

# Traffic Sign Detection and Recognition Model Using Support Vector Machine and Histogram of Oriented Gradient

**Nabil Ahmed**

Department of Computer Science and Engineering, Ahsanullah University of Science and Technology, Dhaka-1208, Bangladesh  
E-mail: nabil.cse.53@gmail.com

**Sifat Rabbi, Tazmilur Rahman, Rubel Mia and Masudur Rahman**

Department of Computer Science and Engineering, Ahsanullah University of Science and Technology, Dhaka-1208, Bangladesh  
E-mail: showmic96@gmail.com, tazmilurrahmanodri@gmail.com, mdrobin.eng@gmail.com, masudur.rahman0413.cse@aust.edu

Received: 27 October 2020; Revised: 28 December 2020; Accepted: 25 February 2021; Published: 08 June 2021

**Abstract:** Traffic signs are symbols erected on the sides of roads that convey the road instructions to its users. These signs are essential in conveying the instructions related to the movement of traffic in the streets. Automation of driving is essential for efficient navigation free of human errors, which could otherwise lead to accidents and disorganized movement of vehicles in the streets. Traffic sign detection systems provide an important contribution to automation of driving, by helping in efficient navigation through relaying traffic sign instructions to the system users. However, most of the existing techniques have proposed approaches that are mostly capable of detection through static images only. Moreover, to the best of the author's knowledge, there exists no approach that uses video frames. Therefore, this article proposes a unique automated approach for detection and recognition of Bangladeshi traffic signs from the video frames using Support Vector Machine and Histogram of Oriented Gradient. This system would be immensely useful in the implementation of automated driving systems in Bangladeshi streets. By detecting and recognizing the traffic signs in the streets, the automated driving systems in Bangladesh will be able to effectively navigate the streets. This approach classifies the Bangladeshi traffic signs using Support Vector Machine classifier on the basis of Histogram of Oriented Gradient property. Through image processing techniques such as binarization, contour detection and identifying similarity to circle etc., this article also proposes the actual detection mechanism of traffic signs from the video frames. The proposed approach detects and recognizes traffic signs with 100% precision, 95.83% recall and 96.15% accuracy after running it on 78 Bangladeshi traffic sign videos, which comprise 6 different kinds of Bangladeshi traffic signs. In addition, a public dataset for Bangladeshi traffic signs has been created that can be used for other research purposes.

**Index Terms:** Traffic Sign, Detection, Recognition, Support Vector Machine, Histogram of Oriented Gradient.

## 1. Introduction

Traffic signs or road signs are pictorial signs erected at the side of or above roads to give instructions or provide information to road users. Traffic signs provide drivers with valuable information about the road, in order to make driving safer and easier [1]. Many countries have adapted these pictorial traffic signs as early as the 1930s, due to the gradual increase in traffic and the use of automobiles. Most traffic signs around the world are designed based on the Annexe 1 of the Vienna Convention on Road Signs and Signals (1968) [2], which lays out the general guidelines for the design of traffic signs. A traffic sign detection and recognition system detect the traffic signs from the surroundings and also identifies the type of the traffic sign. As in most other systems utilizing machine learning and computer vision, there are two major components in a typical traffic sign detection system, a trained model and a real-time detection of traffic signs along with recognition of the category of the traffic sign with the help of the classifier. Through the coordinated implementation of these components, a traffic sign detection and recognition system perform its task. In doing so, it conveys the message or instruction depicted on the traffic sign. This contributes greatly in assisting the drivers to identify the instructions of the traffic signs and act accordingly.

Authors such as Chakraborty et. al. [3], Zhao et. al. [5], Sugiharto et. al. [23], Rahmad et. al. [7] and Ardianto et. al. [6] have developed traffic sign detection and recognition systems using SVM classifier. On the other hand, Roxas et. al.

[8], Wang et. al. [9] and Filatov et. al. [22] worked on the development of traffic sign detection and recognition systems using Neural Networks. In most of these works, Histogram of Oriented Gradient (HOG) property of the traffic signs has been utilized to identify the type of the traffic sign. All of the approaches mentioned here have worked primarily with static images. Furthermore, out of the above-mentioned authors, only [3] have developed a traffic sign detection and recognition system for Bangladeshi traffic signs. Hence, we have considered [3] as the best work regarding the development of a Bangladeshi traffic sign detection and recognition system. However, like the other works mentioned here, it has not devised a mechanism for detection and recognition of traffic signs from video input. Also, it has a lower accuracy compared to the other systems, and also lacks a specific dataset composed of Bangladeshi traffic signs.

