Bachelor of Science in Computer Science & Engineering

Traffic Light Detection Using Modified K-means Clustering Algorithm for Color Blind People

by

Afzalul Alam

ID: 1504092

Department of Computer Science & Engineering

Chittagong University of Engineering & Technology (CUET)

Chattogram-4349, Bangladesh.

May, 2021

Traffic Light Detection Using Modified K-means Clustering Algorithm for Color Blind People

Submitted in partial fulfilment of the requirements for
Degree of Bachelor of Science
in Computer Science & Engineering

by

Afzalul Alam
ID: 1504092

Supervised by

Dr. Pranab Kumar Dhar
Professor
Department of Computer Science & Engineering

Chittagong University of Engineering & Technology (CUET)
Chattogram-4349, Bangladesh.

The thesis titled '**Traffic Light Detection Using Modified K-means Clustering Algorithm for Color Blind People**' submitted by ID: 1504092, Session 2019-2020 has been accepted as satisfactory in fulfilment of the requirement for the degree of Bachelor of Science in Computer Science & Engineering to be awarded by the Chittagong University of Engineering & Technology (CUET).

Board of Examiners

______________________________ Chairman

Dr. Pranab Kumar Dhar
Professor
Department of Computer Science & Engineering
Chittagong University of Engineering & Technology (CUET)

______________________________ Member (Ex-Officio)

Dr. Md. Mokammel Haque
Professor & Head
Department of Computer Science & Engineering
Chittagong University of Engineering & Technology (CUET)

______________________________ Member (External)

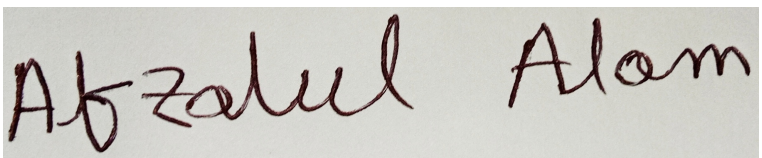
Dr. Mohammed Moshiul Hoque
Professor
Department of Computer Science & Engineering
Chittagong University of Engineering & Technology (CUET)

Declaration of Originality

This is to certify that I am the sole author of this thesis and that neither any part of this thesis nor the whole of the thesis has been submitted for a degree to any other institution.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. I am also aware that if any infringement of anyone's copyright is found, whether intentional or otherwise, I may be subject to legal and disciplinary action determined by Dept. of CSE, CUET.

I hereby assign every rights in the copyright of this thesis work to Dept. of CSE, CUET, who shall be the owner of the copyright of this work and any reproduction or use in any form or by any means whatsoever is prohibited without the consent of Dept. of CSE, CUET.



Signature of the candidate

Date: 30.05.2021

Acknowledgements

The success and final outcome of this thesis required a great deal of direction and assistance from a large number of people, and I consider myself quite fortunate to have received this by the end of my thesis. Professionally and personally, it has been a rewarding experience. All of what I've accomplished has been possible only because of such oversight and aid.

Above all, I want to express my gratitude to Allah for allowing me to finish this thesis. Following that, I'd like to offer my heartfelt gratitude to Dr. Pranab Kumar Dhar, Professor, Department of Computer Science Engineering, Chittagong University of Engineering Technology (CUET), for his advice, encouragement, and unwavering support during my thesis research. I appreciate his numerous probing inquiries, which forced me to consider things from many angles, as well as his unwavering support throughout the process. I appreciate all of the department's Teaching Staff's ongoing encouragement, support, and advice.

Finally, I want to express my gratitude to my father and mother for their unwavering love, support, encouragement, and assistance in all part of my life and academic career.

Abstract

Traffic Light(TL) detection is an important sector in computer vision and image processing. Traffic light detection is rapidly getting its high application in various sectors of computer vision. Especially in self driving cars its getting high attention. Several studies have been conducted in recent years in attempt to build an ideal TL detector that would overcome environmental problems such as lighting and other hurdles such as scale, appearance change, noise, and so on. Developing an ideal TL detector that meets both speed and accuracy requirements is difficult. Also, color blind people can't distinguish between red and green color that's why they can't correctly identify traffic signal. Being motivated by this issue we are proposing a framework of TL detection assembling modified k-means clustering algorithm, circle detection algorithm aided with contour detection method and circle filtering aided with BFS algorithm to achieve accurate detection result.

Keywords: Modified K-means clustering Algorithm, Contour detection, BFS algorithm, Color blindness.

Table of Contents

Acknowledgements **iii**

Abstract **iv**

List of Figures **viii**

List of Tables **ix**

1 Introduction **1**

1.1 Introduction 1
1.2 Traffic Light Detection Framework 2
1.3 Difficulties 2
1.4 Applications 3
1.5 Motivation 3
1.6 Contribution of the thesis 4
1.7 Thesis Organization 4
1.8 Conclusion 5

2 Literature Review **6**

2.1 Introduction 6
2.2 Related Literature Review 6
2.3 Conclusion 10

3 Methodology **11**

3.1 Introduction 11
3.2 Steps of Proposed Traffic Light Detection Method 11
3.3 Image Pre-processing 12
3.3.1 Scaling 13
3.3.2 Elimination of Area 13
3.3.3 Noise removal 14
3.3.4 Converting to HSV 14
3.4 Segmentation 15
3.4.1 Modified K-means clustering algorithm 15
3.4.2 Creating Three New image and Processing 16
3.5 Circle detection and filtering 18

3.5.1 Circle detection 18
3.5.2 Circle filtering 19
3.6 Applying voting on filtered circle 20
3.7 Comparison between traditional and modified K-means clustering 21
3.8 Conclusion 23

4 Results and Discussions 24
4.1 Introduction 24
4.2 Dataset Description 24
4.2.1 LISA traffic light dataset 24
4.2.2 WPI traffic light dataset 25
4.2.3 Bangladeshi traffic light dataset 25
4.3 Impact Analysis 27
4.3.1 Social and Environmental Impact 27
4.3.2 Ethical Impact 27
4.4 Evaluation of Framework 27
4.5 Traffic light detection in various condition 28
4.6 Traffic light detection when traffic sign present in image 37
4.7 Result Analysis 43
4.8 Comparison with Other Model 44
4.9 Conclusion 45

5 Conclusion 46
5.1 Conclusion 46
5.2 Future Work 47

List of Figures

1.1 Block diagram of Traffic Light Detection Framework. 2

3.1 Steps of the proposed detection method. 12
3.2 Overview of the Pre-processing steps. 12
3.3 Input image. 13
3.4 Image after scaling to 700x700. 13
3.5 After elimination of lower 40% Area. 14
3.6 After noise removal. 14
3.7 After converting to HSV image. 15
3.8 Red component of HSV image. 16
3.9 Green component of HSV image. 17
3.10 Yellow component of HSV image. 17
3.11 Binary representation of red component. 17
3.12 Binary representation of green component. 18
3.13 Binary representation of yellow component. 18
3.14 Filtered circle of red component. 19
3.15 Filtered circle of green component. 20
3.16 Filtered circle of yellow component. 20
3.17 After combing red, green and yellow component. 21
3.18 Displaying result after applying voting. 21
3.19 Result of segmentation using centroids (0,0),(0,300),(300,0). 22
3.20 Result of segmentation using centroids (0,0),(100,300),(300,500). . . 23

4.1 Sample images from LISA dataset 25
4.2 Sample images from WPI dataset 26
4.3 Sample images from Bangladeshi dataset 26
4.4 Input image. 29
4.5 Resized image. 29
4.6 After elimination of lower 40% area 30
4.7 After applying Gaussian filter. 30
4.8 After converting to HSV image. 31
4.9 Red component of HSV image 31
4.10 Green component of HSV image. 32
4.11 Yellow component of HSV image. 32
4.12 Binary image of red component. 33

4.13 Binary image of green component. 33
4.14 Binary image of yellow component. 34
4.15 Filtered circle of red component. 34
4.16 Filtered circle of green component. 35
4.17 Filtered circle of yellow component. 35
4.18 After combing filtered circle of red, green and yellow component. . 36
4.19 After applying voting on filtered circle. 36
4.20 Input image. 37
4.21 Image after scaling to 700x700. 37
4.22 After elimination of lower 40% Area. 38
4.23 After noise removal. 38
4.24 After converting to HSV color space. 38
4.25 Red component of HSV image. 39
4.26 Green component of HSV image. 39
4.27 Yellow component of HSV image. 39
4.28 Binary image of red component. 40
4.29 Binary image of green component. 40
4.30 Binary image of yellow component. 40
4.31 Filtered circle of red component. 41
4.32 Filtered circle of green component. 41
4.33 Filtered circle of yellow component. 41
4.34 After combining filtered circle of red, green and yellow component. 42
4.35 Displaying result after applying voting. 42

List of Tables

4.1 Confusion matrix for LISA dataset. 43
4.2 Confusion matrix for WPI dataset. 43
4.3 Confusion matrix for Bangladeshi dataset. 43
4.4 Our proposed model performance. 44
4.5 Comparison on LISA dataset 44
4.6 Comparison on WPI dataset 44

Chapter 1

Introduction

1.1 Introduction

The inability to detect colors in normal lighting conditions is known as color blindness. Some colorblind people are unable to see certain colors, while others perceive colors differently. Color blindness occurs when one or more sets of retinal cones that perceive color in light and transfer information to the optic nerve fail to mature properly. Color blindness can be caused by a variety of factors, including genetics, diabetes, aging, and drugs. Color blindness is an inherited condition that affects males more frequently than females. According to Prevent Blindness, an estimated 8 percent of males and less than 1 percent of females have color vision problems.

Because many people do not perceive color blindness to be a major problem, it is frequently overlooked. Being color blind, on the other hand, can be a major problem for a person's well-being and safety in everyday life, as well as for the safety of those around them when operating a vehicle. Red-green color deficiency is the most common form of color blindness. That's why most the color blind people cannot distinguish between the green and red light of the traffic signal. So, an automated traffic light detection system with high accuracy may help the color blind people to overcome this problem.

This chapter will provide an overview of the traffic light detection framework as well as the inherent problems of the assignment. This chapter will also explain the motivation for this thesis and how it contributes to the field.

1.2 Traffic Light Detection Framework

In the field of computer vision, Traffic Light Detection (TLD) is widely considered as a very popular and active research field. Traffic light detection is a process of finding traffic light position in an image, determining the color of the detected light in order to identify the state of the traffic light.

The main flow of work of the Traffic Light Detection framework can be discussed as-

1. Preprocessing steps are performed to get the generalized format of the captured image.
2. Image segmentation performed by modified K-means clustering algorithm.
3. Circle detection and filtering performed by contour detection and BFS algorithm.
4. Voting of filtered circles to select traffic light and identifying it's current state.

Block diagram of Traffic light detection framework is shown below:

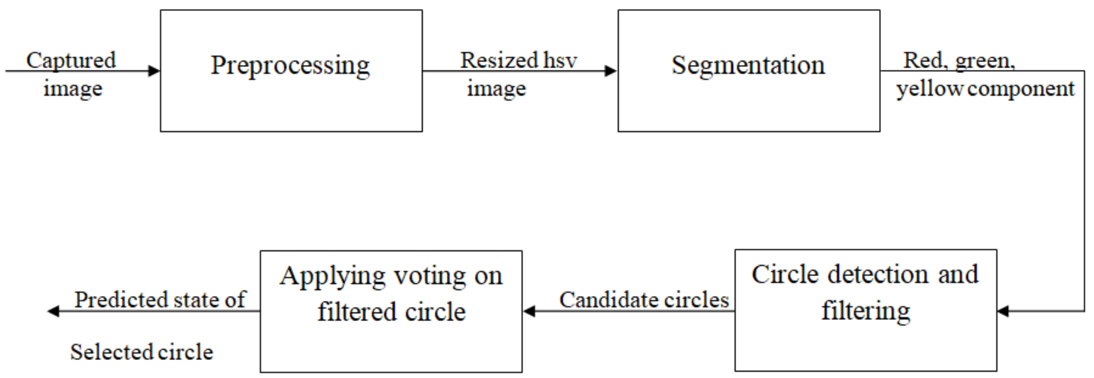


Figure 1.1: Block diagram of Traffic Light Detection Framework.

1.3 Difficulties

The task of traffic light detection is mainly based on the key features of the target's appearance model. At times, various conditions of the environment play a major role in the way these features are perceived. The major challenges in traffic light detection are enlisted as follows:

1. **Position, size and shape:** Traffic box position can vary region to region, some region it may be horizontal or vertical position. Size of the box also vary having traffic light ranging from one to three in the traffic box. Also, due to the different size of the traffic box shape of the traffic light may be of different shape.
2. **Illumination:** Image may be of different illumination due to the varying sunlight at different time of the day.
3. **Noise:** Noise may be added in the captured image. Especially at night noise can be of high magnitude.

1.4 Applications

Traffic light conveys specific instructions to the drivers of the vehicle on the road. This instructions are important for the drivers in order to avoid accident or traffic rules violation. Traffic light recognized from this process may be used for a wide range of applications, such as:

1. Helping color blind and visually impaired users to correctly identifying traffic signal and understanding its instructions.
2. Enabling self driving cars to correctly identifying traffic signal and taking proper actions.
3. Helping drivers who unconsciously break traffic light signal by providing them a warning message.

1.5 Motivation

With the advancement of technology many wondering product is being developed. Self driving vehicles is one of product which is getting huge market and attention of mass people. A self driving vehicles depends on computer vision for its core operation such as obstacle detection, lane detection, speed control etc. Traffic light detection is one of the important task that is performed by self driving vehicles. To avoid accident at intersections and traffic rules violation it needs a accurate traffic light detector.

Color blindness is defined as the inability of seeing colors in normal lighting conditions. Especially, they don't see the red and green color of traffic light. Color blindness is an inherited condition that affects males more frequently than females. Color vision disorders affect approximately 8% of males and less than 1% of females, according to Prevent Blindness.

Therefore, helping the color blind individuals to perceive the right color of the traffic light and assisting self driving vehicles to accurately detecting and identifying traffic light is the primary motivation of this thesis work.

