

Auto Vehicle Driving Assistance

Khalid A. Al-Shalfan

Abstract—As the growth of automation of daily life careers increases, An intelligent vehicle driving system became an essential necessity nowadays. One of its functions is the ability of recognizing the traffic lights and detecting their status. In this paper, a system for monitoring the traffic light status is presented, by passing the video frames through shape and color classifiers. The shape classifier is based on the extracted edges as the classifier parameter. It uses Multilayer Feed forward Neural Network for classification. Alternately, the shape classifier can be implemented using template matching where the traffic light is detected using its perspectives ratio and the matching with prepared traffic light templates is accomplished. The algorithm tries to find the area of interest where the traffic light may be found. This is done based on SUSAN reliable corner detection procedure, because noise tends to cause false corners. A feature of the SUSAN corner detector is its ability to find corners by direct analysis of the image intensities rather than their derivatives, thus removing the need for smoothing to reduce noise. This makes it faster than most of the other corner detectors like Harris corner detection algorithm and it also provide better localization of the corners regardless of the mask size used. The color classifier uses a range of the red color histogram to recognize the red color of the traffic light. The experimental results show that the proposed system has high performance both in shape and color detection.

Index Terms—Intelligent driving systems, image processing, image matching, neural network, artificial intelligent.

I. INTRODUCTION

In Saudi Arabia, the statistical data reported by the department of statistics of King Abdul-Aziz City for science and technology (KACST) manifested that approximately 4,293 people were killed and more than 30,439 injured in the year 2003 due to traffic accidents [1]. Passing the red traffic light is one of the main causes. Drivers tend to drive while using their cell phones which often results in reduced the attention to the road. Such factors result in an increased rate of accidents and fatalities. In that context, the idea of an intelligent vehicle was come up, where they have useful tools that provide them with the ability to reinforce safety. Recently, one of the advanced techniques that have been widely investigated is the ability of recognizing the traffic lights and detecting their status.

The field of traffic has seen big improvements with new more powerful techniques being introduced and applied research supported by large companies in the automotive industry [2]. Zhuowen and Ron [3] proposed a system that generates hot-spot maps using color indexing and edge gradient indexing and recognizes traffic lights using Markov

Chain Monte Carlo method. They thought that color image segmentation methods are time consuming with results that were not good enough for further processing. So they developed a new method that integrates bottom-up and top-down methods using the Markov Chain Monte Carlo method. They combined traditional techniques such as indexing and the Generalized Hough Transform together to show that they could be nicely integrated in random processes. Chung, Wang and Chen [4] proposed a traffic light detection system at intersections. They employed the fuzzy and morphological techniques since the researches related to utilize fuzziness and morphological methods on applications have been proven very powerful. They designed an algorithm to detect horizontal lights. However, the algorithm can be easily adapted to detect a vertical layout traffic light. A fast background generation method is employed to generate and update background images. This method is very quick but it does not create a perfect background image. Illumination analysis is applied on the background images to determine the daytime or nighttime. The fuzzy methods and morphological techniques are utilized together to find the candidate traffic light areas. With the relative spatial and temporal information, the scales, positions, and timing sequences of traffic lights are acquired. Their system has been tested on outdoor complex scenes, with different illumination conditions and different camera angles.

A system enhanced with vehicle data and GPS is proposed [5]. This system consists basically of three modules: detector, tracker and classifier. The detector has the most critical job. It has to deal with large amount of data. The traffic light is detected either by shape or texture. The circle finder is based on a generalized Hough transform using Sobel based gradient directions. For the shape based detector a Matched Filter Algorithm is used. The tracker groups candidates from consecutive images. It compensates for drop outs from the detector using a suppressed spontaneous detection hypothesis. The classifier makes a decision by soft majority voting as soon as the collected probability mass is large enough. They used feed-forward neural networks for classifier adaptation. The color detector uses a modified Gaussian-distribution classifier. Then the result is segmented by the use of connected component analysis. Eddowes and Krahe [6] worked on a project named "Traffic Lights Recognition in a Scene using a PDA". Their project is concerned with the recognition of pedestrian traffic lights, for Blinds' usage. The system passes two stages, a segmentation stage in which the contour of human is detected. And a recognition stage in which the decision is made whether the traffic light red or green. In the segmentation stage regions out of interest are eliminated based on color detection. And then the contour of the human is calculated using morphology. In the recognition stage a structural method is used. The

Manuscript received February 12, 2015; revised June 5, 2015.