Due to the limitations of the above-mentioned authors, including Chakraborty et. al., we have set forth certain objectives to be fulfilled in our research work. Our work aims to develop a more efficient traffic sign detection and recognition system that is capable of detecting Bangladeshi traffic signs efficiently from video input, which is trained using a dataset made up of images of Bangladeshi traffic signs. Thus, through our research work, we aim to achieve the following objectives:

1. Develop a traffic sign detection system of greater accuracy.
2. Develop a traffic sign detection system capable of detecting traffic signs from video input.
3. Create a definite dataset made up of images of Bangladeshi traffic signs.

This article proposes an automated approach to detect and recognize Bangladeshi traffic signs using SVM classifier, based on their HOG property. This approach consists of two major phases, the first being training phase and the second being detection and recognition phase. In the training phase, a well-defined dataset on Bangladeshi traffic signs will be formed. This dataset will be used to generate our SVM classifier model for Bangladeshi traffic sign classification based on HOG. In the second phase, the actual detection and recognition of Bangladeshi traffic signs from video data takes place. The traffic signs are detected from videos, rather than static images. In the recognition part, the separated traffic sign object is sent to the classifier to classify it into one of the six different categories of Bangladeshi traffic signs. Through the steps elaborated here, this model will accurately identify the Bangladeshi traffic signs.

We implemented our approach on Python using the external libraries of OpenCV and NumPy. Overall, the preliminary evaluation, which was carried out using 78 sample videos, exhibits accuracy, precision and recall at 95%, 92.5% and 100% respectively. These figures are better than most of the methods utilizing SVM classifier. Thus, it can be said that the detection and recognition of Bangladeshi traffic signs can be conducted with sufficient efficiency using SVM via classification on the basis of HOG properties.

In summary, the article makes the following major contributions:

1. An automated technique has been proposed in order to detect and recognize traffic signs using SVM based on HOG properties from video input.
2. A repository comprising of sample images of Bangladeshi traffic signs collected from the streets of Dhaka has been created.

The remainder of the article is organized as follows: Section II discusses the existing works related to traffic sign detection and recognition systems, and Section III describes the methodology of the research. Section IV describes the proposed traffic sign detection and recognition approach, while Section V discusses results from a preliminary evaluation using sample videos of Bangladeshi traffic signs. Section VI shows a case study of the whole approach and finally, Section VII concludes the article.

## 2. Related Work

### 2.1. Bangladeshi Road Sign Detection Based on YCbCr color model and DtBs Vector

Here, Chakraborty et. al. [3] has proposed a system for detecting Bangladeshi traffic signs based on the border rim color properties and the shape of the traffic sign. Since most of the Bangladeshi traffic signs possess red or blue rim, hence the RGB color space of the image is converted to YCbCr color space. It is a family of color spaces that consists of three components Y (luma or illumination), Cb (blue-difference) and Cr (red-difference)[4]. The candidate regions are detected from Cb and Cr color space. Statistical value thresholds are used for color segmentation, and closing operation is done to eliminate noise. In the next step, labeling and filtering is used to extract the shapes. Regions that satisfy the geometric properties of road signs are considered as candidate objects. Finally, Distance to Borders (DtBs) vector is used to verify the Region of Interest (ROI) for detecting the Bangladeshi traffic sign.

In this approach, the properties intrinsic to Bangladeshi traffic signs have been utilized in color segmentation. A similar approach can be used in our work for detecting Bangladeshi traffic signs. However, the accuracy is noticeably lower (91.73%). Therefore, instead of YCbCr space, we can do the segmentation in HSV space for better segmentation. Also, we can introduce a detection system from video input for greater practicality of its application.

## 2.2. *Research on Road Traffic Sign Recognition Based on Video Image*

A system for detecting traffic signs using image collected from videos was proposed by Zhao et. al. [5]. At first the image preprocessing is done in 3 steps: - (i) Reduction of size to 384\*216, (ii) Adjustment of brightness using histogram equalization, (iii) Median filtering of image. Median Filtering is an image filtering method in which each pixel value in an image is replaced with the median value of its neighboring pixels including itself [25]. Next, the segmentation of color image is done based on HSV color model, followed by morphological processing and area filtration respectively. Then the traffic signs are segmented according to their circularity, rectangularity and elongation. Information such as area, center of gravity and smallest rectangle containing the traffic sign is stored in the matrix. The extracted traffic signs are classified using a SVM classifier based on their HOG property.

Here, the traffic sign images were segmented in HSV color space rather than YCbCr. Also, the accuracy stood at a slightly improved figure than [3] (94%). Hence in our approach, we can utilize the HSV color space for image segmentation for improved accuracy. They utilized video images for their detection system, which will also be useful for our video input-based system.