1.6 Contribution of the thesis

Thesis or research work is done to meet a certain set of objectives, such as the quality of findings or the time it takes to process data. The thesis areas of contribution can be summarized as follows:

1. Segmentation of the image into three components using modified K-means clustering algorithm.
2. Filtering of detected circles in the image using fill percentage of circle and by defining a fixed window in the image.
3. Selection of target circle using voting from pre-calculated traffic box position.
4. We created a small dataset of traffic lights in Bangladesh.

1.7 Thesis Organization

The rest of this thesis report is organized as follows:

- Chapter 2 provides a synopsis of previous research in the topic of traffic light detection.

- The proposed methodology for detecting traffic lights is described in Chapter 3. In the proposed framework, segmentation is performed by modified K-means clustering algorithm. Circle in each segment are detected by contour detection and filtered using fill percentage and fixed window. For the classification process, voting from pre-calculated traffic box position is applied in this thesis.
- The working dataset is described in Chapter 4 along with an examination of the performance metric for the proposed framework.
- The overall summary of this thesis work is presented in Chapter 5, along with some future recommendations.

1.8 Conclusion

This chapter provides an overview of traffic light detection. The summary of the Traffic light detection framework is explained in this chapter, along with the problems. The inspiration for this work, as well as the contributions made, are also detailed here. The background and current situation of the problem will be discussed in the following chapter.

Chapter 2

Literature Review

2.1 Introduction

The aim of this study is to look at the difficulties that will be encountered when performing traffic light detection research and to include a thorough analysis of various detection approaches. This chapter addresses various detection methods used by different researches, as well as the results of these researches on various data, by offering a brief review of previous studies.

2.2 Related Literature Review

J Al-Nabulsi et al.[1] suggested a method for detecting traffic lights that involves comparing the candidate traffic light with some in-house collected traffic light models using correlation. The density thresholding technique is used to identify the green or red spots in the observed traffic light during traffic light detection. The other segments are discarded, leaving only the nominee green or red spots. One of the study's limitations is that when the traffic light model does not match the collected models, the light is discarded and no output is generated. In addition, no comparisons with other proposed methods to evaluate the performance of their suggested model were shown.

In R Kulkarni et al.[2] using transfer learning, a deep neural network-based model for accurate detection and recognition of traffic lights was proposed. For transfer learning, the approach employs TensorFlow's faster region-based convolutional network (R-CNN) Inception V2 model. The model was trained on a dataset containing various images of traffic signals classified into five categories in accordance with Indian Traffic Signals. The model achieves its goal by identifying the correct class type of traffic light. One of the study's flaws is that it need a lot of GPU resources to process the images. In addition, no comparisons with other proposed methods to evaluate the performance of their suggested model were shown.

D Yudin et al.[3] proposed a traffic light detector based on a fully convolutional neural network (FCNN) for segmenting traffic lights on images and subsequent clustering, allowing them to obtain traffic light bounding boxes. One of the study's flaws is that it need a lot of GPU resources to process the images. The data set used was small, with 500 training photos and 107 testing photos. In addition, no comparisons with other proposed methods to evaluate the performance of their suggested model were shown.

In PS Swami et al.[4] using canny edge detection and circular hough transformation, a system for detecting traffic lights was suggested. They used the technique to determine the form of a traffic light as well as the light present on the signal (red,green). They give a voice message to the blind person after they have been identified the light. The data set used was small, with only 70 images for validation purpose. The execution time is longer, ranging from 7.09 to 20.02 seconds per image. In addition, no comparisons with other proposed methods to evaluate the performance of their suggested model were shown.

A Ochoa et al.[5] suggested a Bat Algorithm and Data Mining-based Hybrid Intelligent Application. It analyzes images captured with a camera to classify regions of the traffic light. Three straight line equations that delimit the RGB space are used to classify the colors (red, yellow, and green) that are displayed in the traffic light. No comparisons with other proposed methods to evaluate the performance of their suggested model were shown.

SH Lee et al.[6] proposed a Haar-like feature-based algorithm for traffic light detection and recognition. They learned about the traffic light image using Haar-like features and then used the learning data to detect the candidate location. The pre-learned SVM(Support Vector Machine) classifier verifies the detected candidate image, and binarization and morphology operations are performed on the validated candidate image to detect the traffic light object. One of the study's flaws is that it only considered daytime images. In addition, no comparisons with other proposed methods to evaluate the performance of their suggested model were shown.

E Lee et al.[7] proposed a method that uses a deep neural network (DNN) to detect small traffic lights (TLs) in images captured by cameras mounted in vehicles. The proposed TL detector has a DNN architecture of encoder-decoder with focal regression loss; this loss function reduces loss of well-regressed easy examples. The proposed TL detector uses freestyle anchor boxes positioned at random positions in a grid cell of an input image to detect small objects near the grid cell's borders. One of the study's flaws is that it only considered daytime images. Furthermore, they just detected traffic box position and did not perform traffic light classification.

In X Li et al.[8] for on-vehicle camera applications, a robust traffic light recognition model based on vision information is proposed. Their contribution is primarily divided into three categories. First, the aspect ratio, area, location, and background of traffic lights are used as prior knowledge to minimize computational redundancy, resulting in a task model for traffic light recognition. Second, they suggest a set of improved methods based on an aggregate channel feature approach to improve accuracy, including changing the channel feature for each type of traffic light and creating a fusion detector structure. Third, they implement an inter-frame information analysis approach that uses previous frame detection information to change original proposal regions, improving accuracy even further. One of the study's flaws is that it only considered daytime images. Furthermore, they just detected traffic box position and did not perform traffic light classification.

Y Lu et al.[9] presented a novel paradigm that generalizes and uses a visual attention model to improve detection efficiency without sacrificing accuracy. The attention model is intended to produce a small number of candidate regions at a suitable scale in order to better identify and classify small targets. One of the study's flaws is that it only considered daytime images. Furthermore, it need a lot of GPU resources to process the images.

In Z Ouyang et al.[10] for the autonomous vehicle network, they suggested a lightweight, real-time traffic light detector. A heuristic candidate area selection module is used to find all possible traffic lights, and a lightweight Convolution Neural Network (CNN) classifier is used to classify the data. One of the study's flaws is that it require a lot of GPU resources to process the images. Furthermore, only considered daytime images.

JG Wang et al.[11] suggested a novel real-time approach that uses high dynamic imaging and deep learning to identify a traffic light. In their process, traffic light candidates are reliably identified in low exposure/dark frames and correctly classified in consecutive high exposure/bright frames using a deep neural network. In dark frames, this dual-channel mechanism can make full use of undistorted color and shape details, as well as the rich background in bright frames. To eliminate lights of different colors simultaneously in the dark channel, a non-parametric multicolor saliency model is proposed. To minimize the number of false positives in the bright channel, a multiclass classifier with a convolutional neural network (CNN) model is used. Incorporating temporal trajectory monitoring improves efficiency even further. One of the study's flaws is that it only considered daytime images.

In A Alam et al.[12] for traffic light detection and identification, a vision-based algorithm is proposed. The surrounding outdoor environment is captured using a monocular camera. The detection system is trained with the intensity features derived from the models of the traffic light. In the acquired image, similar features are searched in a predefined region of interest. Candidates that are highly compatible are considered suitable traffic light candidates. One of the study's flaws is that their dataset is small. Only 100 images was used to check the reliability and accuracy of the proposed system.

Z Chen et al.[13] demonstrated a novel approach that incorporates computer vision and machine learning techniques to accurately identify and classify different types of traffic signals, such as green and red lights in both circular and arrow forms. To find the candidates, color extraction and blob detection are used first. Following that, a multiclass classifier based on a pretrained PCA network is used to obtain frame-by-frame performance. Moreover, to avoid occasional misses, an online multiobject monitoring strategy is used, as well as a forecasting tool to rule out false positives. To boost the detector's efficiency and manage traffic light transitions, many additional optimization techniques are used.

2.3 Conclusion

A thorough literature review is highlighted in this chapter. The researchers' feature extraction techniques and classifiers are covered in this chapter. The proposed technique for traffic light detection is fully explained in the following chapter.

Chapter 3

Methodology

3.1 Introduction

Traffic light detection is a challenging task when occluding elements are present and the lighting changes. Furthermore, when background objects superimpose on our target object, a robust prediction of the target object's prediction is required.

Because of these factors, a model based on the modified K-means clustering technique has been presented, which takes advantage of the target object's color and texture qualities. This chapter goes through the many phases of the proposed approach in great depth.

3.2 Steps of Proposed Traffic Light Detection Method

Figure 3.1 depicts the steps of the suggested detection approach, which aids in the accurate detection of the target object. The image from which the regions of interest are to be detected is used as input in the preprocessing step. Then, image is resized to predefined size, lower 40% area is eliminated, noise is removed and image converted to HSV color space.

In the segmentation stage, HSV image is used as input. This image is divided into red, green and yellow cluster. Then, three image is created from cluster pixels value and each image is converted to binary image.

In the circle detection part, three binary image is used as input. All circles is detected and filtered. Using filtered circle, a single image is created in this step.

In the recognition step, voting is applied on the filtered circle from pre-calculated traffic box position. Circle with maximum vote is the desired traffic light.

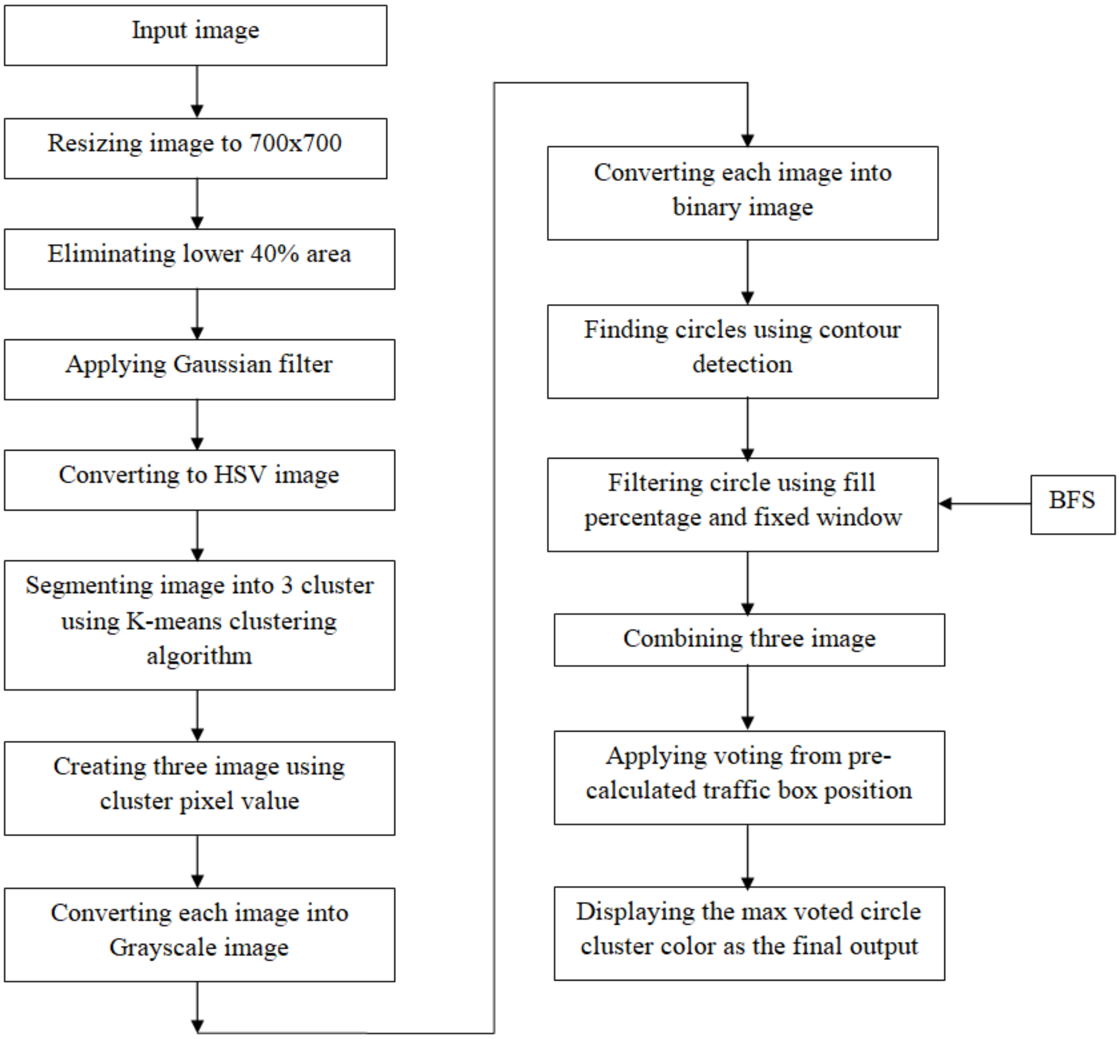


Figure 3.1: Steps of the proposed detection method.

3.3 Image Pre-processing

The first phase in the Traffic Light Detection system is pre-processing. It is important to ensure that the photographs taken are of high quality. These steps are shown below-

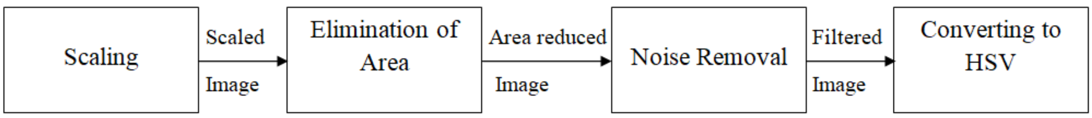


Figure 3.2: Overview of the Pre-processing steps.

3.3.1 Scaling

The resizing of a digital image's height and width is known as image scaling. Each input in a Traffic Light Detection system is converted into an image. Since different cameras capture images in different sizes, the size of the image will vary from one to the next. The extracted function vector's size is proportional to the image's size. When larger images are fed into feature extraction methods, the computational cost rises.

Scaling down an input size is used in this project. It's common to lose visual information when scaling down. A standard size is used as a parameter to achieve the trade-off between computation cost and image detail. All of the collected images have been scaled to a matrix of 700x700 pixels. The effect of image scaling is shown in Figure 3.4.