Khalid A. Al-Shalfan is with College of Computer & Information Sciences, Allmam University, Kingdom of Saudi Arabia (e-mail: kshalfan@gmail.com).

objective was to guarantee 100% the recognition of green lights for reasons of safety but that was not the case. There were cases in which two green and red traffic light were not recognized.

In this paper, a traffic light recognition system is developed, which supports the capabilities of the driving assistant system by monitoring the current traffic light status and providing them with valuable safety-critical information. Such a system can prove to be importance in the sense that it can alert the driver to the traffic light situations that he might overlook. It might also help color blind people in traffic light status recognition. The Traffic light recognition system is intended to be used in a real-time environment. Therefore, the algorithms must be fast and reliable enough to achieve the system objective. However, the system does not handle unexpected changes in the environment such as broken-down traffic lights. In Saudi Arabia, the traffic light lies in the left or right side of the road. The system concerned on left traffic lights only. This choice is selected after a comprehensive analysis of the Saudi traffic system. Traffic light shapes are mostly restricted to three shapes, as shown in Fig. 1. These three shapes originated traffic light comes from the lying of the lights in the traffic light sides. The existence of another shape is rare in the studied environment. The system then will be limited on the detection of the three shapes.

The system detects traffic lights with two sequential classifiers, one a shape classifier and the other a color classifier. The shape classifier's task is the recognition of the traffic light's shape while the color classifier uses the histogram of the red color to verify the detected object and to decide the traffic light status. The shape classifier can be implemented using one of two approaches. The first approach uses a neural network where recognition is based on the traffic light edges. The second approach uses template matching based on the perspectives of the traffic light.

II. DESIGN DECISION

The system was designed to be operated in two stages, the first stage is to classify the traffic light shape and the next is to detect its color. This decision increases the system reliability, since the two classifiers consolidate each other.

A. Shape Classification

Classify shapes is achieved using two methodologies, neural networks and template matching. These methodologies were chosen due to the fact that they offer a high degree of reliability. An area of interest is selected in order to increase the efficiency of the system. Discarding an area of 228×84 from the bottom, 92×240 from the right and 228×30 from the top of the image will speed up the system since the searched area will be decreased to 37.4%, while the detection reliability will not be affected. Frame acquisition must occur after a suitable period of time. We choose one second as the time interval for frame capturing. We selected this value in order to avoid the missing of traffic light switching events as much as possible. Furthermore, reducing the time interval between frame acquisitions increases the processing time without a noticeable gain in results.

In designing the object detector a number of issues must be dealt with. First, features must be generated. The generated features should inhibit high information backing in order to remove information redundancies. The image's pixels have a large degree of correlation due to the internal morphological consistencies of real-world images. Thus, using image processing in order to generate suitable features becomes essential. In the case of traffic light detection, the traffic light shape embraces a high level of object representation since traffic lights have a distinguishable shape. This representation can be used to represent the traffic light even if it is far away. The object shape can be obtained through edge segmentation [7], [8]. For object classification, a neural network is used [9], [10]. It is suitable for pattern recognition. Feedforward backpropagation neural networks can be used efficiently to classify object.

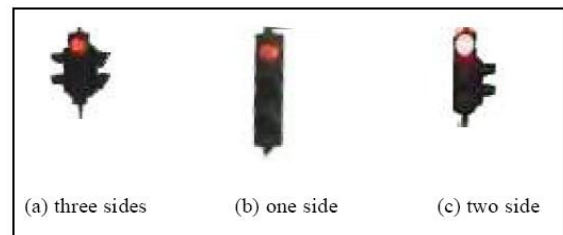


Fig. 1. Traffic lights shape: (a) TL1 shape, (b) TL2 shape, (c) TL3 shape.