## 2.3. *Real-Time Traffic Sign Recognition using Color Segmentation and SVM*

According to Ardianto et. al. [6], a traffic sign detection system can be implemented using HOG and SVM. In the detection step, the image is first enhanced and filtered using histogram equalization, color segmentation as well as image enhancing with Gaussian and Laplacian filter. The histogram equalization is done in the YCbCr color space, and the color segmentation is done in the HSV color space. The image is also enhanced with Gaussian and Laplacian filter. After enhancement and filtering, contours are generated by Canny edge detection and candidate boxes are generated on the contours. Boxes that are too big or too small are eliminated. The traffic signs are classified based on their HOG feature using SVM classifier. For recognition phase, the input frame is divided into multiple blocks and processed in parallel.

The accuracy of this approach stands at 94.9%, which is even higher than [5]. Although this approach also uses HSV color space for image segmentation, it also makes use of the YCbCr space for image enhancement by histogram equalization. Thus for greater accuracy of detection, we have also considered conversion of our video input to YCbCr color space for enhancement.

## 2.4. *Indonesian Traffic Sign Detection and Recognition Using Color and Texture Feature Extraction and SVM Classifier*

Rahmad et. al. [7] has devised a method of traffic sign detection using SVM classifier and color and texture feature extraction. Their approach has been able to detect s damaged signs, faded color, and natural condition signs. There are two main steps in this approach. At first, a dataset consisting specifically of Indonesian traffic signs is created, in order to develop a proper system for Indonesian traffic sign detection. Then, the image is segmented based on RGBN, and then thresholded. The blobs are then extracted, and those too large or too small are removed. Bounding boxes are then generated for each candidate region. The candidate regions' HOG is extracted. Based on this HOG feature, the traffic signs are classified by an SVM classifier.

In this approach, the accuracy of the work stood at 86%. Although it is quite low, this method has also used the concept of developing a traffic sign dataset of Indonesian traffic signs for this system. Therefore, in order to fulfill our objective of developing a traffic sign detection and recognition system for Bangladeshi traffic signs, it has become evident that a dataset specifically made of Bangladeshi traffic signs needs to be made.

## 2.5. *Vision based traffic sign compliance evaluation using convolutional neural network*

A system devised by Roxas et. al. [8] not only detects and recognizes the traffic signs, but also classes the traffic signs as being either compliant or non-compliant with the Vienna Convention. Due to the large amount of data in the dataset and the diversity of the traffic sign types, the classification method of Convolutional Neural Network (CNN) has been implemented. A dataset is made from images in google street view. The input images are fed into the AlexNet architecture. The compliance classification of the input traffic signs is done using transfer learning.

In this system, the accuracy of detection stands at 91.45%. However, the 2-class classification system, as well as the means by which the dataset is procured in real time, has been considered by us for our improved approach in devising a Bangladeshi traffic sign detection and recognition system.

## 2.6. *Research and application of traffic sign detection and recognition based on deep learning*

Another work implemented by Wang et. al. [9] uses convolutional neural network to detect and recognize traffic signs. It uses Caffe [10], an open-source framework, to train a model using traffic sign training set that can detect and recognize traffic signs. Training and verifying a deep convolutional neural network traffic sign recognition model requires a large amount of traffic sign data as a basis. [11] In this model, the German GTSRB [20] dataset has been used. [12] At first the pixel values are all normalized to values between 0 and 0.5. This is done to adjust the model to perform in neural networks, as the neural network performs best when the values are between 0 and 1. This work uses a VGG-16

[13] as a front-end network structure of the SSD [14] algorithm to detect and identify traffic signs. For each input image, the model uses the convolution layer to extract the features of the image. Then probability and regression analysis of the target object coordinates is done on different scale feature maps. Non-maximum suppression algorithm is then used to eliminate the redundant test boxes. The final test results will then determine the correct output.

This traffic sign detection model uses the GTSRB traffic sign dataset and achieves an accuracy of 96%. The GTSRB dataset used by this traffic sign has traffic sign images that share many similar properties with Bangladeshi traffic signs. Also, the image normalization is used for efficient detection of traffic signs, which was also considered for implementation in our proposed system.

### 3. Research Methodology

The focus of this research is to develop an efficient traffic sign detection and recognition system for Bangladeshi traffic signs, which will be trained with the help of a dataset made up of Bangladeshi traffic signs.

The first step would be to form a dataset of Bangladeshi traffic signs. For this purpose, pictures of traffic signs would be collected from the streets of Bangladesh, and they would be stored into separate categories based on the type of the traffic sign. This dataset will be used to train our classifier model.

Next, we generate a SVM classifier model using OpenCV library of Python [17]. This is because The classification of positive and negative data is better in SVM [24]. The SVM classifier will classify the traffic signs based on the HOG property.

After the training phase, we design a procedure to detect traffic signs from the video input and send it to the classifier for classifying, also using OpenCV. The procedure would process each frame of the video for detecting traffic signs. The detected traffic sign objects would then be sent to the classifier.

