International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 11 | Nov 2021 www.irjet.net p-ISSN: 2395-0072

Traffic Lights Detection Using Machine Learning

Rohith Reddy Vangala[1], Gurram Narasimharao[2], Jayanth Sattineni[3], Aamir Feroze Siddiqui[4]

Abstract - *Real-time detection of traffic lights from video feed has several real-life applications, including self-driving cars, automatic train signal detection, etc. This paper describes a method to detect red and green circular traffic lights from a video feed recorded by a camera mounted on a moving vehicle. In this approach, we first set thresholds for red and green in the RGB colour space and then identify blobs in the frame that meet this threshold. After blob detection, we perform a closing operation to eliminate gaps. We then identify the centre and radius of the circular blobs and draw a circle around the detected traffic signals. The program has been implemented on Matlab on an Intel i5 processor and tested several video samples to determine detection accuracy. Further, we also tested the performance of the system under noisy imaging conditions. The approach meets real-time processing requirements of 15 frames per second.*

***Key Words*: Real-time, traffic-lights, video-feed, blob-detection, closing-operation, Matlab, noisy-imaging conditions**

1. INTRODUCTION

Real-time detection of traffic lights from a video feed finds several applications in today's world. Most of the trains in Indian Railways today rely on the manual detection of the train signal, increasing the possibility of accidents. In this scenario, a system that automatically detects the traffic signal from the video feed and alerts the driver can improve road and railway safety. This system can also assist the driver in identifying traffic lights in the situation of poor visibility due to fog or other environmental conditions. Traffic lights are of different shapes, sizes, and colours. In this paper, we focus on the detection of red and green circular lights. The same approach can later be extended with slight modifications for traffic lights of other colours and shapes.

1.1 Challenges

First, differentiating many bright red-coloured and green-coloured objects in the video from traffic lights. Secondly, the environment in which the video is recorded also influences the accuracy of detection. The presence of fog or dust particles or low light conditions decreases detection accuracy, leading to more false negatives and false positives. Gaussian noise, which is introduced due to electronic components such as sensors, also reduces detection accuracy. Finally, there is also a need to process the frames in real-time and make critical decisions on accelerating or decelerating the vehicle. This might necessitate hardware acceleration by implementing the program on an FPGA.

1.2 Related Works

Existing literature on traffic light detection predominantly focuses on image processing and computer vision algorithms to detect traffic lights. Second, using machine learning to train the system to perform classification of detected traffic lights and improve performance. This paper focuses solely on Image Processing techniques for the detection of traffic lights.

2. IMPLEMENTATION & RESULTS

2.1. Approach

First Step: Detect blob to identify red and green traffic signals. To identify the blobs, we have used a threshold for the RGB colour values.

Second Step: Generate binary image from an RGB image with blobs having a value of 1 (represented as white regions of the image) and the rest having the value 0. Fig. 2. shows a binary image after performing colour thresholding on the original frame.

Fig. 1. Source image that is used as a reference for further demonstrations.

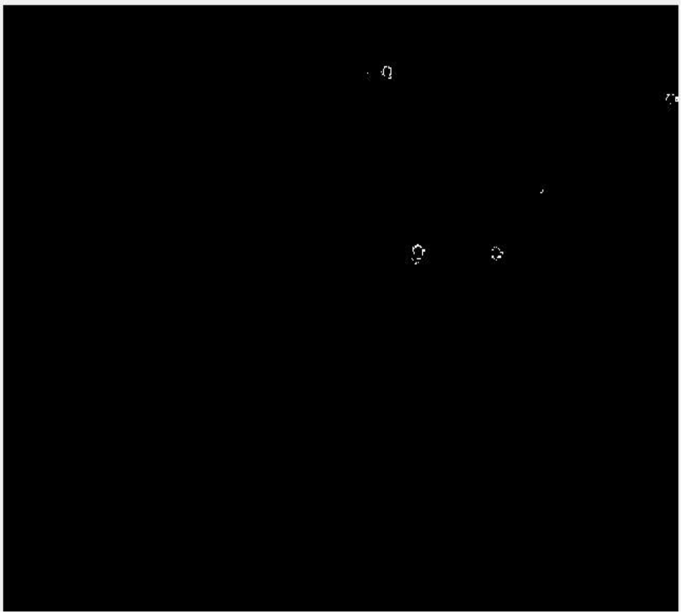


Fig. 2. Binary image based on detected blobs

Third Step: Perform a closing operation on this binary image to fill the gaps. The closing operation is a morphological operation in digital image processing which applies a dilation followed by erosion and eliminates gaps and holes. Fig. 3. shows the result after performing a closing operation on the binary image.