Constructing an Artificial Neural Network (ANN) is a significant action where the form of inputs, number of layers, and the nodes in each layer as well as the network architecture play a critical role in the training procedure. Therefore, they must be designed carefully since they affect the training convergence. The network is designed to have four layers where the first layer is fed with the pixels of the binary image as neural inputs. The output is a 4 bit pattern holding the classification result. Two hidden layers are used to increase the network performance. Shape classification is performed by matching the image with presorted object templates [11], [12]. Template Matching relies primarily on the templates used. Collecting templates is a big issue because the matching process differs with templates of variable distance from the traffic light. Furthermore, a primary requirement was to cover all the variations in orientation of the same traffic light in order to get a robust result. So, initially we must collect a sufficient number of templates. A sufficient number of templates of red traffic lights is to be collected. The matching can be done by directly measuring the similarity values between two images' pixels. The range of accepted maximum similarity values must be specified. Then the coordinates of the points of maximum similarity values in the original image are identified in order to determine the traffic light position in the image. Since TL1 has a regular geometric shape, it is possible to benefit from this criterion during shape classification. The traffic light width and height are determined in order to calculate the ratio (height/ width). See Fig. 2.

The matching can be performed based on a perspectives ratio decision. The match will occur between the traffic light and template whose perspective ratios are most similar. The range of the accepted maximum matching values resulting from the matching is specified.

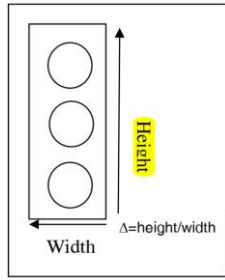


Fig. 2. Identifying the traffic light perspective to ensure correct recognition.

B. Color Classification

The hue-saturation system is the most widely used color space in traffic light detection because it is a very important and attractive color model for image processing applications. HSV space represents colors similar to the human eye color sensing manner [13] and has a capability to reject unwanted color shades. HIS color space gives more details about color information since it is represented by a double cone as in Fig. 3, whereas HSV is represented by one cone only as in Fig. 4. This representation of HSI involves more mathematical operations. So, our image is converted to an HSV color space which offers good color representation and less mathematical operations.

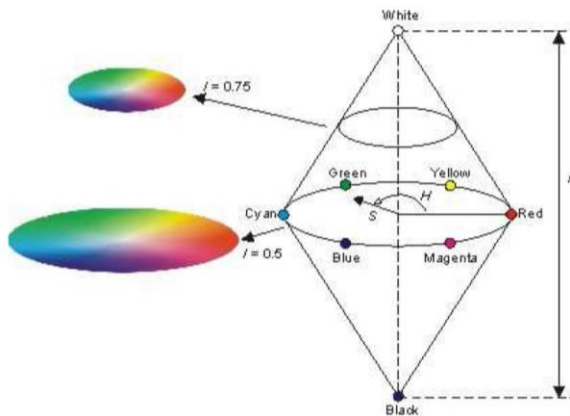


Fig. 3. HIS color space.

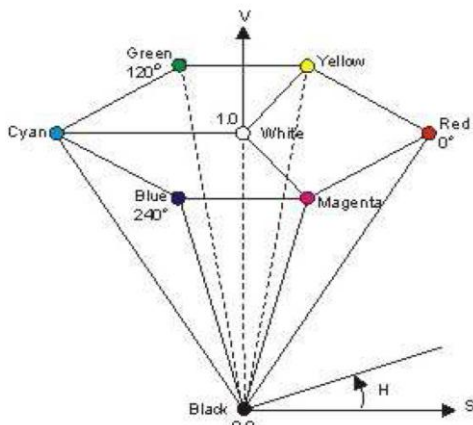


Fig. 4. HSV color space.

After converting the color of the detected shape, threshold segmentation; using a one dimensional histogram, is used to segment the detected shape into a red color region. This technique is used due to the requirement to bind the red region to a specific threshold. However, in region-based segmentation (region growing) one requires a seed point to

begin the growth of the segmented region which cause a growth in processing the image.

III. DESIGN SPECIFICATION

This section encompasses more detailed design specifications. The design stages stated before are described with sequence diagrams shown in Fig. 5 and Fig. 6.