Finally, to test the efficiency of our procedure, we provide a series of video inputs to our system. The system is considered to run successfully only if the accuracy is above 96%, greater than the literature works studied.

Compared to the other literary works, the advancing features of this approach will be a system to detect Bangladeshi traffic signs effectively from a video input effectively. Also, the classifier will be trained with a dataset made up exclusively of Bangladeshi traffic signs.

### 4. Proposed Approach

The proposed approach is used to train a traffic sign recognition model and uses that model to recognize traffic signs detected by the system. The approach detects traffic signs using image processing techniques and recognizes traffic signs with a model trained using SVM, based on the HOG property of detected traffic signs.

The approach has been illustrated in Fig. 1. Fig. 1 depicts a flow chart showing all the steps of the entire proposed system, from start to end. The parallelograms depict the Input/Output, the rectangles show the actions while the diamonds symbolize the conditional flows. The approach shown in Fig. 1 can be divided into two parts:

1. Training the model
2. Detection and recognition of traffic sign object from source file

The sequence to the right of the chart, starting from the 'Start' step upto the 'Generated SVM Model' phase, shows the step (i). All the remaining portions in the chart, starting from the 'Play Video File' phase after the 'Generate SVM Model' phase, upto the end, is part of step (ii). There are two loops in this phase. The first is after the 'Find Largest Traffic Sign-Like Object' phase, where the system decides whether to send the detected object to the classifier or not. An affirmative course causes the flow of action to move to the 'Generated SVM Model' in the training phase. The second loop is after displaying traffic sign type, when it is decided whether or not further video frames can be extracted. A lack of frames will terminate the process.

At first, a dataset of Bangladeshi traffic signs is created. The traffic signs are separated into classes. This dataset is then loaded into the system, along with the label associated with it. The data is then shuffled, and the images are all set to a uniform position. The HOG is determined for the dataset. The data is separated into training and testing portions. Using the training data, an SVM model is generated. For the SVM model to be generated, the Gaussian Kernel parameter ( $\gamma$ ) and misclassification cost parameter ( $C$ ) are also set accordingly. This is to ensure that a proper tradeoff occurs between bias and variance. Radial basis function is used as the kernel method for our classifier. The classifier is set to C-Support Vector type, since there are more than two classes. Using these factors, the classifier model is generated by training it using the dataset and HOG features. This trained model is crucial for conducting the actual recognition function of the traffic sign. Thus, the training phase can be considered as the first phase of traffic sign detection and recognition system.

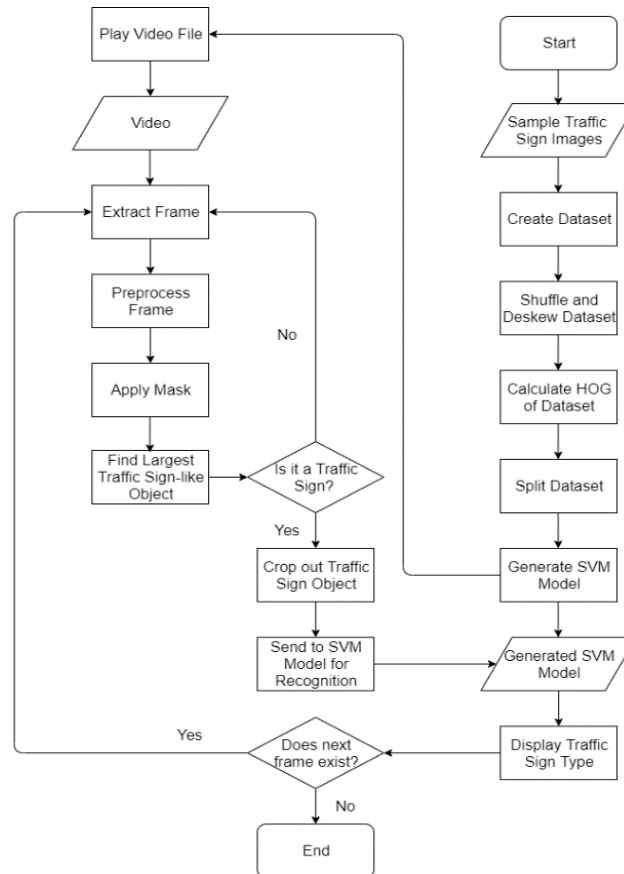


Fig.1. Proposed Approach for our System

The second phase is the traffic sign detection and recognition phase. In this phase, a video is used as input and the traffic signs are detected for each frame of the video. Every frame is preprocessed by contrast stretching, Laplacian of gaussian and binarization. The connected components are detected and the objects smaller than threshold value are then eliminated from the resultant binary image. A color mask is also applied to eliminate objects having the undesirable colors. Then, the remaining contours are extracted one by one. The traffic sign-like objects are selected and only the largest one is sent into the classifier model. The classifier model extracts the HOG value of the traffic sign-like object and compares it to the HOG values of the other traffic sign-like objects. The class of traffic sign that bears the closest value to the extracted traffic sign is assigned to the latter. This process is repeated for every frame individually.

