrates the license plate numbers after they have been segmented.

Figure 16: Extracted license number

License plate number recognition

The digits and words are categorised using Image category classification after the license plate number is segmented by constructing a histogram of visual word occurrences that reflect the images. A bag of visual words or bag of features is a histogram that is used to train an image category classifier. The classifier is trained using a custom dataset of digits and the word "" Libya. To train and test the classifier, a total of 550 images are employed. The dataset has 323 images for training and 137 images to test the classifier. The evaluation of the classifier produces an average accuracy of 0.966. Figure (17) illustrates the outcome of the license plate recognition step.

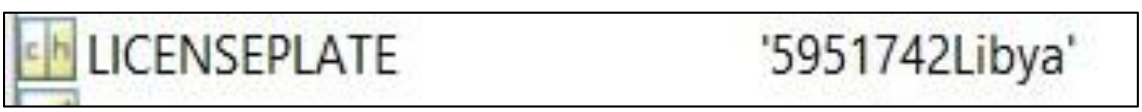

Figure 17: Result of license plate recognition.

RESULTS and DISCUSSIONS

To test the system's functioning correctly, we generated a timing diagram with Modalism-Intel FPGA. In the timing diagram, the traffic lights for the three roads are represented by R1light, R2light, and R3light, where 100, 010, and 001 refer to green, yellow, and red, respectively. The inputs IS1, IS2, IS3, IS1_BACK, and IS3_BACK reflect the IR sensors for ROAD1, ROAD2, ROAD3, ROAD1_BACK, and ROAD3_BACK, with values ranging from 00000 to 11111 reflecting the varied traffic levels on every road. For simulation purposes, EM input represents a passing emergency vehicle on a certain road. The EM input values are 0xx, 100, 101, and 110, which represent no emergency vehicles, an emergency vehicle on ROAD1, an emergency vehicle on ROAD2, and an emergency vehicle on ROAD3. The state register reflects the current status of the traffic system; the delayR1G, delayR2G, and delayR3G indicate the time set to ROAD1, ROAD2, and ROAD3 for the green signal, respectively; and the delay RY represents the time assigned to all roads for the yellow traffic light. Green light periods of 60, 48, 36, 24, 12, and 1 second are available. These values are determined by the active infrared sensors. Figure (18) illustrates varying traffic signal timings based on traffic density on each road. The green light period is determined by the number of active sensors on each route. Here in the timing diagram, you can see the IS2 (the infrared sensors on ROAD2) value is "10111" The number of active sensors is four based on the fact that the green signal period for ROAD2 is decided to be 36 sec. Also, you can see the IS3 (the infrared sensors on ROAD3) value is "00110" The number of active sensors is two based on the fact that the green signal period for ROAD3 is decided to be 24 sec. Figure (19) illustrates a scenario in which the ROAD1_BACK value is "11111," indicating an extremely congested route, if we provide a green light to any road that would trigger the situation illustrated in Figure (8). This problem can be solved by switching on all red lights on all routes, allowing the roadways to become less congested. As shown in the timing diagram, the traffic system enters an all-red state (ALR) until the ROAD1_BACK value changes to "01110," indicating that the road has become less congested. Figure (20) illustrates an emergency vehicle on ROAD3 (EM value "110") while the yellow light on ROAD1 is activated. The traffic system prioritizes ROAD3 to allow the emergency car to pass when the emergency has passed (EM is "000"), the system returns to the original state when ROAD1's yellow light is on, and the system proceeds on the normal state transitions. Figure (21) illustrates the FPGA implementation of the system.