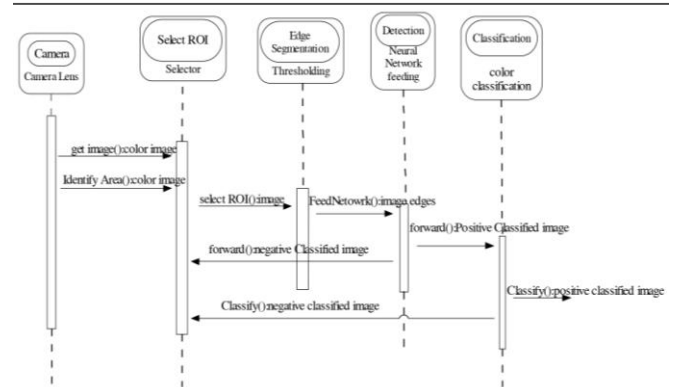


Fig. 5. Sequence diagram for neural network.

Fig. 5 illustrates the system design using a neural shape classifier. It shows that the area of interest is identified as the image is acquired from camera. The selector selects a region to pass it to the two classifiers. The image is then passed to the edge extractor and the image edges passed to the neural network. If the classifier detects the traffic light shape the result will be passed on to the next stage, otherwise another region will be selected. The color classifier will make the decision on the traffic light status and when the classification fails another region will be selected.

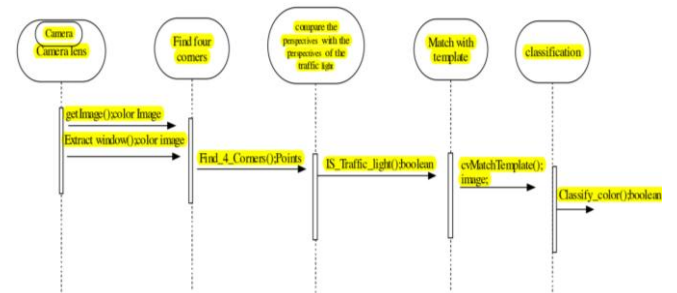


Fig. 6. Sequence diagram for template matching.

Fig. 6 illustrates the system design using a template matching classifier. It shows that as the image is acquired from camera, a window is extracted. The system finds four corners in the extracted image. The perspective ratio of the corners is calculated and compared with the traffic light perspective ratio. Then the matching is performed with the stored templates. If a match is found the shape is passed on to the color classifier to detect the red color.

IV. SYSTEM IMPLEMENTATION

A. Shape Classification

1) Neural Network

Before the recognition system is implemented, stand alone

algorithms are developed in order to set up the system detection phase. These algorithms are implemented to prepare the training set for training and to train the network. A pre-action is taken to make the training data ready. Traffic light 'bmp' images of size 38×42 pixels were collected. This size is chosen in the sense that it is the suitable size to specify the traffic light shape from a predefined distance; see Fig. 7a. These data sets are categorized into three groups; TL1, TL2, and TL3 traffic lights groups. Each training group is considered as an individual class. This categorization enhances the training and future processing. A fourth group, with the same size that represents a nontraffic light data set, is considered as another class. The program loops through these data sets, reducing their size by half; i.e. 20×21, using the Pyramid Down algorithm, which downsamples the image by rejecting even rows and columns. The edges are extracted by applying the Canny algorithm on each image; Fig. 7b. The resulting binary image bits are stored in one file as a training element; Fig. 7c. Sequential access files are used for data storage in the implementation phase.

A neural network is constructed out of 4 layers, an input layer of 420 inputs, two hidden layers of 281 and 148 neurons respectively and an output layer of 4 outputs. The summation function is initially set to be sigmoid. The network connections weights are initialized using a Widrow-Nguyen algorithm.

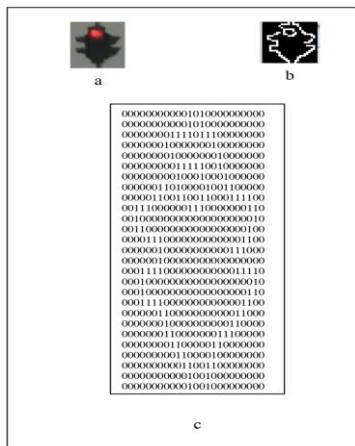


Fig. 7. Training set sample of 42×38.