In the following section, the classification model training phase (Subsection A) and the traffic sign detection and recognition phase (Subsection B) have been elaborated in detail.

#### 4.1. Training of Traffic Sign Classifier Model

The proposed method of training the classifier model is depicted in Algorithm 1. Prior to training the model, a dataset of Bangladeshi traffic signs is collected. Our collected dataset consists of 1000 traffic sign images of 6 types. The 'Dataset' in this algorithm is the set of those images and 'Labels' depicts one of the six different types associated with each image. For each class C, the images associated with that class are read. For every one of those images, their associated class is stored in the 'Labels' set. Also, in the process, the images are resized to 32 x 32. The Dataset along with their associated labels are shuffled. Also, the images are all de-skewed to set them all to a uniform position. The HOG is calculated for each image using a HOG descriptor and stored in the 'HOG' set. The HOG descriptor used here has a window size of 20 x 20, block size of 10 x 10, block stride of 5 x 5, cell size of 10 x 10, and is of 64 levels. The Dataset, Labels and HOG are split into the training and testing parts respectively. 90% is allocated for training and the rest is allocated for testing. Using the training set of HOG and Labels, an SVM model is trained. In this approach, the value of  $\gamma$  is set to 0.50625, C is set to 12.5, to ensure a low bias and high variance. The Radial Basis Kernel function is utilized in this model and the type of classification is C-Support Vector Classification due to more than two classes being present. The resultant model is used in the next step to detect and classify the traffic signs.



**Algorithm 1** Generation of SVM Classifier Model

---

**Input:** Dataset of Bangladeshi Traffic Sign Images  
**Output:** SVM Classifier using HOG

- 1: *Dataset* ← **Null**
- 2: *Labels* ← **Null**
- 3: **for** each class *C* **do**
- 4:   *Dataset*[*C*] ← *ReadData*(*C*)
- 5:   **for** each data *D* in *Dataset*[*C*] **do**
- 6:     *Labels*[*D*] ← *C*
- 7:     *D* ← *resize*(*D*,32,32)
- 8:   **end for**
- 9: **end for**
- 10: *Dataset* ← *ShuffleData*(*Dataset*)
- 11: *Dataset* ← *Deskew*(*Dataset*)
- 12: *HOG* ← *CreateHOGDescriptor*()
- 13: *HOGList* ← **Null**
- 14: **for** each data *D* in *Dataset* **do**
- 15:   *HOGList*[*D*] ← *ComputeHOG*(*HOG*, *D*)
- 16: **end for**
- 17: *train\_data*, *test\_data* = *SplitData*(*Dataset*)
- 18: *train\_label*, *test\_label* = *SplitData*(*Labels*)
- 19: *train\_hog*, *test\_hog* = *SplitData*(*HOG*)
- 20: *model* ← *TrainSVM*(*train\_data*, *train\_label*, *train\_hog*)

---

**2. Detection and Recognition of Traffic Sign from Video Input**

The detection and recognition phase consists of providing a video stream as input, and detecting the traffic sign and its type from the video stream, frame by frame. This task occurs in several elaborate steps, which have been highlighted in Algorithm 2. The steps occur here frame by frame.

**Algorithm 2** Similarity Measurement algorithm

---

**Input:** A video file 'video\_file'  
**Output:** traffic sign image and its type for each frame in video file

- 1: **for** frame in video\_file **do**
- 2:   frame ← *resize*(frame)
- 3:   frame ← *HistogramEqualize*(frame)
- 4:   frame ← *LaplacianOfGaussian*(frame)
- 5:   frame ← *Binarize*(frame)
- 6:   components ← *ConnectedComponents*(frame)
- 7:   **for** C in components **do**
- 8:     **if** C < threshold **then**
- 9:       frame ← *RemoveComponent*(frame, C)
- 10:    **end if**
- 11:   **end for**
- 12:   frame ← *ApplyMask*(frame)
- 13:   contours ← *FindContour*(frame)
- 14:   Sign ← **Null**
- 15:   Largest\_Coefficient ← **Null**
- 16:   **for** C in contours **do**
- 17:     distances ← *FindDistances*(C)
- 18:     radius ← *max*(distances)
- 19:     normalized ← *normalize*(distances, radius)
- 20:     **if** *average*(normalized) ≥ *similarity\_threshold* **and** *verage*(normalized) > Largest\_Coefficient **then**
- 21:       Sign ← C
- 22:       Largest\_Coefficient ← *average*(normalized)
- 23:     **end if**
- 24:   **end for**
- 25:   SignImage Crop(frame, Sign)
- 26:   SignType model(SignImage)
- 27:   Display(frame, SignImage, SignType)
- 28: **end for**