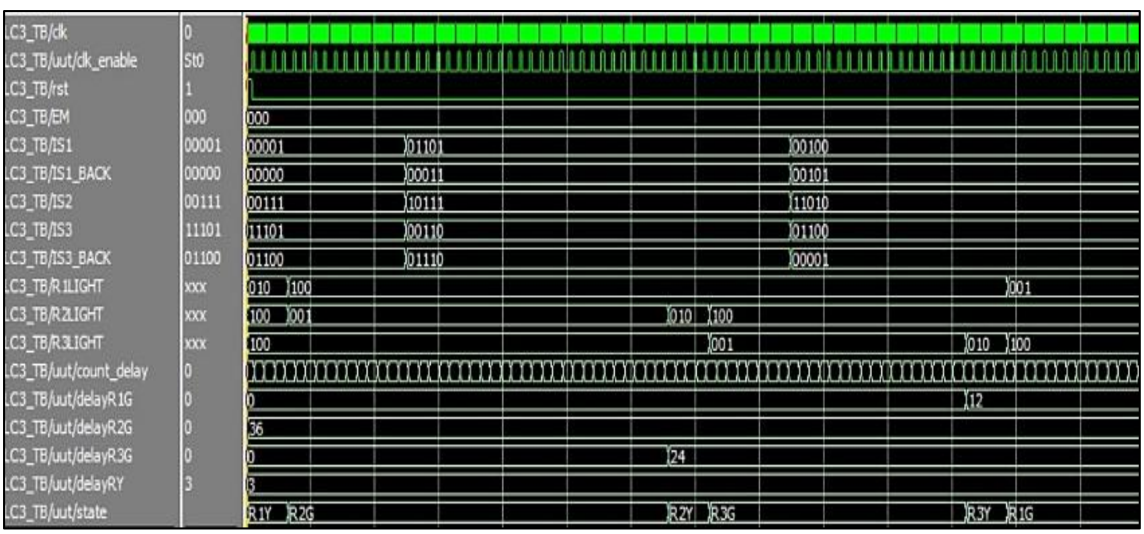

Figure 18: The timing diagram for the real-time density-based traffic system.

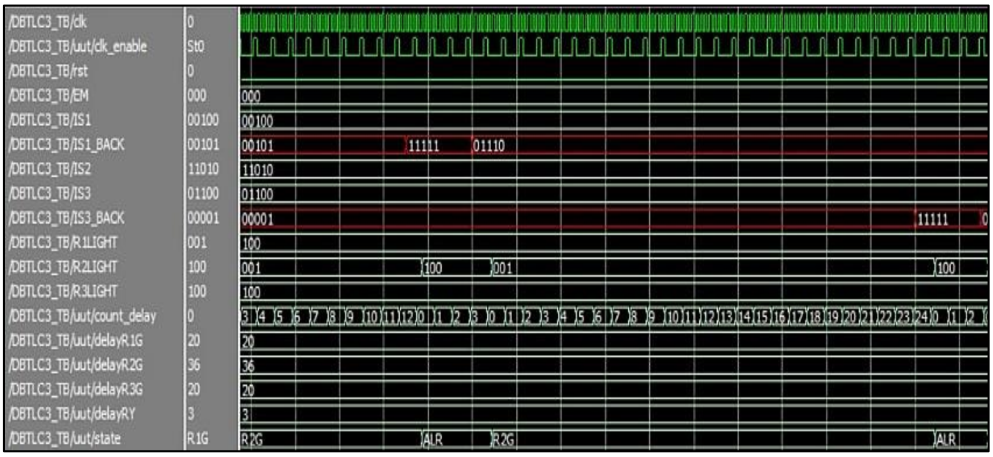

Figure 19: The timing diagram for the real-time density-based traffic system with road blockage signal.

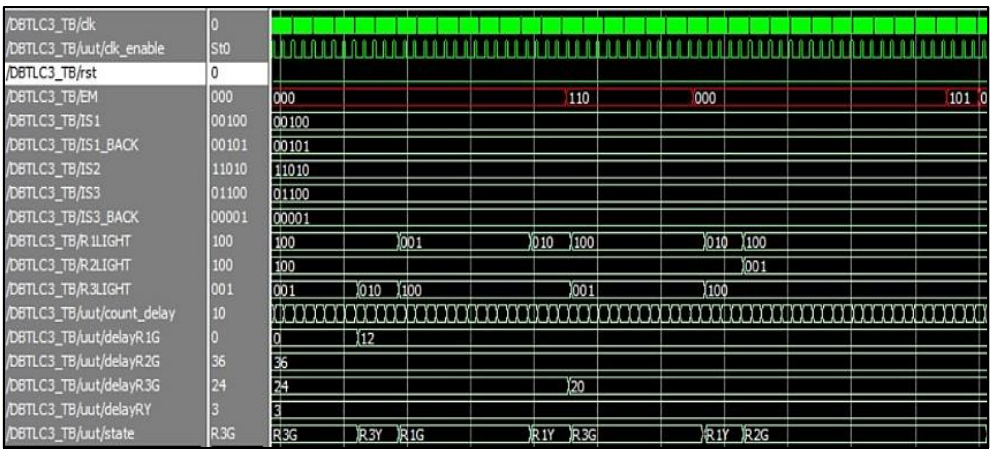

Figure 20: The timing diagram for the real-time density-based traffic system with an emergency car on ROAD3.