The Widrow-Nguyen algorithm picks small random values as initial weights for the neural network. The weights are then modified in such a manner that the region of interest is divided into small intervals. It is then reasonable to consider speeding up the training process by setting the initial weights of the first layer so that each node is assigned its own interval at the start of training. As the network is trained, each hidden node still has the freedom to adjust its interval size and location during training. However, most of these adjustments will probably be small since the majority of the weight movements were eliminated by Widrow-Nguyen method used in setting their initial values. The network training starts by loading the training set prepared previously. The training is performed with the parameters specified in Table I. After training the network performance is measured. If it is not high, the network is trained with another training parameters or with another training set. After a reasonable result is obtained the corresponding network is approved and saved in a file.

TABLE I: TRAINING PARAMETERS

Tring parameter	Values
Learning rate	0.9
Epochs	1,000
Error rate	0.001

The system captures a video sequence from a video file or from a camera of size 320×240. The system obtains a frame every 1 second. The extracted frame is then processed as follows:

1) Identifying Area of Interest

As stated earlier, 37.4% of the extracted frame will be the identified area of interest that should be scanned. The scanning process is performed by continuously selecting a rectangular area. In each iteration a region of 38×42 is selected for further processing. The adjacent region is selected in the next iteration; Fig. 8, using a 19 and 21 pixel walking step in the X and Y directions, respectively.

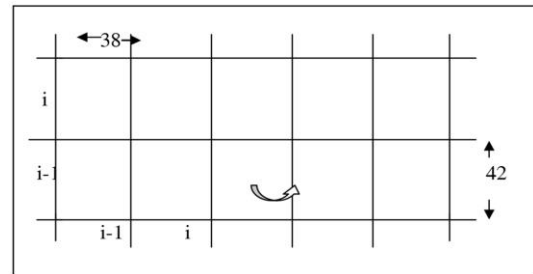


Fig. 8. Scanning a frame.

2) Shape Classification

The selected region is resized to 20×21 using a pyramid Down algorithm. The region edges are extracted by applying the Canny algorithm. The resultant binary image bits are fed to the trained network. The network output is tested and if a positive result is obtained, the selected region is passed to the color classifier, otherwise the program proceeds with the scanning and the next scanned region is processed.

B. Template Matching

The implementation of template matching can be roughly divided into the process of preparing the templates and the matching process. The matching can be done by matching the image and the templates directly. Alternately, the matching will be more reliable when the perspectives ratio of the rectangular traffic light is used. The implementation is discussed below:

- 1) Templates Preparation: The templates are collected from realistic images of Riyadh's roads. The collected templates have special characteristics. They must have a size less than or equal to the size of the matching window. Furthermore, the templates must be clear; this means that they preferably be in clear background with as little a number of objects as possible. Also, occlusion will affect the matching. Such occlusion may occur because of the trees or the sign in the front of the traffic light. As a result, the templates were collected with a clear background without occlusion.
- 2) Matching: The matching process is accomplished. It compares the overlapped patches of the image scene with that of the stored image using (1).

$$R(x, y) = \frac{\sum_x' y'[T'(x', y') \cdot I'(x+x', y+y')]}{\sqrt{[\sum_x' y'^2 T'(x', y')^2 + \sum_x' y'^2 I'(x+x', y+y')^2]}} \quad (1)$$

where I denotes image, T -template, R -result.

The summation is carried out over template and/or the image patch: $x'=0..w-1, y'=0..h-1$. The point for which $R(x, y)$ is the largest is considered as the location of the object within the image scene. Thus, we are left with a set of points (x_i, y_i) , where i represents the number of stored images of the object. `cvMatchTemplate()` specifies the result image where you can find the maximum or minimum match's value. The location of the match within the result image can be found by using the function `cvMinMaxLoc()`. However, a big maximum value can be found within images that contain objects that differ completely from the traffic light. The solution was to search for the maximum value of the match between the image and all of the available templates. The process is illustrated below with the experimental images, see Fig. 9.

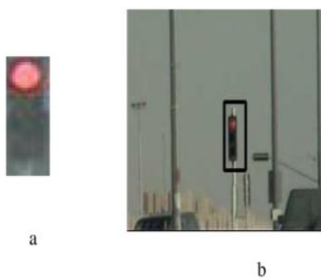


Fig. 9. Matching without using traffic light perspective. (a) Template sample, (b) Mark the area centered by the maximum value of the match.