---

At first, the frame is preprocessed. This is because the automatic detection of the signs is difficult when worked with the real time scenery, as the background is highly complicated for identifying and recognizing the signs in

the road sides [21]. The frame preprocessing is done in four steps, which are resizing, histogram equalization, Laplacian of Gaussian and Binarization respectively. In resizing, the frame is resized to 640 x 480 dimensions. The histogram equalization of only the Y component in YCbCr format in frame is performed. Then Laplacian of Gaussian transformation is applied on the frame. Finally, the frame is converted to binary form using a threshold of 32, where pixel values below 32 are suppressed.

After preprocessing of images, all the connected components within the binary image are detected. The components whose size is less than 300 are suppressed. Next a mask is applied on the frame. To generate the mask for this purpose, at first, we took the HSV form of the image. Since the HSV color space is less sensitive to variable lighting conditions, many researchers considered using this method for traffic sign segmentation [19]. The blue, white and black masks are taken from this HSV form separately. These three masks are then combined to create our desired mask. Finally, the mask is applied in the bitwise AND operation of the binary frame with itself, to remove the unnecessary colors. Now the frame is ready for detection of traffic sign objects.

All the contours of the processed frame are now extracted. These contours are iteratively accessed to find the traffic sign-like object. For each contour, the centroid is calculated and the distance of each border pixel from the centroid is calculated. The largest distance is taken as the radius. Then each distance is normalized by dividing each distance by the radius, and the mean of the normalized values are taken. Thus, the mean normalized value can be within 0 to 1, 1 indicating a complete similarity to radius  $r$ , that is, a perfect circle. For our work, we have considered a threshold value of 0.65. Any contour that has a mean normalized value greater than 0.65 is considered an ideal traffic sign candidate. If multiple such contours are detected, only the largest one is considered. This concludes the detection process of traffic sign object.

After finding our desired contour, the object encircled by the contour is cropped from the frame and sent to the trained model for recognition. The model calculates its HOG value and compares it to the HOG values of each of the 6 classes of traffic signs. The class that closest matches the candidate object in terms of HOG value is assigned to it. This completes the recognition process of the traffic sign object.

## 5. Implementation and Result Analysis

To verify the validity of our approach, the proposed approach for Bangladeshi traffic sign detection and recognition system has been implemented. A dataset of Bangladeshi traffic signs has been created with 1000 images of 6 different classes, each class containing approximately 170 images. This dataset was used to train the model. The entire approach was implemented using Python language along with several external libraries, which includes OpenCV, Numpy and Scikit. For verifying the correctness of the model, the metrics of the confusion matrix of traffic signs has been taken into consideration.

### 5.1. Dataset of Bangladeshi Traffic Signs

The name of our dataset is Dhaka City Traffic Sign Dataset[15]. The dataset has been formed by taking pictures of traffic signs from various parts of Dhaka city. There are a total of approximately 1000 images of different traffic signs taken in various parts of Dhaka city. All of these traffic signs are classified into 6 different classes, each class consisting of approximately 170 sample traffic sign images. Each class stands for a specific traffic sign. The classes are mentioned in Fig. 2.

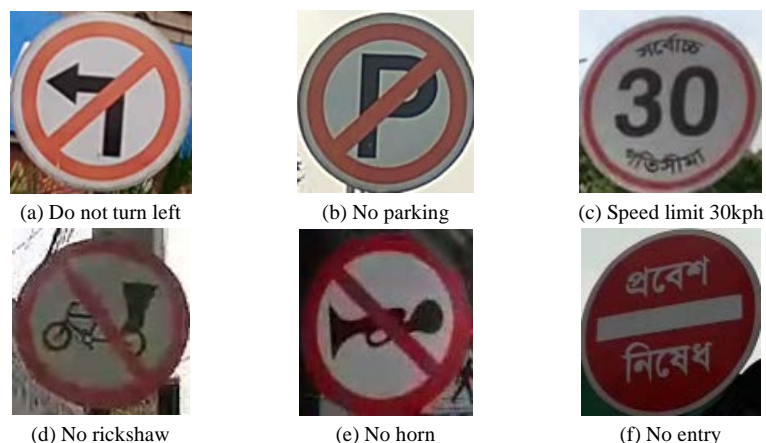


Fig.2. Traffic sign classes used in the dataset

For our traffic sign images, at first full-sized pictures of traffic signs are taken. Only the traffic signs are cropped out from those full-sized images. These are the final images used in our dataset.