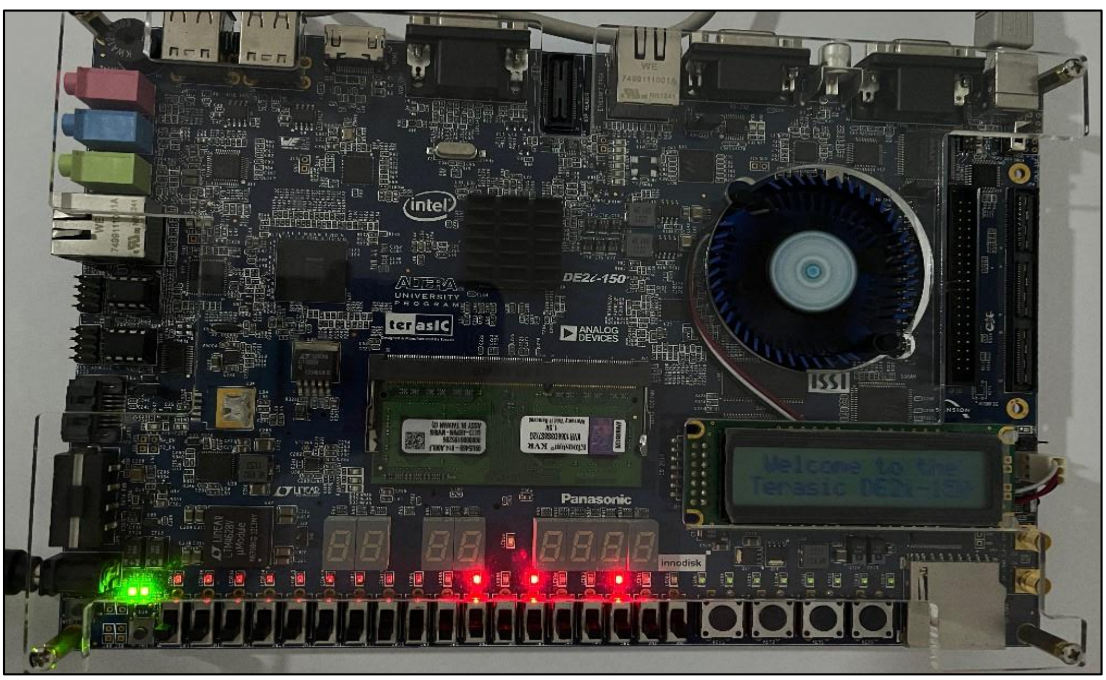

Figure 21: The traffic system on FPGA.

MATLAB is used to develop the Libyan license plate recognition system. The license plate detection stage was carried out with the use of an ACF license plate detector, which was trained on 3069 images and tested on 746 images of license plates with various illuminations, distances, and tilted license plates. The license plate detector's performance on the test data yields an average accuracy of 0.9, as illustrated in Figure (22). Because of the pre-processing stage, which greatly aided in enhancing the segmentation step, the license plate number segmentation step performed perfectly on 70 test images. The classifier was trained using a dataset of numbers and the word "ليبيا" Libya in the license plate number recognition stage. To train and test, 550 images were used in total. The dataset was split into 323 images for training and 137 images to test the classifier. The testing yields an average precision of 0.966.