Matching the extracted image directly with the template has a minimum control on the matched images. There must be a way to judge that traffic lights exist in the extracted window before applying the match. This is carried out by manipulating the extracted picture to search for rectangle-like shape with specific perspectives. In Saudia Arabia, rectangle like traffic lights have a well known perspective. We studied the prevalent shapes of traffic light under the assistance of the traffic center of Riyadh. The algorithm tries to find the area of interest where the traffic light may be found.

The fastest method to do this is the use of a reliable corner detection procedure, because noise tends to cause false corners. A feature of the SUSAN corner detector is its ability to find corners by direct analysis of the image intensities rather than their derivatives, thus removing the need for smoothing to reduce noise. This makes it faster than most of the other corner detectors, and it also provide better localization of the corners regardless of the mask size used. The basis for the SUSAN principle is the use of an approximately circular mask with an area of n_{max} pixels and the nucleus at a corner feature, to give an isotropic response, moving over each pixel in the image as the center or nucleus of the mask. The intensities are said to be similar when their difference is less than a defined threshold t . This threshold can be used to control the quantity of the output without significantly affecting the quality of the results. The response function used in comparing intensities is

$$c(r, r_0) = \exp\left(-\left(\frac{I(r) - I(r_0)}{t}\right)^6\right) \quad (2)$$

where, r_0 is the position of the nucleus in the image, r is the position of any other point within the mask, $I(r)$ is the intensity of any pixel, and t is the difference threshold. With the response function of equation (1), the area of the SUSAN with nucleus r_0 is

$$n(r_0) = \sum_r c(r, r_0) \quad (3)$$

For there to be a corner at r_0 , $n(r_0)$ must be less than half of its maximum possible value n_{max} . The number of pixels $n(r_0)$ is compared with a geometric threshold g , normally set to $n_{max}/2$. This gives a decision function.

$$R(r_0) = \begin{cases} g - n(r_0) & \text{if } n(r_0) < g \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

The function is non-zero at all candidate points for corners. Reducing the geometric threshold g restricts the candidate points to those with sharper angles. The center of gravity of the SUSAN is calculated at each candidate point, according to the following formula:

$$\bar{r}(r_0) = \frac{\sum_r r c(r, r_0)}{\sum_r c(r, r_0)} \quad (5)$$

For any candidate point to be a genuine corner, r must be distinct from r_0 . Thus candidate points are rejected if $|r - r_0| < 2$. As a final test, all the points lying on the line from the nucleus r_0 , through the center of gravity r , to the edge of the mask are tested. If all these pixels are in the SUSAN, the nucleus is accepted as a corner point; otherwise, it is rejected.

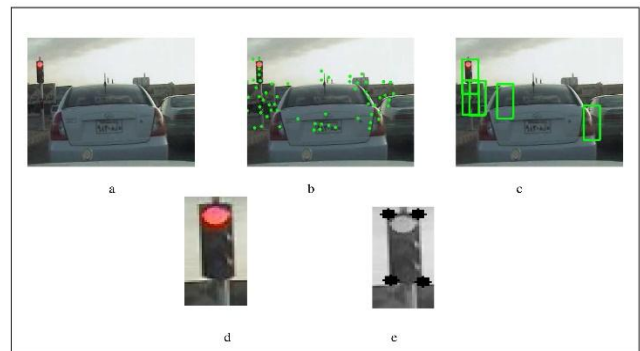


Fig. 10. Matching with using traffic light perspective: (a) Original image, (b) Corners are detected in the original image, (c) Set rectangleure for each detected corner, (d) The extracted window Base on the specified rectangular, (e) Detect the corner of the extracted window.

Now that positions of the traffic light's corners are known, the calculation of the traffic lights' width and height is started. To compute the width and height of the traffic light we must now the corresponding corners. The four detected corners are sorted by the x coordinate then the width and height are computed for each corresponding corners. We must examine the width and height to be sure that it is in the range of the regular traffic light which is in the specified distance. Then the ratio $(\Delta x / \Delta y)$ is calculated and compared to the specified ratio. The threshold is high because we assume that the traffic light is the only object in the window. The algorithm is

illustrated in Fig. 10.

This method is reliable. And while it will never yield an invalid result, this algorithm may miss a traffic light because some traffic lights can have many objects on the background which can affect the corner detection algorithm.