Fig. 3. Result of performing closing operation on binary image. Notice how the holes have been filled to form circular blobs

Fourth Step: Determine the outline or contour of these blobs. For this purpose, we determine the blobs' centre and radius and use this information to draw a circle around the detected blobs in the video. Fig. 4. shows frame with red colour traffic light detected and a cyan coloured circle drawn around it for ease of identification. Fig. 5. shows frame with green traffic lights detected and a red colour marking for ease of identification.

Fig. 4. Red traffic light detection

Fig. 5. Green traffic light detection

Fifth Step: Add Gaussian noise and apply an averaging filter to test the system's robustness under noisy imaging conditions. In Fig. 6. Notice the red circles indicating the large number of false positives detected after adding Gaussian Noise to the original red frame. Also, notice how the actual green lights are not detected. Fig. 7. Shows the result after applying an averaging filter. Notice how the number of false positives has decreased. The red markers at the centre top corner of the frame indicate that the green traffic signal has been detected.

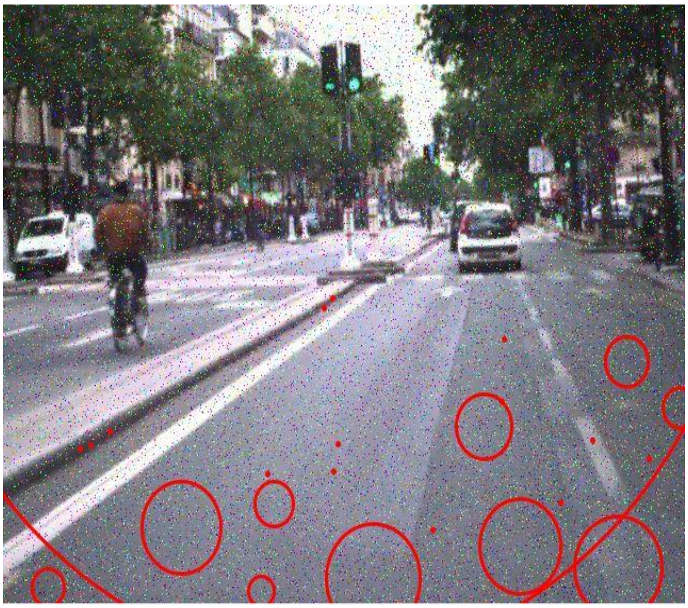
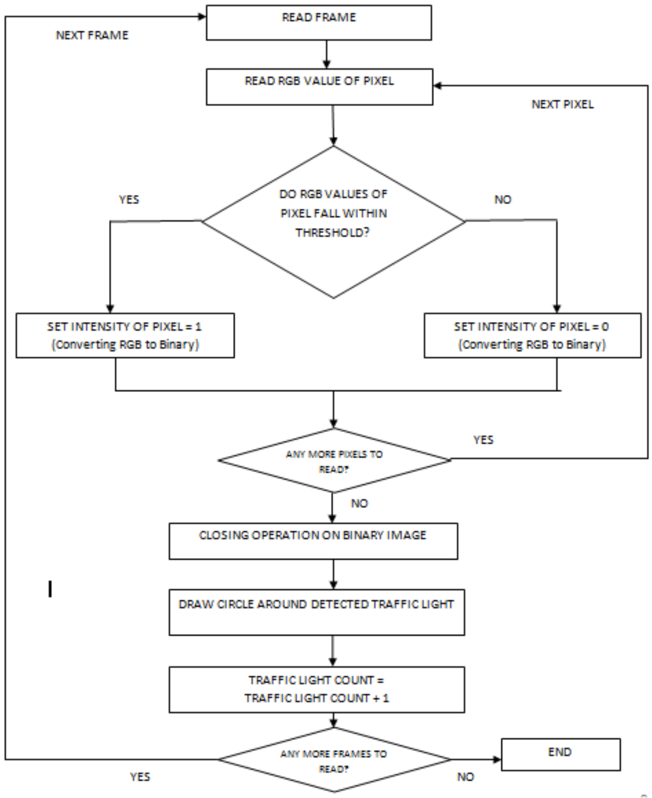


Fig. 6. Green Traffic light detection after adding Gaussian Noise to the original frame.