Figure 3.3: Input image.

Figure 3.4: Image after scaling to 700x700.

3.3.2 Elimination of Area

In this step lower 40% area of scaled image is eliminated, as traffic light always reside in upper area of image. As, a result computation time decrease because pixels number reduce. The effect of elimination is shown in Figure 3.5.

Figure 3.5: After elimination of lower 40% Area.

3.3.3 Noise removal

Noise is a change of intensity values in an image that distorts the original image data. It is an unwanted piece of data that can be affected by a number of factors. It is important to eliminate noise in order to improve clarity and ensure proper information extraction.

Gaussian noise, salt-and-pepper noise, impulsive noise, and other forms of noise are among them. When acquiring an image of a traffic light, noise added in the image. We used Gaussian filtering to remove noise because it is effective when captured image has lot of noise and especially captured in low light. Gaussian filtering can mute that noise. The effect of noise removal is shown in Figure 3.6.

Figure 3.6: After noise removal.

3.3.4 Converting to HSV

RGB color space in sensitive to light. Due to variation of light a color may be look different from its original color. But HSV is insensitive to light, as a result it can tolerate illumination change and holds original color under various lighting condition. For this reason, we convert RGB to HSV color space for correctly detecting red, green and yellow color from the image. The effect of converting to HSV color space is shown in Figure 3.7.

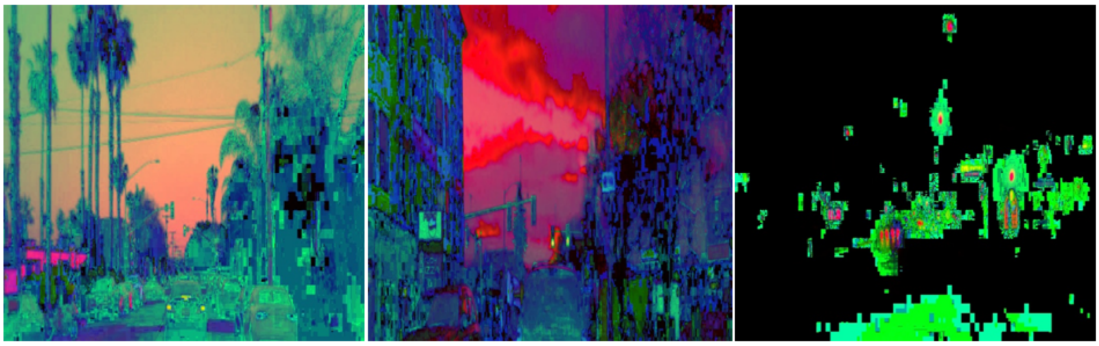


Figure 3.7: After converting to HSV image.

3.4 Segmentation

The proposed framework's second stage is segmentation. Image segmentation is the method of partitioning a digital image into several segments, such as sets of pixels, also known as image objects, in digital image processing and computer vision. The aim of segmentation is to make an image more meaningful and easier to interpret by simplifying and/or changing its representation. Objects and boundaries are usually located using image segmentation.

3.4.1 Modified K-means clustering algorithm

The unsupervised modified K-means clustering algorithm is used to separate the interest area from the context. Based on the K-centroids, it clusters or partitions the given data into K-clusters or bits.

Modified K-means clustering algorithm based segmentation

Step 1: Choosing the number of clusters K. We used K=3 for segmentation image into 3 cluster.

Step 2: Initialization of fixed K points, the centroids.

Step 3: Calculating color distance of each pixel from K=3 centroid using Euclidean distance. The formula given below:

$$distance, d = \sqrt{(x_i - x_k)^2 + (y_i - y_k)^2 + (z_i - z_k)^2} \tag{3.1}$$

where,

$$x_i, y_i, z_i = Pixel\ Hue,\ Saturation,\ Value\ (HSV)\ value$$

$$x_k, y_k, z_k = Centroid\ Hue,\ Saturation,\ Value\ (HSV)\ value$$

Step 4: Finding smallest distance from centroid, if it is within the threshold assigning it to the closest centroid, else discarding the pixel.

Step 5: Calculating the new centroid, repeating step 3 to 5 until centroid value became constant.

3.4.2 Creating Three New image and Processing

In this step, we are creating three new HSV image from the segmented image pixels. This image are red, green and yellow representing component of this color in the original image. Then, three HSV image is converted to gray-scale image. This grayscale image is converted to binary image using Otsu automatic thresholding algorithm.

Figure 3.8: Red component of HSV image.

Figure 3.9: Green component of HSV image.

Figure 3.10: Yellow component of HSV image.

Figure 3.11: Binary representation of red component.

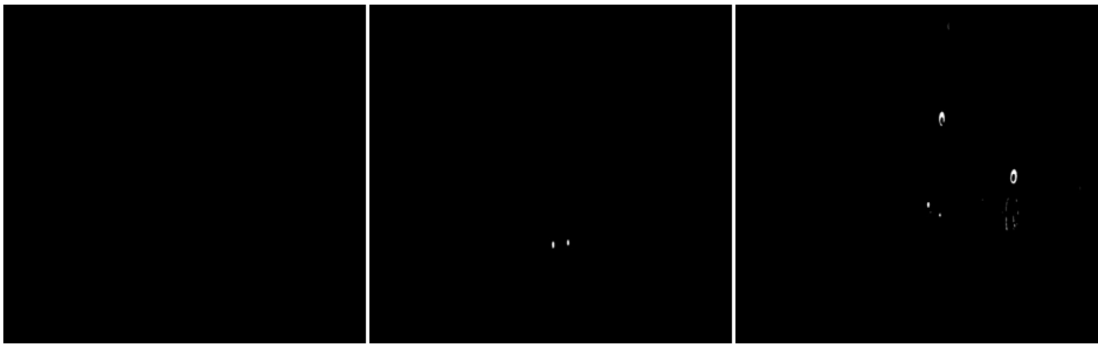


Figure 3.12: Binary representation of green component.

Figure 3.13: Binary representation of yellow component.

3.5 Circle detection and filtering

The proposed framework's third stage is circle detection and filtering. Many image processing techniques include automatic circle identification as a key component. In a two-dimensional space, a circle can be described by:

$$(x - a)^2 + (y - b)^2 = r^2 \tag{3.2}$$

where (a,b) is the center of the circle, and r is the radius. In this step, we have to find center (a,b) and radius r of the circle.

3.5.1 Circle detection

Circle can be detect using several technique. Most frequently used method are Hough Circle Transform and Contour detection. In Hough circle transform we have to do tuning in order to detect all circles in the image. Without proper tuning it may have missed small circles, which results in omission of target circle. But using Contour detection we can overcome this issue and every circle like object can be detected. As, it detect all circles like object we have to apply filtering to reduce circle number to minimize computational cost. Filtering described in the next subsection.

3.5.2 Circle filtering

Filtering is technique to find the best candidate from the given set of data which have potential to become the required output. We used BFS algorithm and fixed window to eliminate non-candidate circles.

Circle filtering using BFS algorithm and fixed window

Step 1: If circle is outside of defined boundary, then discard the circle and go to step 6.
Step 2: Find all the pixel that is inside the circle boundary using BFS algorithm.
Step 3: Calculate number of white pixel among them. Then, calculate fill percentage using the formula:

$$Fill\ percentage = (\ no\ of\ white\ pixel\ /\ total\ pixel\ )\ *\ 100 \tag{3.3}$$

Step 4: If fill percentage is less than 80% discard the circle and go to step 6.
Step 5: Add the circle in the filtered circle list.
Step 6: Stop and return.

Figure 3.14: Filtered circle of red component.

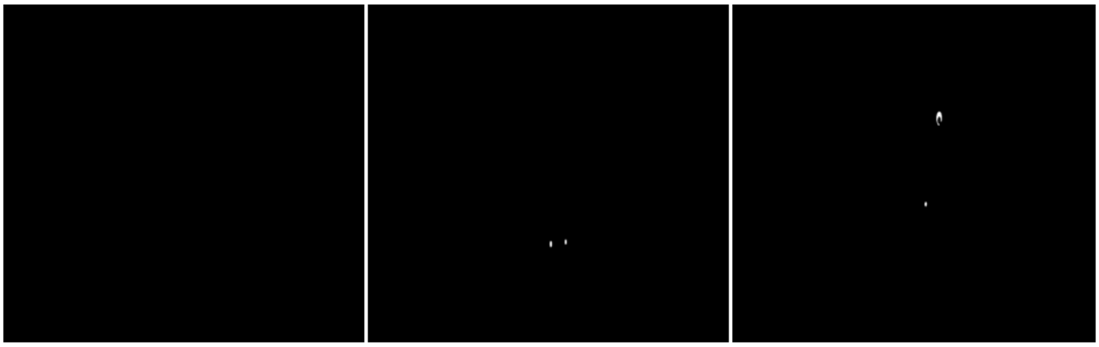


Figure 3.15: Filtered circle of green component.

Figure 3.16: Filtered circle of yellow component.

3.6 Applying voting on filtered circle

This is last step in our proposed framework. In this step, we find the circle that is the formed by traffic light and recognize of color of that light. This step can be called as detection and classification step in our framework.

Voting procedure

Step 1: All the filtered circle from red, green and yellow segment is combined into a single image along with circle segment color.
Step 2: Applying voting from pre-calculated traffic box position, circle which is completely lies inside the traffic box get the vote for that traffic box position.
Step 3: After voting sorting the circle in descending order according to acquired vote.
Step 4: Circle with max vote is the desired circle formed by traffic light. Marking its position and writing its color in the input image.

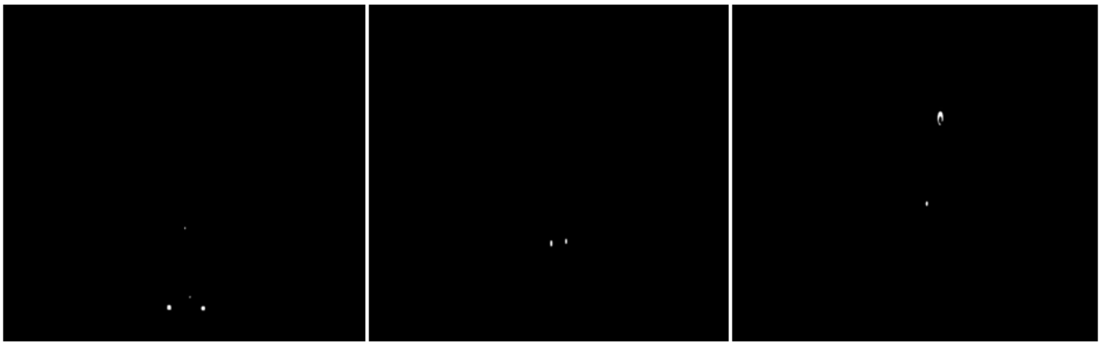


Figure 3.17: After combing red, green and yellow component.

Figure 3.18: Displaying result after applying voting.

3.7 Comparison between traditional and modified K-means clustering

The main difference between traditional and modified K-means clustering is given below:

1. Traditional consider pixel position on 2D grid for segmentation. But modified consider pixel HSV color value for segmentation.
2. Traditional methods do not delete any pixels; however, modified methods only consider pixels that are red, green, or yellow in color and ignore the rest.
3. Traditional image segmentation does not lower picture complexity; it merely divides it into k segments. However, the visual complexity has been reduced, and only useful information has been retained in modified k-means clustering.

The result of segmentation will be dependent on the initial centroids value if we used traditional k-means clustering in our proposed methodology with K=3. The result of image segmentation using traditional K-means clustering with varying initial centroids shown in figure 3.19 and 3.20 .

Figure 3.19: Result of segmentation using centroids (0,0),(0,300),(300,0).

Figures 3.19 and 3.20 show that, based on the initial cluster value, each image will be segmented into the same cluster. As a result, we are unable to retrieve precise information using the traditional k-means clustering approach. A classifier is also required to detect the color of the circle in the image. As a result, traditional k-means clustering will not work with our method.

As a result, we may conclude that modified k-means clustering improves segmentation while eliminating the necessity for a classifier to determine traffic light state.

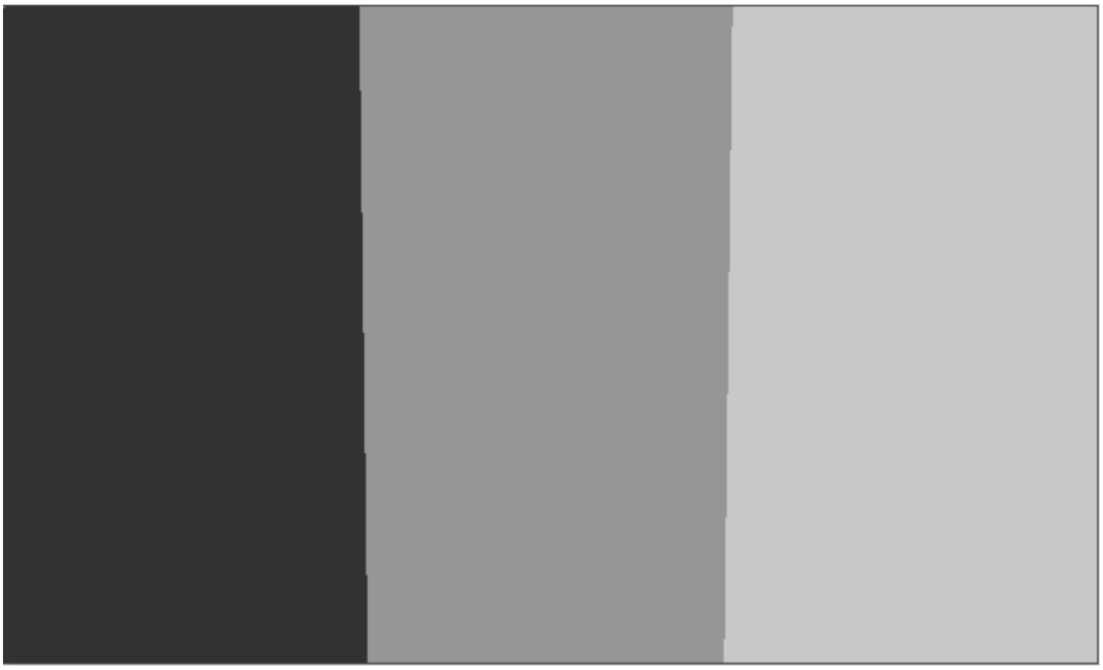


Figure 3.20: Result of segmentation using centroids (0,0),(100,300),(300,500).