C. Color Classification

The color classifier receives a colored version of the selected region that is passed to the network test. This image is converted to HSV color space as Fig. 11(b) and filtered to delete all non-red color from the image. The resulting image is of only red and white colors as Fig. 11(c). The range of the red color is between 110 and 130. The histogram is calculated for this image; Fig. 11(d). The histogram is one dimensional with a size equal to 30 and 130 histogram bins range. The first two bins represent the red color. The two bins are checked to see if they lie within specific thresholds. If true, the detected shape is a traffic light. Otherwise, the detected shape is rejected.

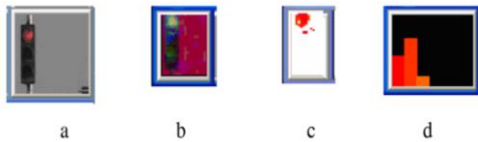


Fig. 11. Color classification stages: (a) Original image, (b) HSV color space, (c) Extracting red color, (d) Histogram.

V. SYSTEM TESTING

A. Testing the Neural Network

A set of video files were taken in different sites in Riyadh city, at distances ranging from 15m to 40m from camera, on different days and different times during the day. The set of tested files was 20 videos covering a total of 359 seconds at a rate of 25 frames per second. The system is tested by processing a frame using a sliding window at 38 and 42 step size in x and y directions, respectively. The processing was on two frames per second. There was no result by this technique. When we take 3 frames, there was a result but still inefficient. When reducing the step size of the sliding window to the half; i.e. 19 and 21, and taking one frame per second, the result is better, as shown in Table II.

The processing time for a frame was shorter in the first and the second technique, and increases in the third one. The following figures show the result of the recognition system on the three types of the traffic light:

• TL3 Traffic Light



Fig. 12. The result of the recognition system on TL3 traffic light: (a) Original image, (b) The sliding window, (c) Reducing the image and applying canny algorithm, (d) Extracting the red color only, (e) Calculate the histogram.

• TL2 Traffic Light

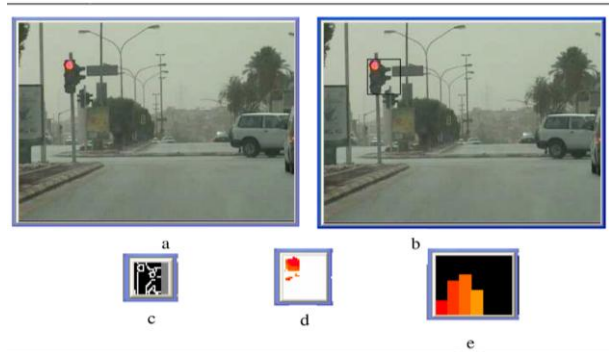


Fig. 13. The result of the recognition system on TL2 traffic light: (a) Original image, (b) The sliding window, (c) Reducing the image and applying Canny algorithm, (d) Extracting the red color only, (e) Calculate the histogram.

• TL1 Traffic Light

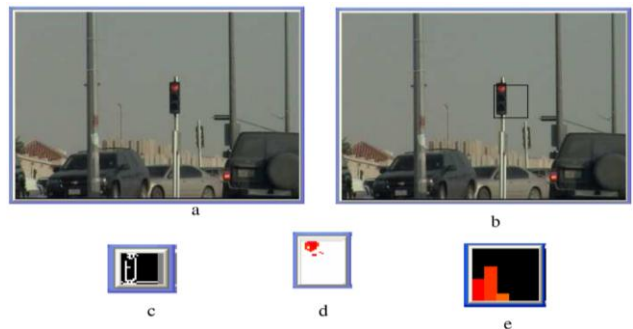


Fig. 14. The result of the recognition system on TL1 traffic light: (a) Original image, (b) The sliding window, (c) Reducing the image and applying Canny algorithm, (d) Extracting the red color only, (e) Calculate the histogram.

B. Testing the Template Matching

Template matching approach is tested over 12 video files with average length equals 6 second. The hit ratio is 83.52% with 0.3975 second average processing time for a frame. Here, the algorithm is tested over some of images captured from a video file, as shown in Table III.

TABLE II: NEURAL NETWORK TESTING

x	y	Frame/sec	Average time* in sec	Correct recognition of TL	Not TL	Percentage
38	42	2	0.22213	0	0	0%
38	42	3	0.23182	4	6	40%
19	21	1	2.23360	255	62	80.4%

* Average time to process one captured frame.