Fig. 7. Green light detection after filtering using Averaging filter.

2.2. Flow Chart

The below program has been implemented on an i5 processor.



3. ANALYSIS OF TESTING ON VIDEO SAMPLES (in no noise condition)

For the detection and recognition of traffic lights, two metrics - precision and recall are used.

Precision = true positives /(true positives + false positives)

Recall = true positives /(true positives + false negatives)

S No.	Num frames	Type of traffic lights	Num traffic lights	True positives	False negatives	False positives	Precision = TP/(TP+FP)	Recall = TP/(FN + TP)
1	185	-	-	0	-	105	0	0
2	455	RC*2	455*2 = 910	455	~400	~367	55.35%	53.21%
3	390	-	-	0	0	106	0	0
4	42	GC*2+ RC*1	42	0	42	42	0	0
5	83	RC*1+ GC*1	83	0	83	83	0	0
6	89	GC*1	0	0	0	77	0	0

Table 1 - Test result of 6 sequences that contain red traffic lights

RC is the number of circular red lights in the above table, and GC is the number of circular green lights. The second sequence shows a precision rate of 55.35% and a recall rate of 53.21%.

Common objects being identified as false positives include car headlights, road signs, hoardings, and other red objects.

S No.	Num frames	Type of traffic lights	Num traffic lights	True positives	False negatives	False positives	Precision = TP/(TP+FP)	Recall = TP/(FN + TP)
1	185	-	-	0	0	0	0	0
2	455	RC*2	455*2 = 910	0	0	25	0	0
3	390	-		0	0	12	0	0
4	42	GC*2 + RC*1	84	84	0	11	88.42%	100%
5	83	GC*1+RC*1	83	83	0	0	100%	100%
6	89	GC*1	89	0	89	89	0	0%

Table 2 - Test result of 6 sequences that contain green traffic lights

In the above table, the fourth sequence has 42 frames, and the type of traffic lights involved are two circular green lights and one circular red light, out which 84 circular green lights are being detected correctly along with 11 common objects as false positives, so the precision rate decreases to 88.42%. In the sixth sequence, actual green light is not being detected, but some other object continuously provides a false green light, resulting in a 0% precision rate. Common objects being identified as false positives include green signboards, hoardings that are green in colour.

4. SCOPE FOR IMPROVEMENT

Since the proposed algorithm is still a work-in-progress, we shall improve the accuracy in the future. One can use machine learning algorithms to classify detected lights and forecasting methods to improve performance. Template matching with a circular light inside can be used to identify the rectangular outline of the traffic signal template. As the vehicle moves through different roads, the traffic lights in these regions exhibit a wide variety in brightness and colour values, necessitating dynamic thresholding.

5. CONCLUSION

In this paper, we propose a system that can detect circular red and green traffic lights. Colour extraction and blob detection are used to identify the traffic signals. The program has been implemented on MatLab on an Intel i5 processor tested on several video samples, and the precision and recall calculated. The performance of the system was also tested under noisy imaging conditions. Computer vision and Image Processing toolbox of MatLab are being used to implement the design. The system has been found to meet the real-time processing requirement of 15 frames per second.

ACKNOWLEDGEMENT

We are thankful to Dr. Subhendu Kumar Sahoo for allowing us to work on this paper and for his continued support and guidance. We would also like to thank the Head of the Department of Electrical and Electronics Engineering Department, Prof. Y Yoganandam. We are also indebted to our friends and family for providing their invaluable support and help.

REFERENCES

1. Zhilu Chen, Xinming Huang, "Accurate and Reliable Detection of traffic lights using multiclass learning and multiobject tracking" in Proc.IEEE Intelligent Transportation Systems Magazine, 2016, pp. 28 - 42