## 5.2. Environmental Setup

The algorithm was implemented using Python programming language as mentioned before. The version of Python used to implement the approach was Python 3.7. The project was implemented with the help of the following tools:

- JetBrains PyCharm[16]
- OpenCV[17]
- Numpy[18]

JetBrains Pycharm is the IDE on which the Python program was implemented. The program was implemented with the help of external libraries such as OpenCV and Numpy. OpenCV is an open source library that performs functions related to computer visions as well as machine learning. OpenCV was used to calculate the HOG, generate the model as well as process the frames and detect traffic sign objects. Numpy library is used to perform advanced calculations on large collections of data, and in this approach, it was used for the same purpose.

## 5.3. Preliminary Results

The results have been generated through observation of 78 input video files consisting of one or more traffic signs. The results of the traffic sign detection through the proposed approach have been shown as in the confusion matrix table (Table 1) given. For our matrix in Table 1, the row labeled ‘Traffic Sign’ indicates the input videos containing traffic signs while the row labeled ‘Non-Traffic Sign’ indicates the input videos not containing any traffic signs. On the other hand, the column indicating ‘Traffic Sign’ denotes the videos in which traffic signs were detected while ‘Non-Traffic Sign’ indicates the videos in which traffic signs are not detected. Here, 72 input videos provided are traffic signs videos, and the remaining 6 videos do not contain any traffic signs. Out of the 72 traffic sign videos, the traffic signs were successfully detected in 69 of them, and the system could not detect traffic signs in 3 of them. On the other hand, out of the 6 non-traffic sign videos, none of the videos had traffic signs detected in them, thus bringing the false positive count to 0. The precision, recall and accuracy of our approach were calculated by equations (1), (2) and (3) respectively. Here, actual traffic sign, actual non-traffic sign, detected non-traffic sign and detected traffic sign stand for true positive (t.p.), false positive (f.p.), true negative (t.n.) and false negative (f.n.) respectively.

$$precision = \frac{t.p.}{t.p.+f.p.} \quad (1)$$

$$recall = \frac{t.p.}{t.p.+f.n.} \quad (2)$$

$$accuracy = \frac{t.p.+t.n.}{t.p.+f.p.+t.n.+f.n.} \quad (3)$$

The precision, recall and accuracy of our approach is calculated at 100%, 95.83% and 96.15% respectively.

Due to the application of custom mask after conversion of the preprocessed frame to HSV form, only the traffic sign objects were separated accurately. Thus, no no-traffic sign objects were separated as traffic sign object, and in videos that do not have traffic sign, no traffic sign objects were detected by the system. Thus, a precision of 100% was obtained in this approach. Also from Table 2, it can be seen that the accuracy of the proposed approach has reached 96.15%, exceeding all of the SVM-based methods and most of the neural network-based methods, due to the HSV-based mask application in the preprocessed frames of the input video. Due to the large number of videos (78) upon which the system was tested for obtaining the results, with those videos being real-time videos of Bangladeshi traffic signs, hence there is little room for doubt on the results’ accuracy and reliability.

Table 1. Confusion Matrix of traffic sign detection of the proposed approach

	Detected Traffic Sign	Detected Non-Traffic Sign
Actual Traffic Sign	69	3
Actual Non-Traffic Sign	0	6



Table 2. Accuracy rates and datasets of previous approaches

SVM-Based Methods		
Paper	Accuracy	Dataset
Chakraborty et. al.[3]	91.73%	Real-time Bangladeshi Traffic Signs
Zhao et. al.[5]	94%	Real-time Chinese Traffic Signs
Sugiharto et. al.[23]	82.01%	Real-time Indonesian Traffic Signs
Rahmad et. al.[7]	86%	Real-time Indonesian Traffic Signs
Ardianto et. al.[6]	94.9%	GTSDb, GTSRB, Linköping University, Real-Time Taiwanese Traffic Signs
Neural Network-Based Methods		
Paper	Accuracy	Dataset
Roxas et. al.[8]	91.45%	Google Street View
Wang et. al.[9]	96%	GTSDb, GTSRB
Filatov et. al.[22]	87%	Real-Time Russian Traffic Signs

#### 5.4. Comparative Analysis

A comparative analysis between the method implemented in this work and the methods implemented in other works has been discussed here. The set of inputs and the dataset used here are different from the other works. This is because the dataset used here is the Dhaka City Traffic Sign dataset and the input used here are video sequences taken in various parts of the city. The comparison of accuracies of the studied research works with the research done in this system is demonstrated in the next two bar diagrams:

From the papers mentioned in Table 2, it is evident that the accuracy hovers within 85-95%. The majority of papers have mentioned using real-time images of various traffic signs for training their model. For many of those researchers, no specific dataset has been mentioned. The accuracy of those works has been no more than 94% and no less than 82%. However, two of the papers used GTSRB[20] and GTSRD dataset. Their range of accuracies were within 87-96%. Similarly, this method implemented a dataset created using Bangladeshi traffic signs and derived an accuracy of 96.15%. The reason being that the papers which didn't use any fixed dataset often used a small number of random pictures to train their model, thus resulting in a low level of accuracy. On the other hand, this system uses a dataset built specifically with Bangladeshi Traffic Signs, consisting of 1000 images divided into 6 types of traffic signs.