TABLE III: TEMPLATE MATCHING TESTING

Original image	Window establishing	Window extracted	Extracted window with four corner	Template sample	result
					Success
					Success

VI. CONCLUSION

A traffic light recognition system is a computer vision application. It can be integrated with other systems such as an

intelligent vehicle to make a powerful autonomous system. Currently, it is used to predict red traffic lights, since it is the essential risk factor in traffic accidents. Different methods for traffic light detection are investigated throughout this project. One of the most robust methods is based on edges as features of the traffic light and uses neural networks as a shape classifier. Then the color classifier is used to confirm the object. Although the system is assumed to be a real time application, the desired efficiency was not achieved. This is because the system scans a video frame, then regions are selected and each one has to enter the system classifiers. This process affects the program efficiency. However, a balance of sorts between reliability and efficiency was achieved.

SUSAN techniques is been used to process the frames more efficiently because its ability to find corners by direct analysis of the image intensities rather than their derivatives, thus removing the need for smoothing to reduce noise. A proposed method is based on detecting the corners of the image in order to select regions. Another one is to perform some of the operations; such as the resizing and Canny operator, on the whole frame rather than just the selected region. In addition, we have presented the architecture of shape classification system that utilizes template matching using perspectives of the traffic light. Using the perspectives of the traffic light will gave a powerful result. However, it failed under some conditions such as occlusion, low contrast and crowded backgrounds. It constitutes, however, a first and important step in the recognition of traffic lights. This method can be developed to cover all traffic light shapes and at any distance from the camera.

REFERENCES

- [1] Accidents statistics. [Online]. Available: <http://www.trafficsafety.org.sa/statistics.htm>
- [2] H. Fleyeh and M. Dougherty, "Road and traffic sign detection and recognition," *Advanced OR and AI Methods in Transportation*, 2003.

- [3] T. Zhuowen and L. Ron, "Automatic recognition of civil infrastructure objects in mobile mapping imagery using a Markov random field model," *ISPRS*, Vol. XXXIII, Amsterdam, 2000.
- [4] C. Y. Chung, W. J. Ming, and C. S. Wang, *A Vision-Based Traffic Light Detection System at Intersections*, 2002.
- [5] F. Lindner, U. Kressel, and S. Kaelberer, "Robust recognition of traffic signals," in *Proc. IEEE Intelligent Vehicles Symposium*, June 2004, pp. 49-53.
- [6] D. Eddowes and J. krahe, *Traffic Lights Recognition in a Scene Using a PDA*, 2003.
- [7] Edge detection. [Online]. Available: http://en.wikipedia.org/wiki/Edge_detection
- [8] B. Green. (2002). Edge detection tutorial. [Online]. Available: <http://www.pages.drexel.edu/~weg22/edge.html>
- [9] A. Dave and M. George, *Artificial Neural Networks Technology: A DACS State-of-the-Art Report*, New York: Kaman Sciences Corporation, 1992.
- [10] F. James and S. David, *Neural Networks: Algorithms, Applications, and Programming Techniques*, Addison-Wesley, 1991.
- [11] R. Lienhart and J. Maydt, "An Extended Set of Haar-like Features for Rapid Object Detection," *IEEE ICIP 2002*, vol. 1, 2002.
- [12] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2001.
- [13] P. Paclik, J. Novovi_cov_a, P. Pudil, and P. Somol, *Road Sign Classification Using the Laplace Kernel Classifier*.



Khalid A. Al-Shalfan currently holds several positions at Al-Imam Muhammad Ibn Saud Islamic University. He has been the dean of the Centre for Business and Entrepreneurship since 2010, the director of the Unit of Science and Technology since 2008, and director of the Funded Artificial Intelligence Program at the College of Computer and Information Sciences since 2007. He is also a member of several different scientific councils and committees at the university, including the Council of Research Deanship and Publishing Committee. He has a PhD in computer vision (computer science), from the University of Bradford in the UK. While doing his PhD, he worked on a project entitled "Investigation of 3-D Reconstruction from Uncalibrated Images". He also holds an MSc degree in real time electronic systems from the same university, and a BSc degree in computer engineering from King Saud University in Saudi Arabia.