3.8 Conclusion

A approach for traffic light detecting framework is discussed in this chapter. The modified K-means clustering technique is used for segmentation. The segmented image was used to detect and filter circles. In terms of classification, a voting mechanism is implemented. The proposed framework's experimental results are examined in the following chapter.

Chapter 4

Results and Discussions

4.1 Introduction

A detailed overview of the overall detection mechanism and its different components was described in the previous chapter. This chapter would address the success of the proposed detection system on different datasets. The primary goal of this project is to see how the proposed system approaches traffic light identification in a real-world situation.

The suggested procedure was written in Python and image manipulation was accomplished with OpenCV. The deployed code was run on a computer with an Intel Core i5 processor and 8GB RAM.

4.2 Dataset Description

We have used three datasets to test the proposed traffic light detector framework performance. They are given below:

1. LISA traffic light dataset also known as VIVA dataset.
2. WPI traffic light dataset.
3. Bangladeshi traffic light dataset collected by us.

4.2.1 LISA traffic light dataset

The LISA traffic light detection benchmark [14] (also called VIVA) is publicly available dataset that includes traffic light data gathered during the day and night in the cities of La Jolla and Pacific Beach in San Diego, California, USA. Totally, there are 10,956 daytime and 11,520 nighttime images at a resolution of 1280*960. Currently, we are using 3,500 images of daytime and nighttime for testing. Sample images from the dataset given in Figure 4.1.

Figure 4.1: Sample images from LISA dataset

4.2.2 WPI traffic light dataset

The WPI traffic light dataset (WPI) is a public dataset [13] released by Worcester Polytechnic Institute researchers, and gathered in Worcester, Massachusetts, USA, over the summer and winter. The original dataset contains 3,480 images at a resolution of 1920*1080. We used 2436 images for traning and 1044 images for testing. Sample images from the dataset given in Figure 4.2.

4.2.3 Bangladeshi traffic light dataset

We have captured 130 images of Bangladeshi traffic light. This images is captured at Tiger pass and Wasa circle of Chittagong. We have used 100 images for training and 30 images for testing. Sample images from the dataset given in Figure 4.3.

Figure 4.2: Sample images from WPI dataset

Figure 4.3: Sample images from Bangladeshi dataset

4.3 Impact Analysis

Impact analysis is an evaluation procedure that aims to give scientifically reliable evidence to support the presence of a service or the implementation of an intervention that aims to make a difference or give a benefit. Impact analysis is, in essence, a means of assessing results in order to answer the question, "Are we making a difference?"

4.3.1 Social and Environmental Impact

The following are the social and environmental implications of our suggested Traffic Detector methodology:

1. It will enable color blind people to correctly perceive traffic light signal. Hence, increases safety of color blind people and other people on the road.
2. It will help unconscious driver to obey traffic signal by giving warning message. Hence, reduce accidents probability at intersection.
3. Because the entire system is powered by electricity, no pollution will be produced. It also uses little power to operate. Hence, it is environment friendly.

4.3.2 Ethical Impact

Ethical issues occur in research at every level of the process, from design to reporting. Anonymity, secrecy, and informed consent are among them. This research was done covering ethical codes like honesty and integrity, carefulness, respect for intellectual property, confidentiality. Hence, it is ethically compliance.

4.4 Evaluation of Framework

For this thesis work, captured images of traffic light were considered as input. Each sample image was propagated throughout the pre-processing steps. Captured images were resized. Lower 40% area of captured image was eliminated.

Gaussian filter was applied to remove noise. Image was converted from RGB to HSV color space.

In segmentation, HSV image is propagated through series of steps. First, HSV image is segmented into red, green, and yellow component using modified K-means clustering algorithm. Using the component pixels value three new HSV image generated containing respective pixels value. Each new HSV image is converted to grayscale image. Then, each grayscale image converted to binary image using Otsu automatic thresholding algorithm.

In circle detection and filtering, circles was detected using contour detection. Then, filtering technique aided with BFS algorithm and fixed window used to select best circle among detected circles.

In voting steps, filtered circle from red, green and yellow component combined into single image. Then, voting from pre-calculated traffic box position begin if the circle lies inside the traffic box it get a vote. Finally, circle with most vote selected as traffic light and result is displayed in the input image.

4.5 Traffic light detection in various condition

We used our proposed framework to process 18 images of various conditions from the LISA, WPI, and Bangladeshi traffic light datasets. There are 12 daytime images and 6 nighttime images among the total of 18 images. Figures 4.4 to 4.19 display the results of each step of our framework on the input images.

The circle formed by the traffic signal is too tiny in the input images (a), (b), (c) and (e) of figure 4.4. The most difficult part is detecting these little circles. Hough circle transform will fail to detect these small circles without proper tuning, and tuning is a time-consuming job. Contour detection, on the other hand, can recognize any circle-like object without the need for any tuning. As a result, we overcome this challenge by employing contour detection to recognize circles.

The biggest problem for the input images (m), (n), (o), (p), (q), and (r) in figure 4.4 is those taken at night. As a result, the image has a lot of noise, street lights, car tail lights, and business sign board lights. In comparison to daytime, this light produces more Circles. We solve this issue by using a filtering mechanism to eliminate non-candidate circles. Using the BFS algorithm, the filtering approach determined the fill percentage of each circle and eliminated circles with fill percentages less than 80%. Additionally, the circle outside the defined window was removed.

The difficulties for the remaining photos were minor, and our proposed method handled them well.

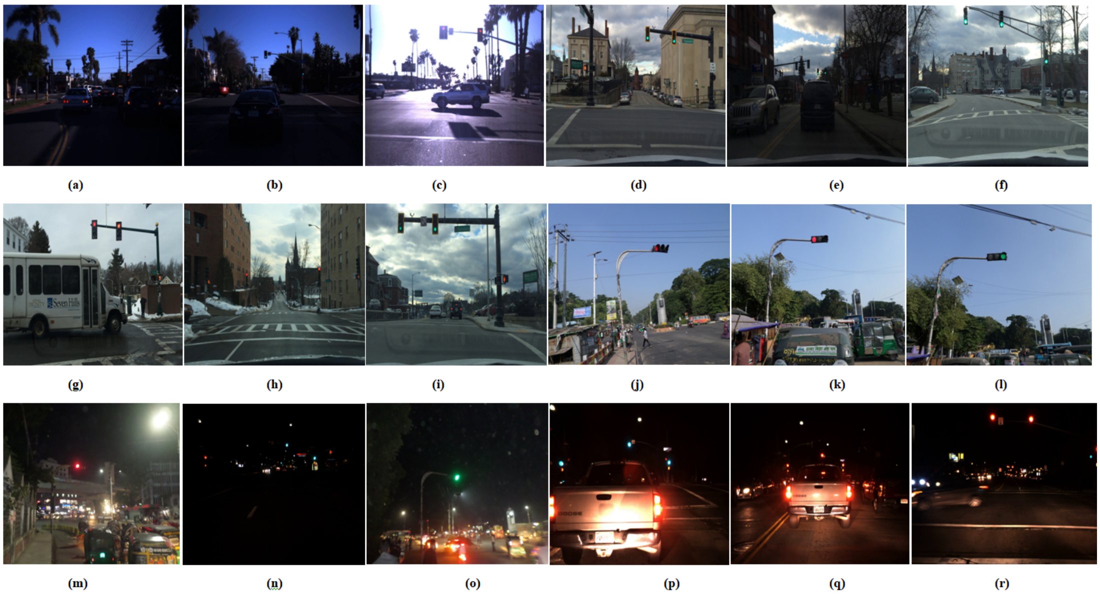


Figure 4.4: Input image.

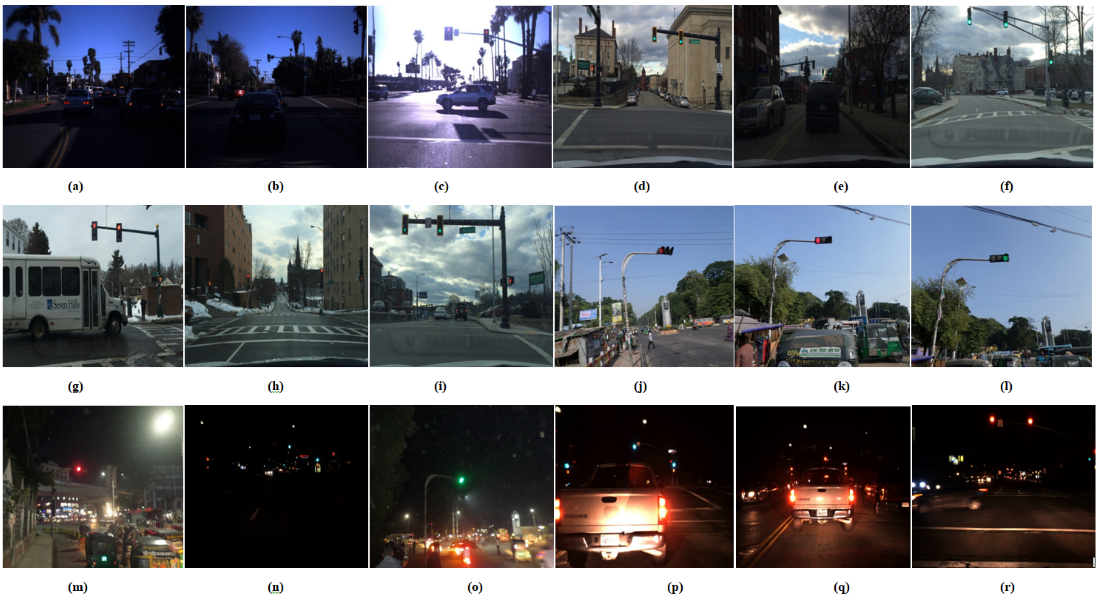


Figure 4.5: Resized image.

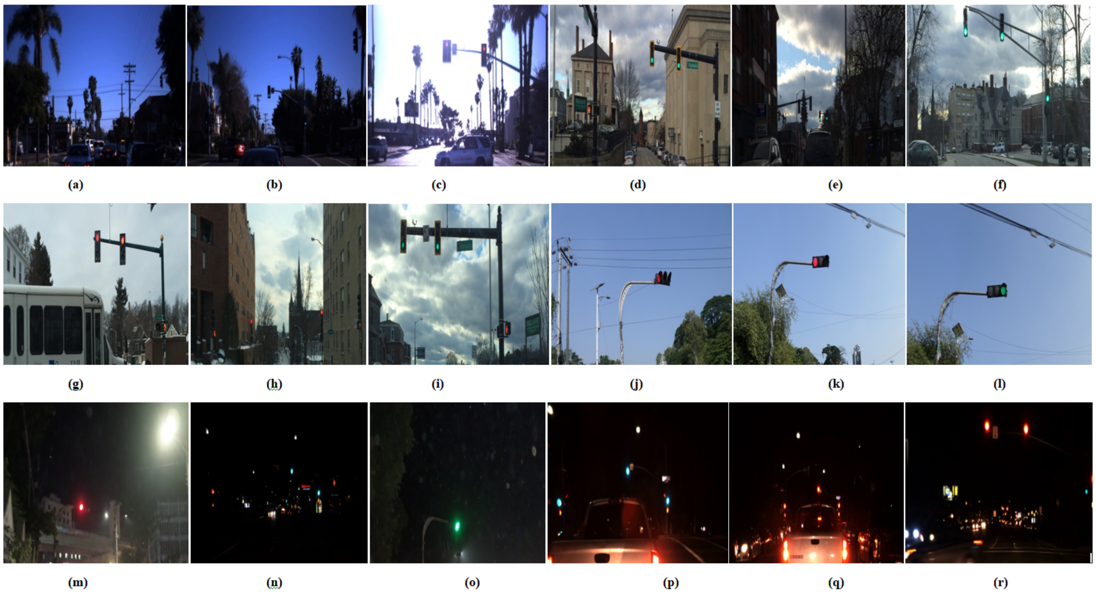


Figure 4.6: After elimination of lower 40% area .

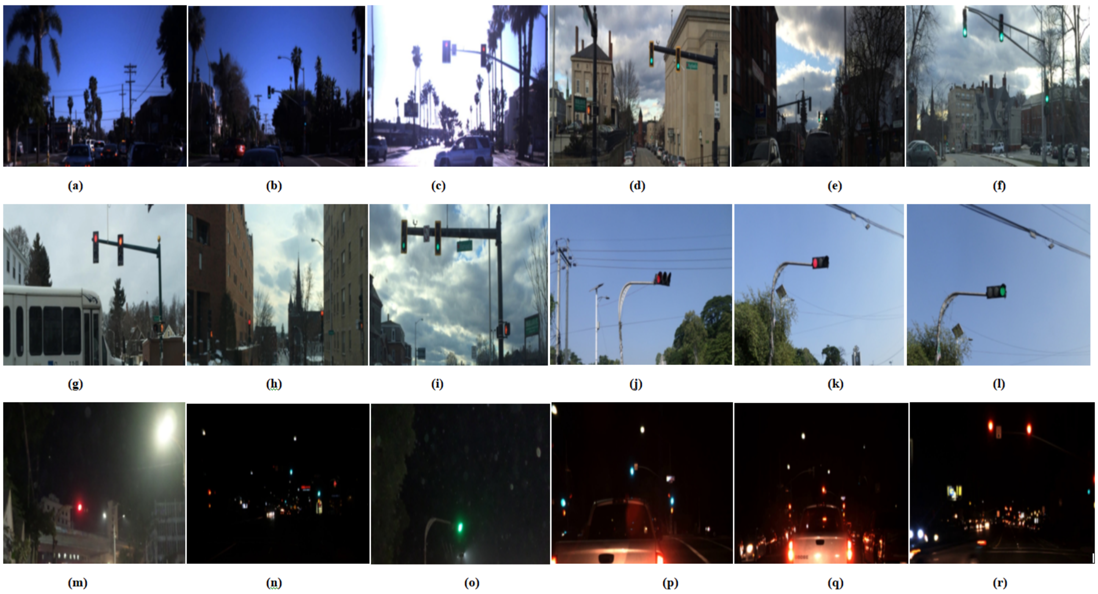


Figure 4.7: After applying Gaussian filter.