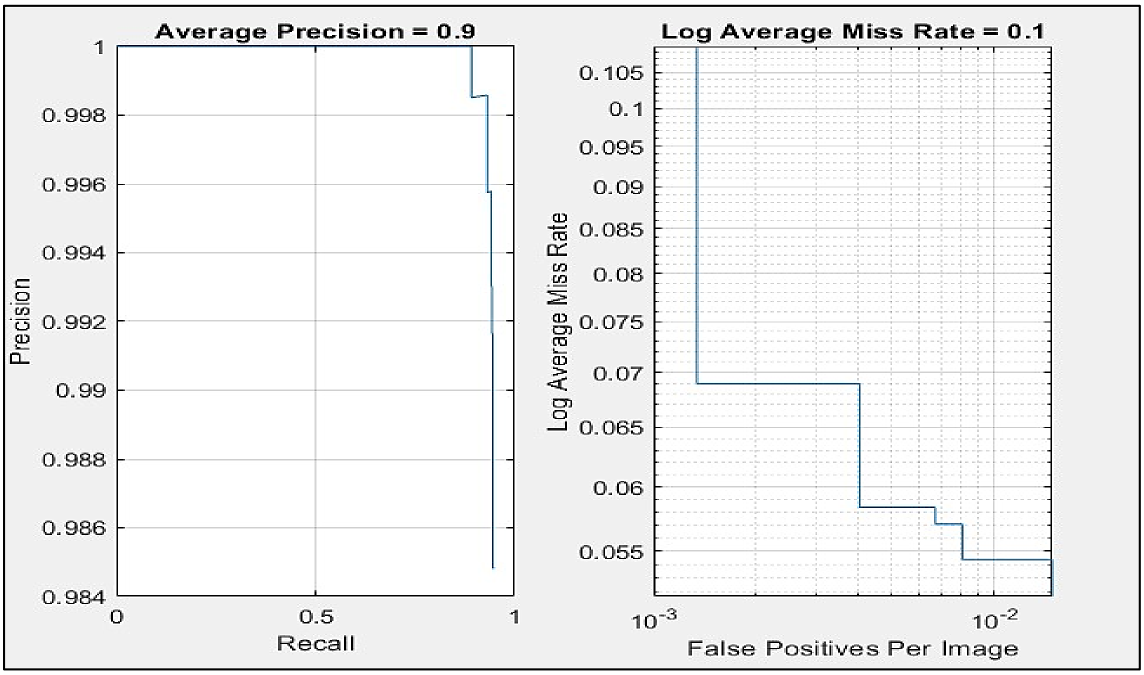

Figure 22: Evaluation of license plate detector.

Comparative analysis

This system is a design of a complete traffic system for managing and monitoring traffic on the FPGA board. In the part of license plate recognition, we aimed to use machine learning instead of digital image processing to detect the license plate region which resulted in a more accurate detection of the license plate region. Table (2) shows a comparison of the license plate detection step between our proposed system which uses machine learning and the license plate recognition system for Egyptian cars which uses image processing techniques.

Table 2: License plate localization precision.

Method	No. of Images	No. of Successful Prediction	precision
Proposed	746	671	90%
[9]	221	173	78%

CONCLUSION

The main contribution of this research is the design, simulation, and implementation of a density-based real-time traffic signal system and a recognition system for Libyan license plates. The traffic system is intended to reduce congestion and traffic flow at junctions. The technology can make real-time choices based on traffic volume and determine the appropriate traffic signal timing for each road. The power of the system to prioritize emergency vehicles and make good decisions to avoid blocking junctions is one of its main advantages. After testing the traffic light system module, the design was simulated and validated using Intel-Altera software utilities. The system was then constructed and tested in hardware using the Cyclone IV GX: EP4CGX150DF31C8 FPGA board to guarantee that it operated as expected. The system for detecting license plates is utilized to track traffic offences through the application of machine learning and image processing algorithms. The MATLAB software package is used to program and test the license plate software.

REFERENCES

[1] B. U. Umar, O. M. Olaniyi, A. James, and O. R. Isah, "Traffic violation detection system using image processing," Computer Engineering and Applications, Vol. 10, No. 2, June 2021.

[2] J. R. Aworemi, I. A. Abdul-Azeez, A. J. Oyedokun, and J.O.Adewoye, "A study of the causes, effects and ameliorative measures of road traffic congestion in Lagos Metropolis," *European Journal of Social Sciences*, vol. 11, no. 1, pp. 119-128, 2009.

[3] P.K. Singh and P.Daniel, "Advanced real traffic light controller system design using cortex-m0 IP on FPGA," IEEE International Conference on Advanced Communications, Control, and Computing Technologies, pp. 1023–1026, 2014.

[4] 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 (IJARCET), vol. 1, no. 7, pp. 2278–1323, 2012.

[5] W. M. El-Medany and M. R. Hussain, "FPGA-based advanced real traffic light controller system design," Proceeding of 4th IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, ISBN: 978-1-4244-1347-8, IEEE, pp. 100- 105, 2007.

[6] Y.Liu and X.Chen, "Design of traffic lights controlling system based on PLC and configuration technology," Proceeding of International Conference on Multimedia Information Networking and Security, ISBN: 978-0-7695-3843-3, pp.561-563, 2009.

[7] L.A.Ali, M.A.Albakkosh, and M.M.Eljhani," University of Tripoli’s Real-Time Density-Based Traffic Light Controller on FPGA, International Science and Technology Journal, September 2022.

[8] S.Singh and S. C. Badwaik, "Design and implementation of FPGA-based adaptive dynamic traffic light controller," International Conference on Emerging Trends in Networks and Computer Communications(ETNCC), pp. 324–330, 2011.

[9] S. Bayoumi, E. Korany, and S. Fathy,“ License Plate Recognition System for Egyptian Car Plates,” International Conference on Image and Video Processing and Computer Vision, July 2010.

[10] A. M. Hassan, S. A. Ghoul, and A. A.Alkabir. "Libyan Vehicle License Plate Recognition with Support Vector Machine," Al-Mukhtar Journal of Sciences, vol. 37, no. 1, pp. 1--13, 2022.

[11] C. TU, S. DU. A hierarchical RCNN for vehicle and vehicle license plate detection and recognition.” International Journal of Electrical and Computer Engineering”, vol. 12, no. 1, pp. 731--737, 2022.