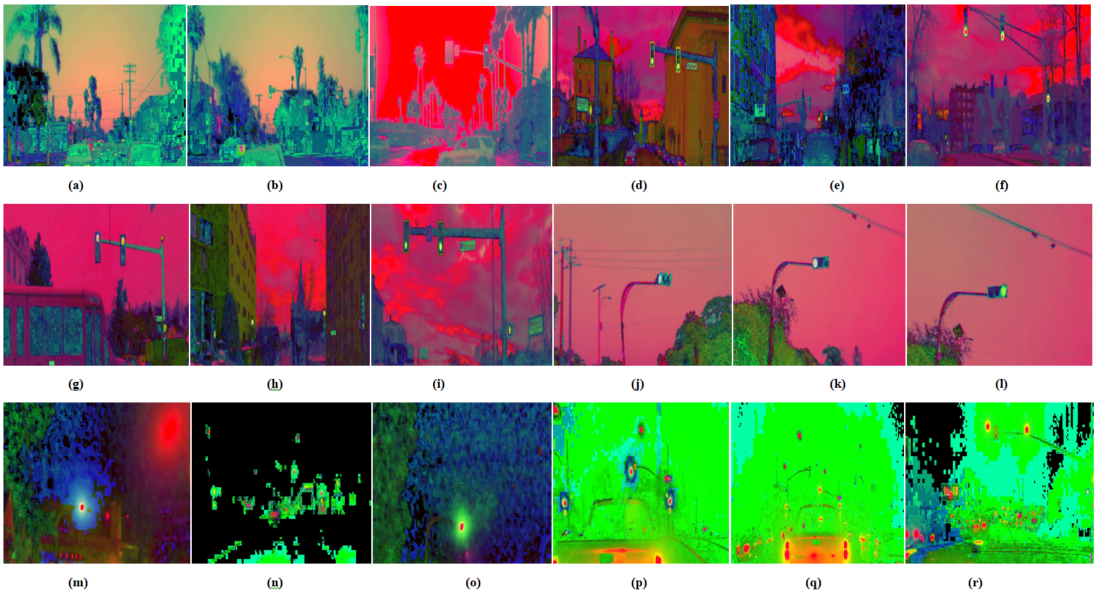


Figure 4.8: After converting to HSV image.

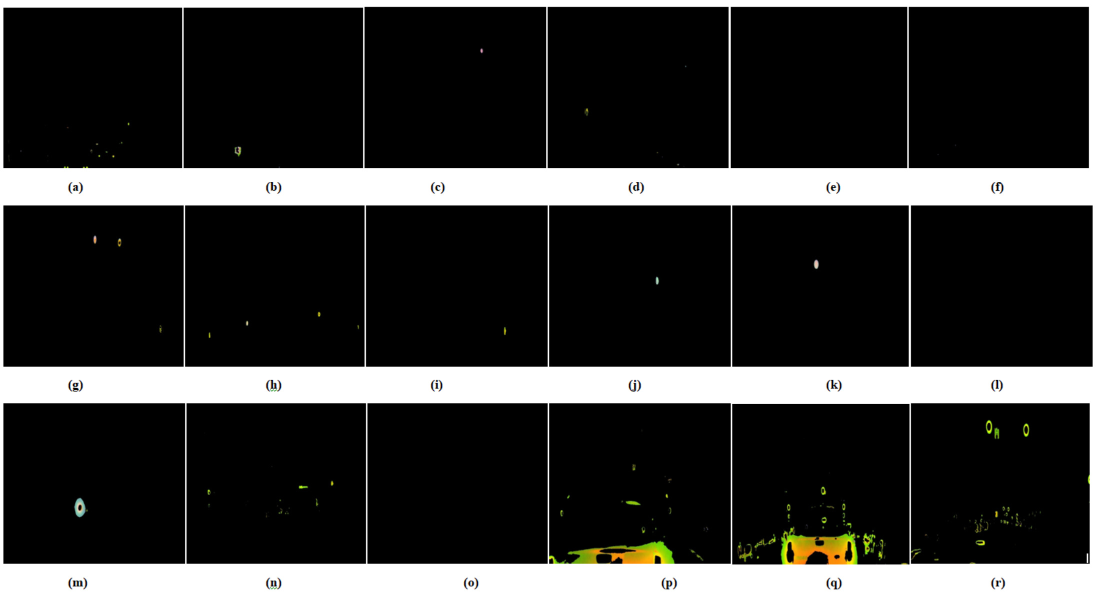


Figure 4.9: Red component of HSV image

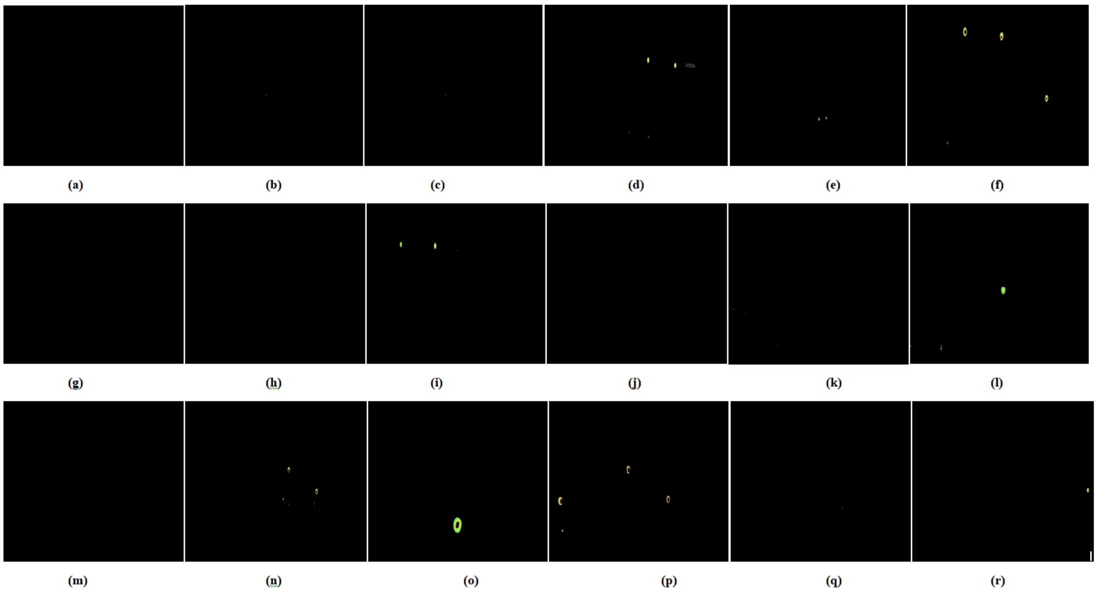


Figure 4.10: Green component of HSV image.

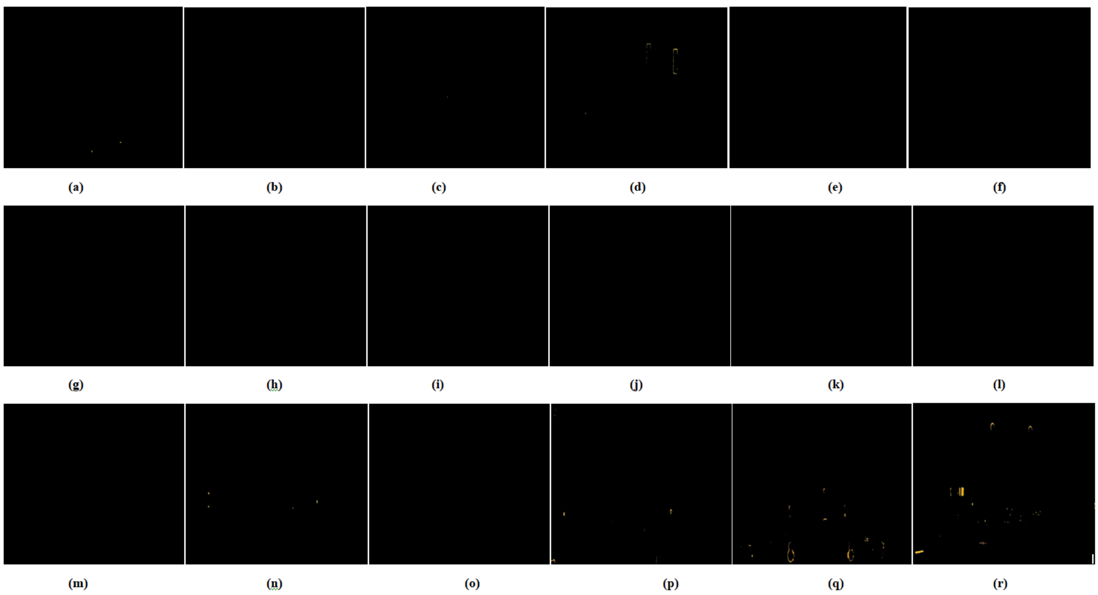


Figure 4.11: Yellow component of HSV image.

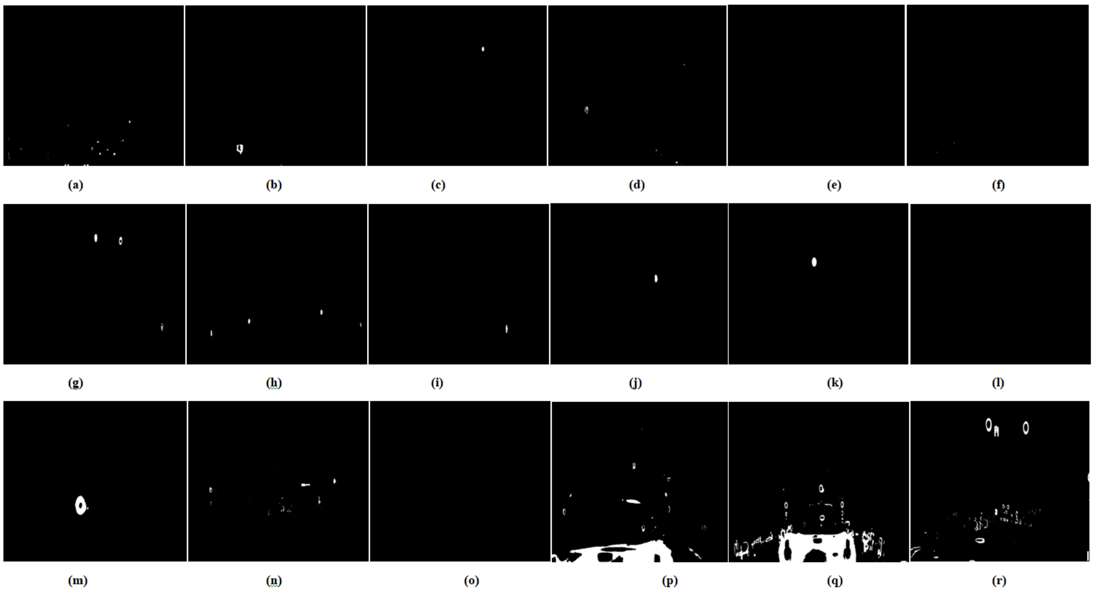


Figure 4.12: Binary image of red component.

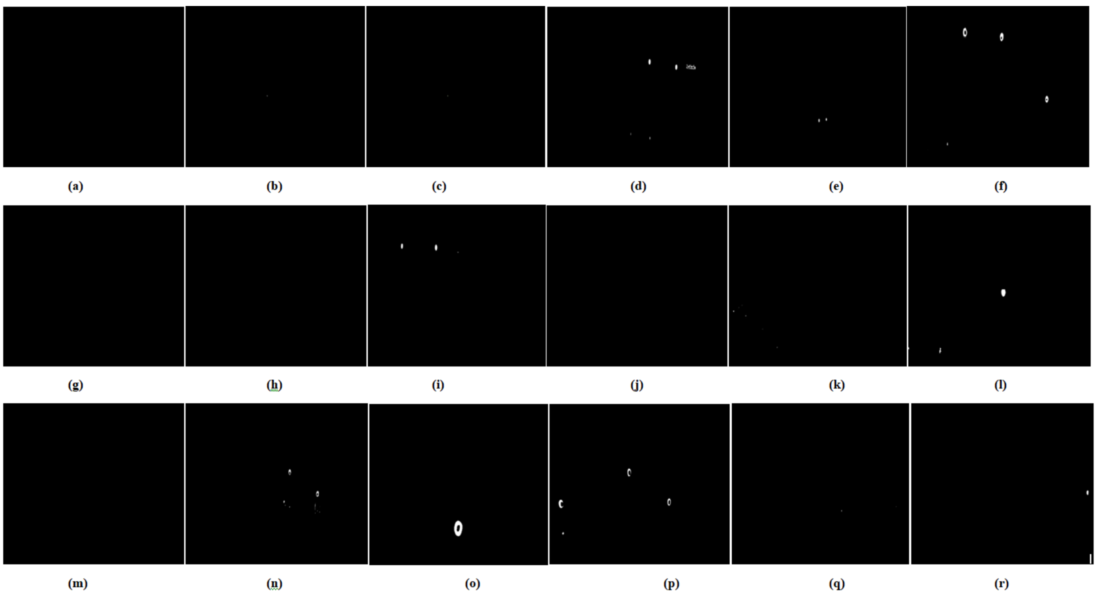


Figure 4.13: Binary image of green component.

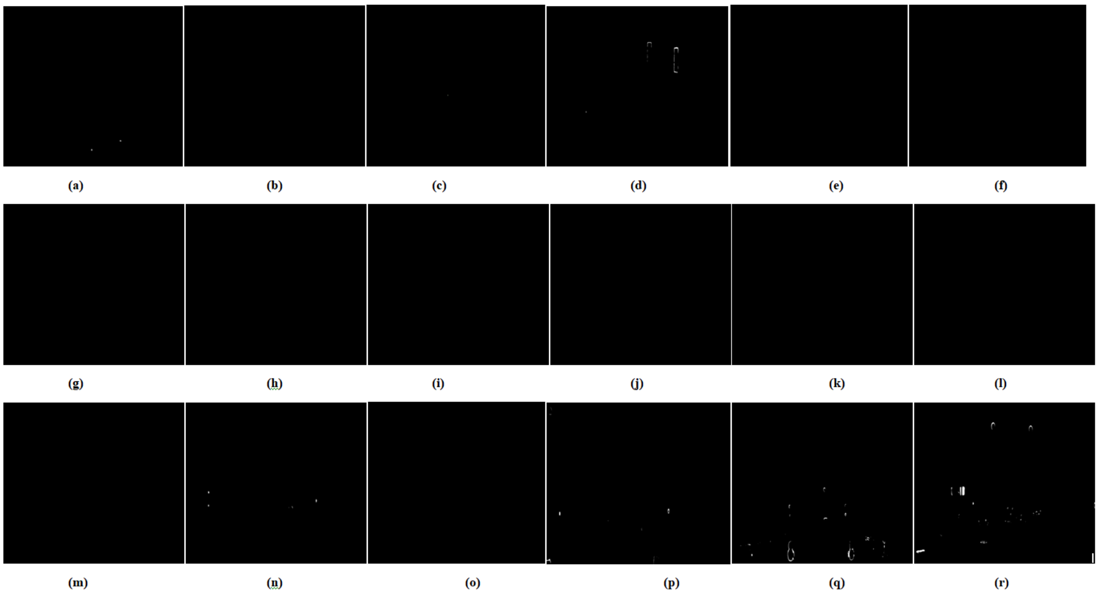


Figure 4.14: Binary image of yellow component.

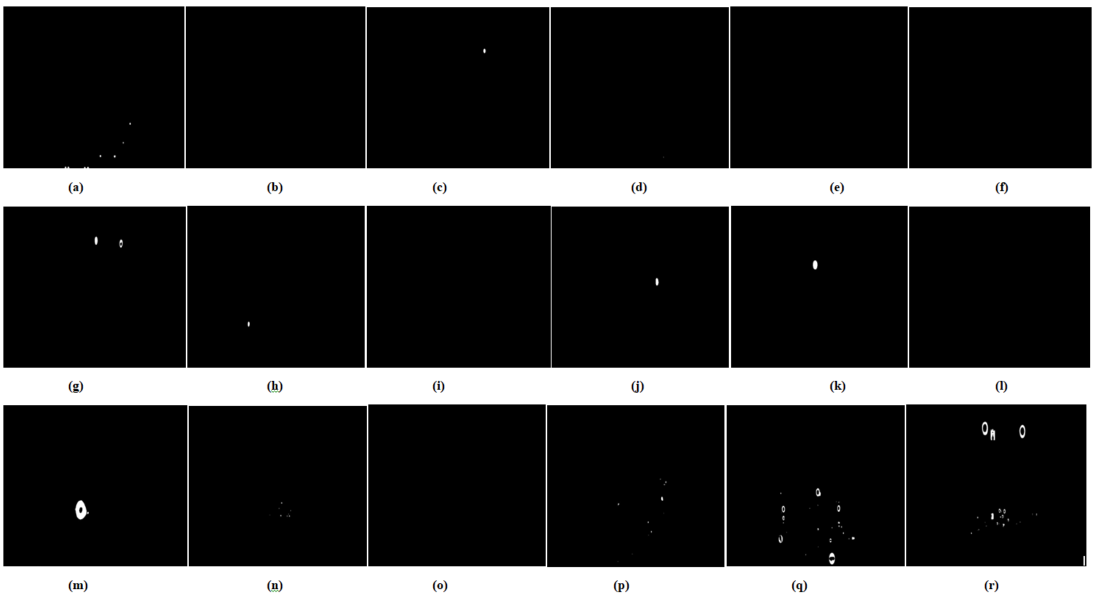


Figure 4.15: Filtered circle of red component.

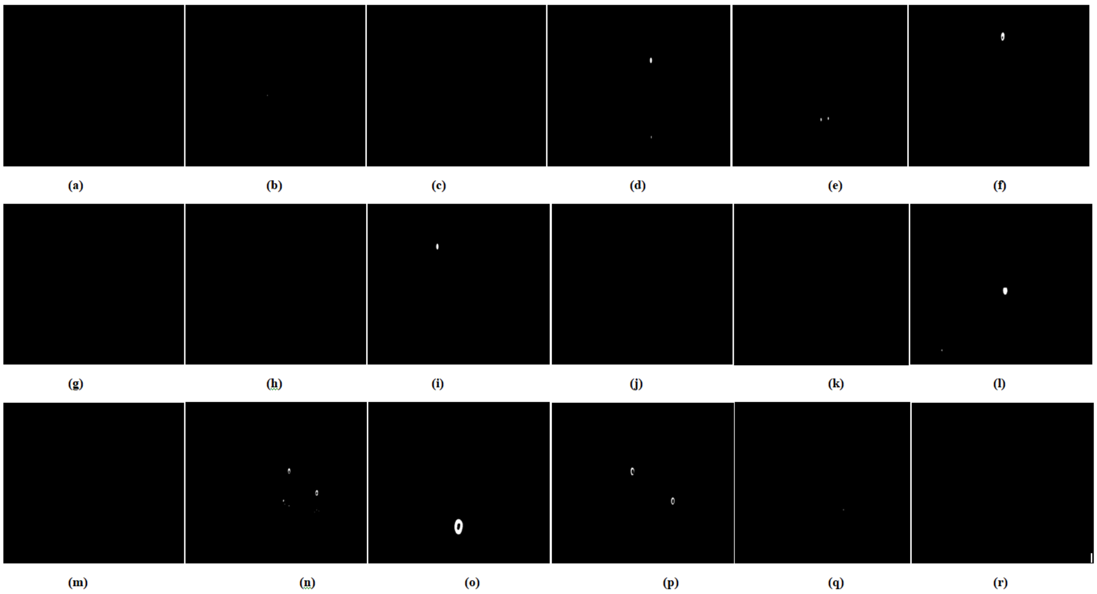


Figure 4.16: Filtered circle of green component.

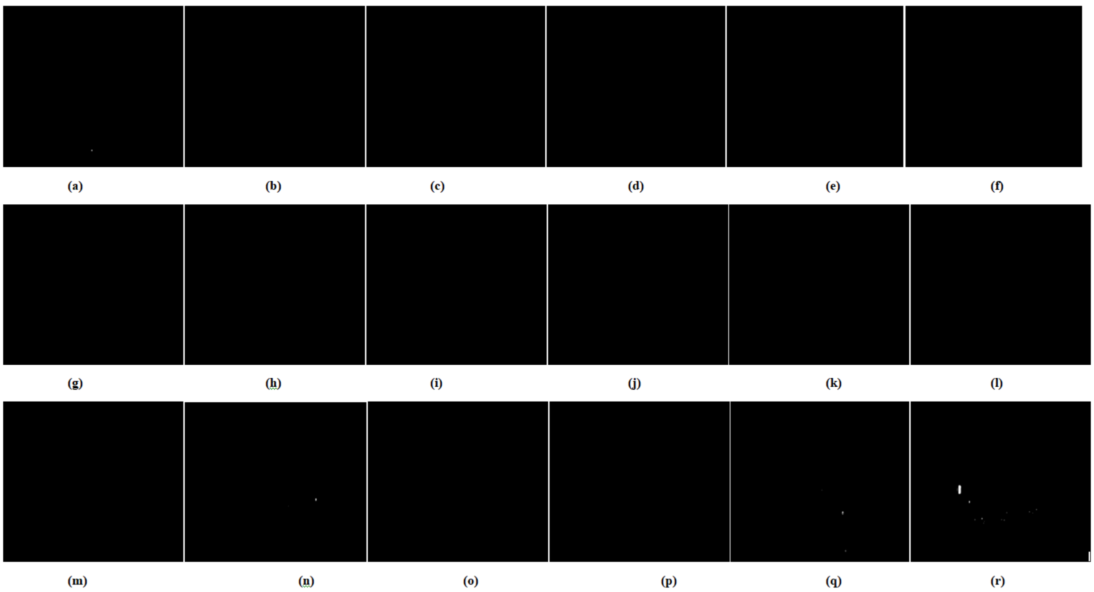


Figure 4.17: Filtered circle of yellow component.

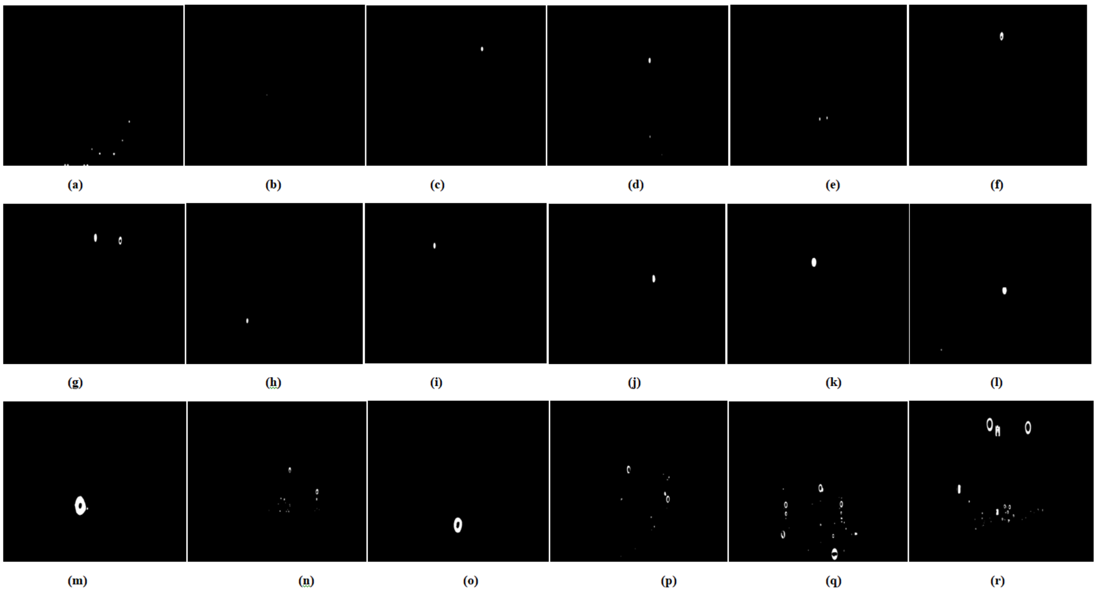


Figure 4.18: After combing filtered circle of red, green and yellow component.

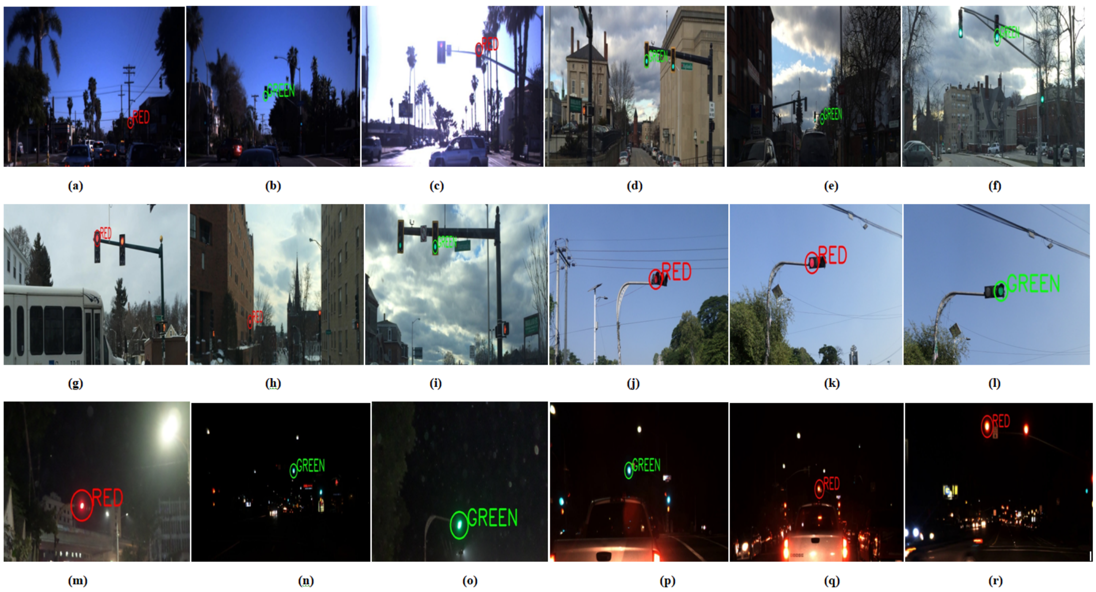


Figure 4.19: After applying voting on filtered circle.

4.6 Traffic light detection when traffic sign present in image

Our proposed methodology can detect traffic light when traffic sign present near to traffic light signal. Filtering is used in our suggested methodology to eliminate the circles created by traffic signs. The processing of images containing traffic signs next to traffic signals is shown in Figure 4.20-4.35. Our model can correctly detect traffic light when traffic/road sign is present close to it, as shown in figure 4.20-4.35.

Figure 4.20: Input image.

Figure 4.21: Image after scaling to 700x700.

Figure 4.22: After elimination of lower 40% Area.

Figure 4.23: After noise removal.

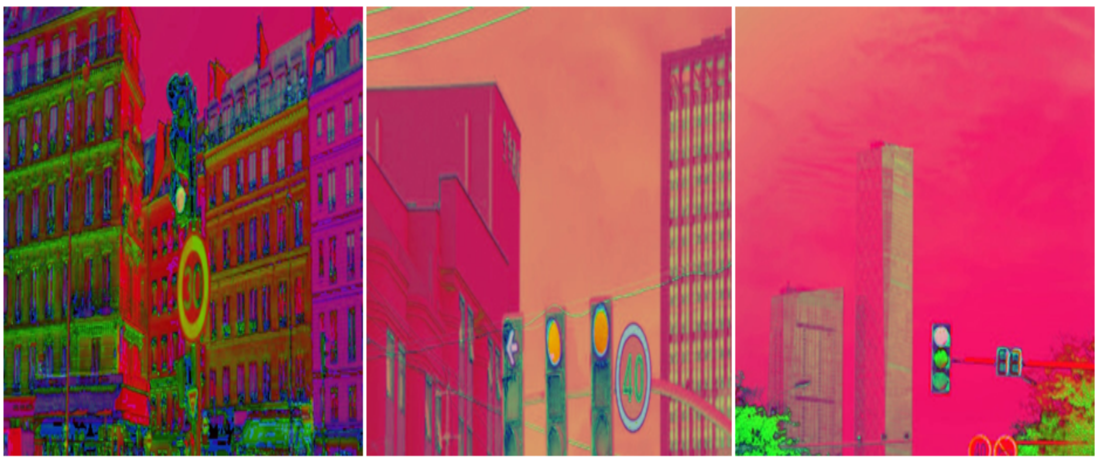


Figure 4.24: After converting to HSV color space.

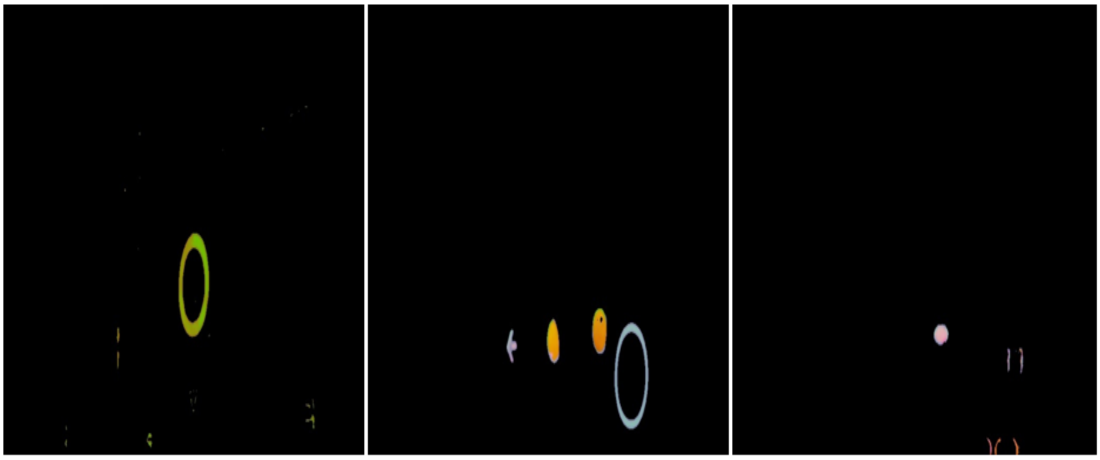


Figure 4.25: Red component of HSV image.

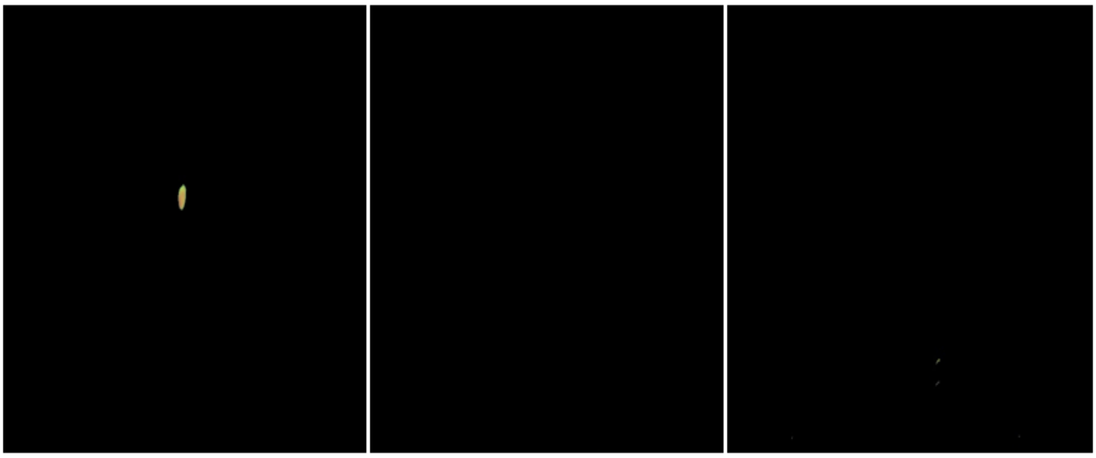


Figure 4.26: Green component of HSV image.

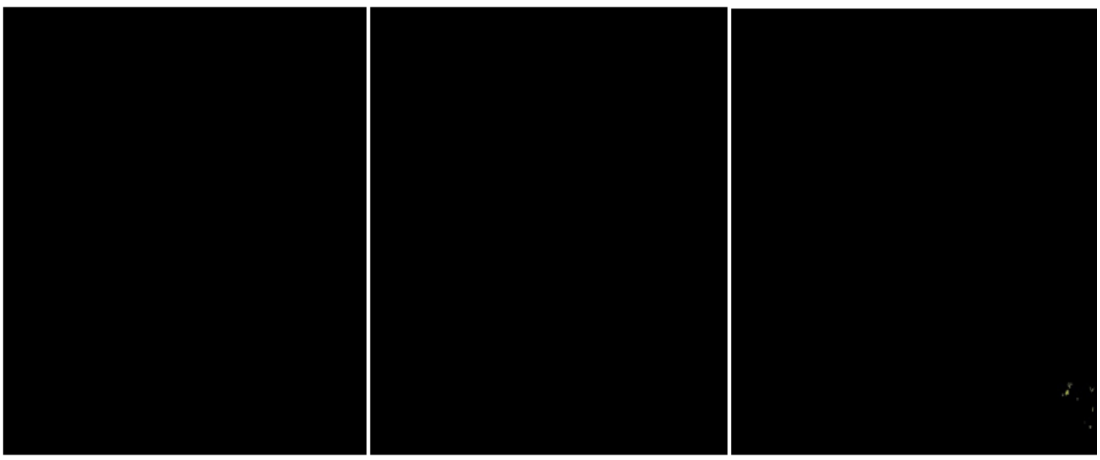


Figure 4.27: Yellow component of HSV image.

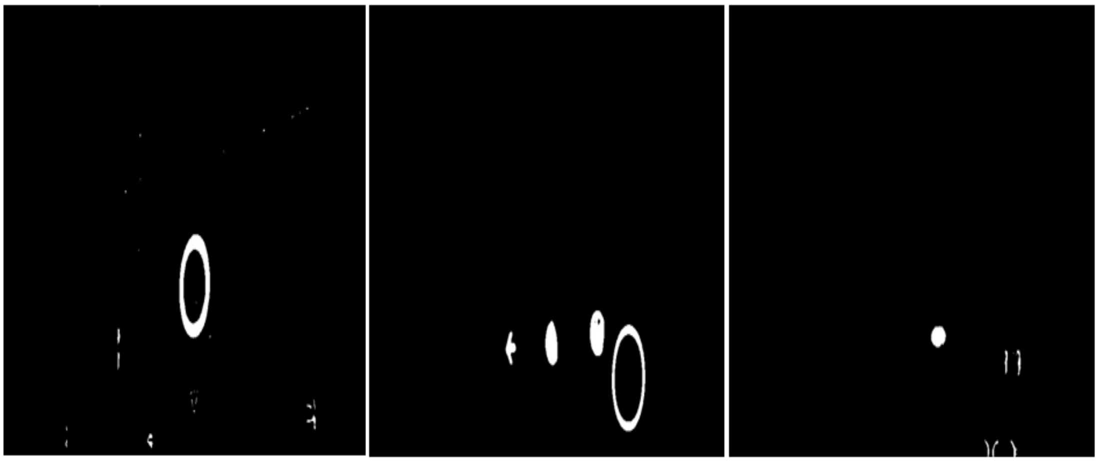


Figure 4.28: Binary image of red component.

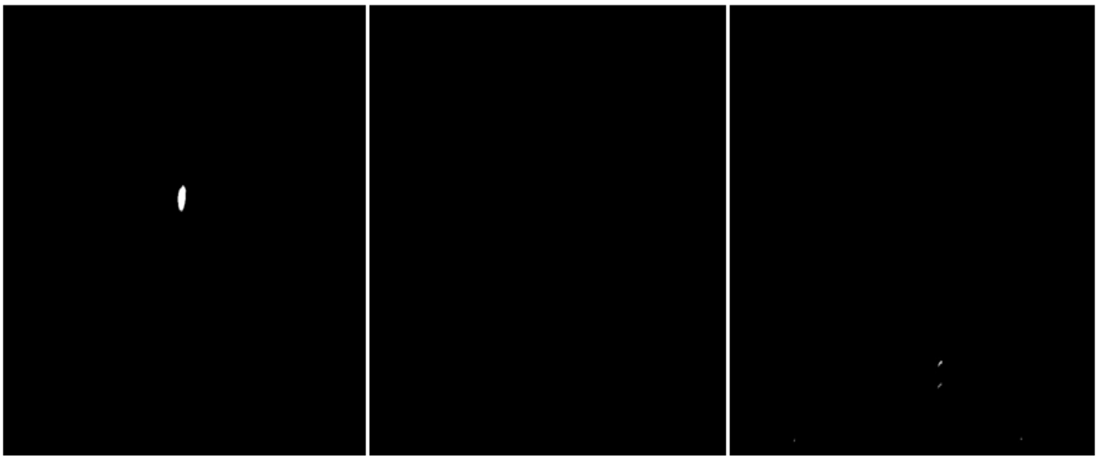


Figure 4.29: Binary image of green component.

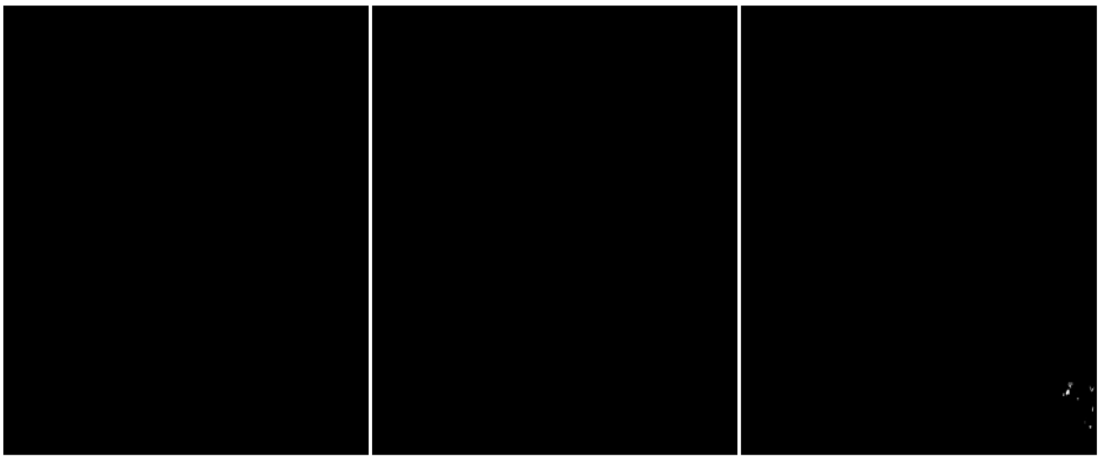


Figure 4.30: Binary image of yellow component.

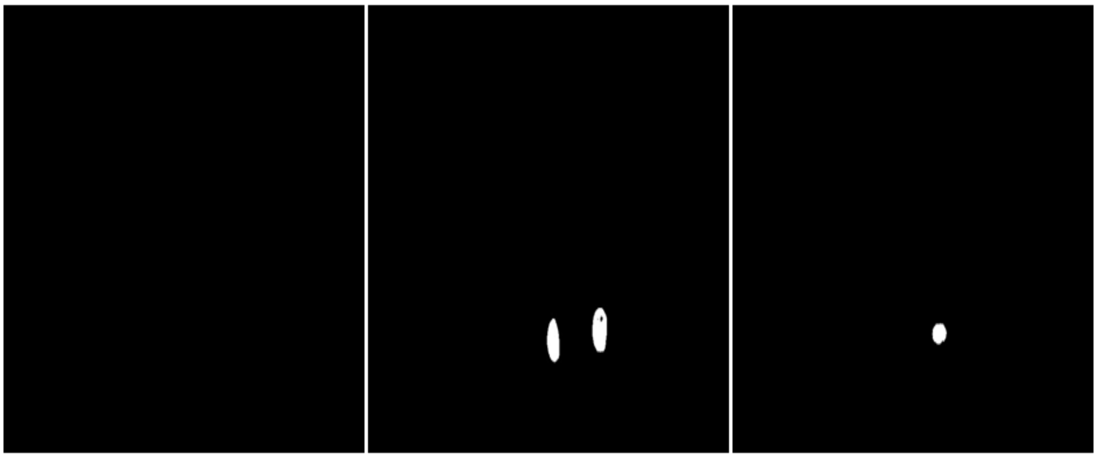


Figure 4.31: Filtered circle of red component.

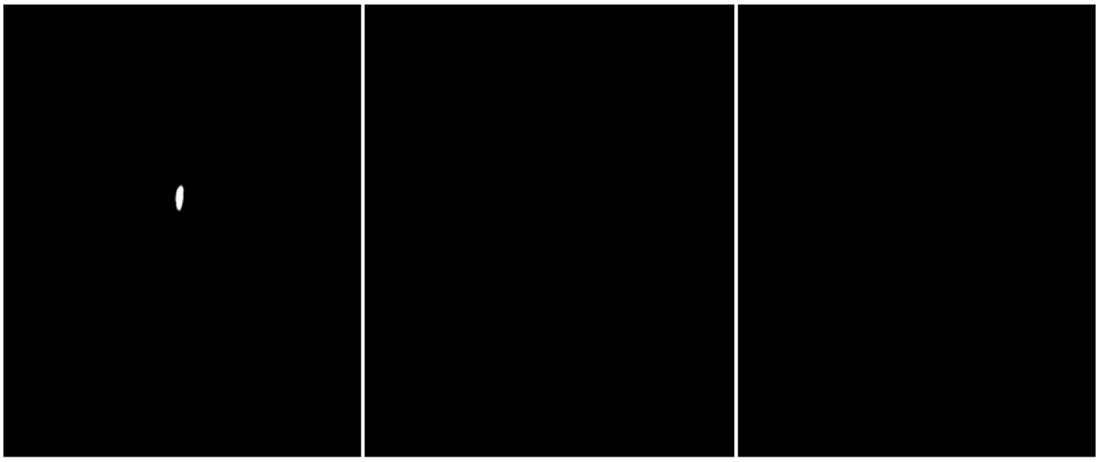


Figure 4.32: Filtered circle of green component.

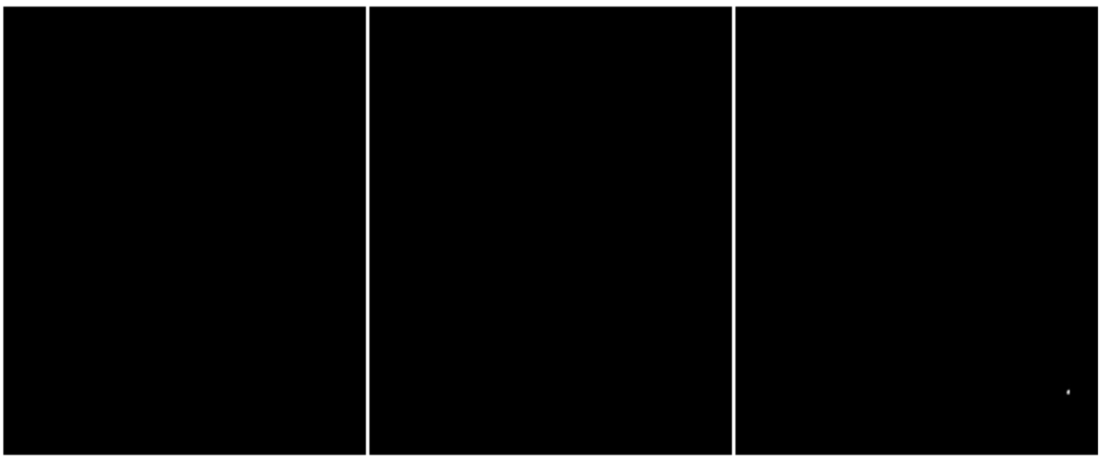


Figure 4.33: Filtered circle of yellow component.

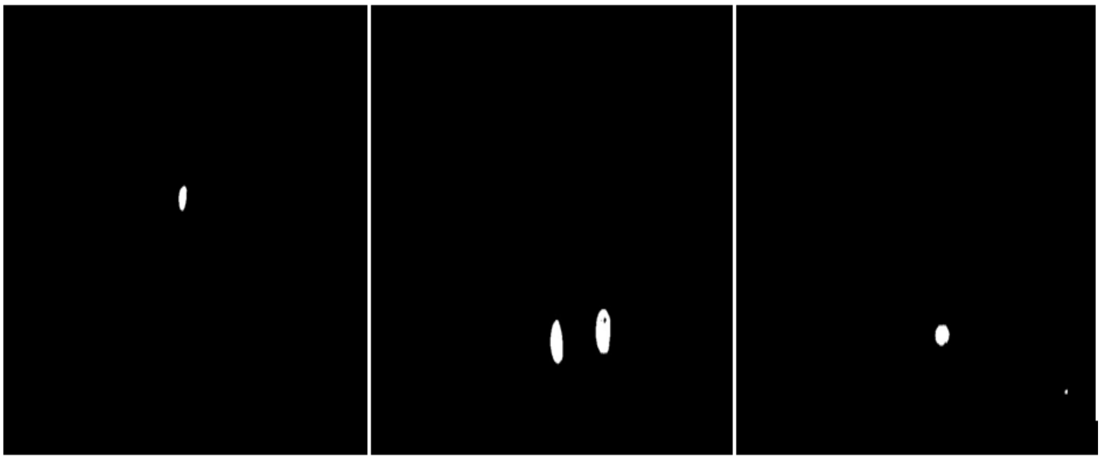


Figure 4.34: After combining filtered circle of red, green and yellow component.

Figure 4.35: Displaying result after applying voting.

4.7 Result Analysis

Result analysis is the process of evaluating system performance in different situation. Our proposed Traffic light framework has been tested in various environment and different time in day. We have used three dataset to evaluate performance of the proposed method. An N x N matrix known as confusion matrix is used to evaluate the performance of a classification model, where N is the number of target classes. The matrix compares the actual goal values to the machine learning model's predictions. This provides us with a comprehensive picture of how well our classification model is working and the types of errors it makes. The confusion matrix in figure 4.1-4.3 depicts the results of our evaluation for the LISA, WPI, and Bangladeshi datasets.

Total Samples = 3,500	**Predicted Red**	**Predicted Green**	**Predicted Yellow**
Actual Red	**1,691**	0	55
Actual Green	46	**1,566**	17
Actual Yellow	22	0	**103**

Table 4.1: Confusion matrix for LISA dataset.

Total Samples = 1,044	**Predicted Red**	**Predicted Green**	**Predicted Yellow**
Actual Red	**325**	0	14
Actual Green	2	**690**	13
Actual Yellow	0	0	**0**

Table 4.2: Confusion matrix for WPI dataset.

Total Samples = 30	**Predicted Red**	**Predicted Green**	**Predicted Yellow**
Actual Red	**13**	0	0
Actual Green	0	**17**	0
Actual Yellow	0	0	**0**

Table 4.3: Confusion matrix for Bangladeshi dataset.

We have calculated accuracy of the proposed framework using the below formula:

$$Accuracy = (correctly\ predicted\ sample\ /\ total\ testing\ sample)\ * 100\% \quad (4.1)$$

Our proposed model accuracy on different dataset given below:

Dataset	Total Testing Sample	Total Correct Prediction	Accuracy
LISA	3,500	3,360	96%
WPI	1,044	1,015	97.22%
Bangladeshi	30	30	100%

Table 4.4: Our proposed model performance.

4.8 Comparison with Other Model

Performance comparison of our proposed model with other model on LISA dataset given below:

LISA Traffic Light Dataset	
Method	**Accuracy**
[9]	91.12%
[10]	92.4%
[15]	92.8%
Proposed method	96%

Table 4.5: Comparison on LISA dataset

Performance comparison of our proposed model with other model on WPI dataset given below:

WPI Traffic Light Dataset	
Method	**Accuracy**
[13]	95.7%
[10]	91.2%
[15]	73.5%
Proposed method	97.22%

Table 4.6: Comparison on WPI dataset

For the following reasons, our proposed model outperforms the other proposed model:

1. For segmenting our input images, we used a modified K-means clustering technique. To segment the images, we employed pixels HSV color measure instead of pixels distance measure. As a result, we were able to remove undesirable items from the image, lowering its complexity. After segmentation, the image only comprises objects that are red, green, or yellow in color. As a result, the image solely contains color candidates for traffic lights.

2. To find the circles in the binary images, we employed a contour detection

algorithm. Any circle-like object of any size can be detected using contour detection. The smallest circle created by a traffic light in the LISA dataset, for example, has an area of 10 pixels, and the contour detection algorithm correctly detected it.As a result, our model outperforms other models in the LISA dataset, which contains a large number of small target circles ranging in area from 10 to 15 pixels. The smallest circle in the WPI dataset has an area of 80 pixels, and contour detection properly detected these circles.

3. Our model uses a filtering strategy for detected circles, which uses the BFS algorithm and a defined window to eliminate circles with a low likelihood of being traffic lights. As a result, the circle created by vehicle tail lights, street lights, and commercial sign board lights is erased. As a result, by employing these techniques, we were able to lower the likelihood of misleading circles being detected as traffic lights.

4. We used voting from pre-calculated traffic box positions to determine which circle among candidate circles has the best chance of becoming a traffic light.

So, our proposed models ability to smart segmentation of input image, detection of small circles, filtering of detected circles and voting technique helped it to perform better than other model.

4.9 Conclusion

The outcome of the identification for Traffic Light is shown in this chapter. The suggested framework's performance is also addressed. The proposed approach gives improved accuracy, as evidenced by the findings. The thesis work is brought to a close in the next chapter.

Chapter 5

Conclusion

5.1 Conclusion

Development of Traffic light detector framework increases safety for vehicles on the road. Color blind people and autonomous vehicles, in particular, require a traffic light recognition method that can accurately decode instructions provided by traffic signals to avoid potentially dangerous situations. The most faced obstacles in detecting traffic light are lighting and other hurdles such as scale, appearance change, noise, and so on. For several reasons, such as the limitations of the standard dataset, suitable feature extraction approach, and classifier accuracy, traffic light detection is still in improvement.

Captured photographs of traffic lights were used as input in this thesis project. Throughout the pre-processing phases, each sample image was propagated. Images were resized after they were captured. The bottom 40% of the obtained image was removed. To reduce noise, a Gaussian filter was used. The RGB color space was transformed to HSV color format.

The HSV picture is transmitted via a series of phases throughout the segmentation process. Using the modified K-means clustering algorithm, the HSV image is first divided into red, green, and yellow components. Three new HSV images having corresponding pixels values were produced using the component pixels value. A grayscale image is created for each new HSV image. The Otsu automatic thresholding algorithm was then used to transform each grayscale image to a binary image.

Circles were recognized utilizing contour detection in the circle detection and filtering process. Then, using a filtering strategy supported by the BFS algorithm and a defined window, the best circle among the detected circles was chosen.

During the voting process, the red, green, and yellow components of the filtered circle were blended into a single image. Then voting begins from a pre-determined traffic box position; if the circle falls within the traffic box, it receives a vote. Finally, the most popular circle is designated as a traffic light, and the outcome is shown in the input image by writing its color adjacent to the marked circle.

Our proposed methodology achieved 96% accuracy on LISA [14] datset, 97.22% accuracy on WPI [13] dataset and 100% accuracy on Bangladeshi dataset.

5.2 Future Work

Despite the efficacy of numerous algorithms, traffic light detection remains a challenge with no bounds, owing to rapid advancements in the field of computer vision. In this thesis work, we only considered circular traffic light and excluded arrow sign of traffic light.

We will prepare large Bangladeshi traffic light dataset incorporating images from worse light conditions and evaluate performance of our proposed method. Since our architecture is designed for detection of traffic light in still image , we'd like to create a tool for video sequences that uses previous detection outcomes to reduce computational costs even more. With recently designed deep learning methods, the existing framework can be improved.

References

[1] J. Al-Nabulsi, A. Mesleh and A. Yunis, 'Traffic light detection for colorblind individuals,' in *2017 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT)*, IEEE, 2017, pp. 1–6 (cit. on p. 6).

[2] R. Kulkarni, S. Dhavalikar and S. Bangar, 'Traffic light detection and recognition for self driving cars using deep learning,' in *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, IEEE, 2018, pp. 1–4 (cit. on p. 6).

[3] D. Yudin and D. Slavioglo, 'Usage of fully convolutional network with clustering for traffic light detection,' in *2018 7th Mediterranean Conference on Embedded Computing (MECO)*, IEEE, 2018, pp. 1–6 (cit. on p. 7).

[4] P. S. Swami and P. Futane, 'Traffic light detection system for low vision or visually impaired person through voice,' in *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, IEEE, 2018, pp. 1–5 (cit. on p. 7).

[5] A. Ochoa and D. Oliva, 'Smart traffic management to support people with color blindness in a smart city,' in *2018 IEEE Latin American conference on computational intelligence (LA-CCI)*, IEEE, 2018, pp. 1–8 (cit. on p. 7).

[6] S.-H. Lee, J.-H. Kim, Y.-J. Lim and J. Lim, 'Traffic light detection and recognition based on haar-like features,' in *2018 International Conference on Electronics, Information, and Communication (ICEIC)*, IEEE, 2018, pp. 1–4 (cit. on p. 7).

[7] E. Lee and D. Kim, 'Accurate traffic light detection using deep neural network with focal regression loss,' *Image and Vision Computing*, vol. 87, pp. 24–36, 2019 (cit. on p. 8).

[8] X. Li, H. Ma, X. Wang and X. Zhang, 'Traffic light recognition for complex scene with fusion detections,' *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 1, pp. 199–208, 2018. DOI: `10.1109/TITS.2017.2749971` (cit. on p. 8).

[9] Y. Lu, J. Lu, S. Zhang and P. Hall, 'Traffic signal detection and classification in street views using an attention model,' *Computational Visual Media*, vol. 4, no. 3, pp. 253–266, 2018 (cit. on pp. 8, 44).

[10] Z. Ouyang, J. Niu, Y. Liu and M. Guizani, 'Deep cnn-based real-time traffic light detector for self-driving vehicles,' *IEEE Transactions on Mobile Computing*, vol. 19, no. 2, pp. 300–313, 2020. DOI: `10.1109/TMC.2019.2892451` (cit. on pp. 8, 44).

[11] J.-G. Wang and L.-B. Zhou, 'Traffic light recognition with high dynamic range imaging and deep learning,' *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 4, pp. 1341–1352, 2018 (cit. on p. 9).

[12] A. Alam and Z. A. Jaffery, 'A vision-based system for traffic light detection,' in *Applications of Artificial Intelligence Techniques in Engineering*, Springer, 2019, pp. 333–343 (cit. on p. 9).

[13] Z. Chen and X. Huang, 'Accurate and reliable detection of traffic lights using multiclass learning and multiobject tracking,' *IEEE Intelligent Transportation Systems Magazine*, vol. 8, no. 4, pp. 28–42, 2016 (cit. on pp. 9, 25, 44, 47).

[14] M. B. Jensen, M. P. Philipsen, A. Møgelmose, T. B. Moeslund and M. M. Trivedi, 'Vision for looking at traffic lights: Issues, survey, and perspectives,' *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 7, pp. 1800–1815, 2016. DOI: `10.1109/TITS.2015.2509509` (cit. on pp. 24, 47).

[15] K. Behrendt, L. Novak and R. Botros, 'A deep learning approach to traffic lights: Detection, tracking, and classification,' in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 1370–1377. DOI: `10.1109/ICRA.2017.7989163` (cit. on p. 44).