Furthermore, through the combination of the useful approaches from the various literary works reviewed by us, we managed to increase our level of accuracy. For example, although [3] has done image segmentation in YCbCr color space, due to our approach using the HSV color space, the accuracy has increased by 4.42%. Aside from that, using YCbCr for image segmentation like [6] and training the SVM model using a specific dataset of Bangladeshi traffic signs have also been important contributing factors, which has led to our achievement of the objectives laid out for this research.

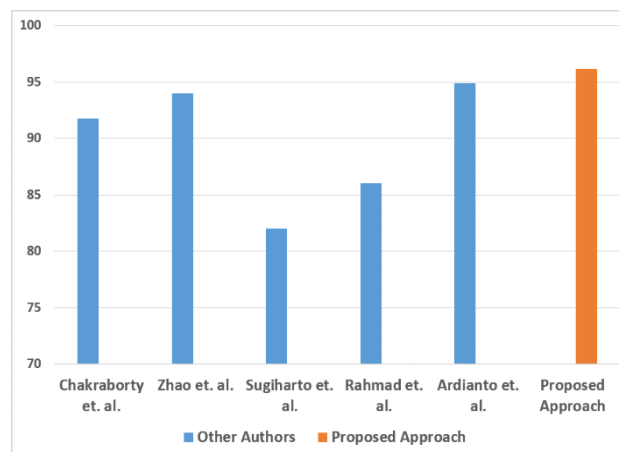


Fig.3. Comparison of proposed approach with SVM-based methods

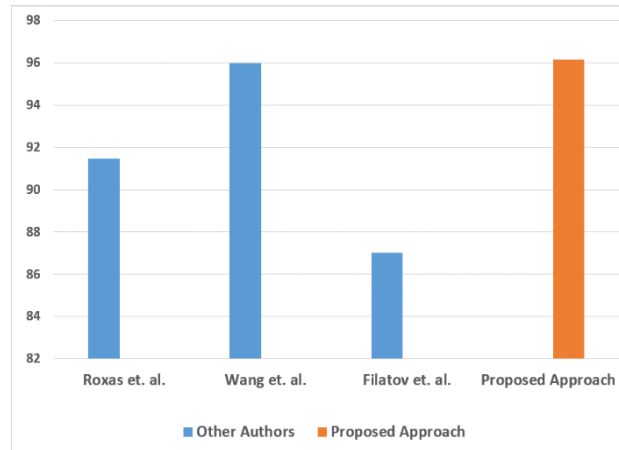


Fig.4. Comparison of proposed approach with Neural Network -based methods

## 6. A Case Study on Detection and Recognition of a Sample Traffic Sign: ‘No Horn’

The result analysis of Bangladeshi Traffic Sign Detection and Recognition shows the proficiency of the proposed approach. However, a step-by-step study may increase the understanding of the approach as well as justify the reason behind the improved results than conventional approaches. Thereby, this chapter provides a phase-to-phase analysis of the detection and recognition of the “No Horn” sign.

To detect and recognize a traffic sign we need to follow our proposed way of Bangladeshi Traffic Sign Detection. We will use the methodology to detect and recognize the “No Horn” sign in the case study.



Fig.5. ‘No Horn’ Sign

The detection of “No Horn” sign consists of two major steps:

- A. Training the model with datasets for the “No Horn” sign.
- B. Detection and recognizing “No Horn” sign.

Each phase is described in the following subsections.

### 6.1. Training the model with the ‘No Horn’ sign dataset

This step consists of some sub-steps. First of them is dataset creation. Before training the model, at first a dataset for the “No Horn” sign is formed. Then comes the part of dataset loading. All the “No Horn” sign images are scanned and loaded into a list while the class itself is loaded onto another list.

Then comes data shuffling and de-skewing. The “No Horn” images are shuffled and de-skewed. This step ensures the uniformness of all images. Next is the calculation of the Histogram of Oriented Gradient (HOG) for the “No Horn” sign. At first the parameters are set for the HOG descriptor. Then the HOG of each “No Horn” image is obtained for the designed HOG model.

After that data splitting occurs. Here the dataset is split into the training and testing portions. The portion of dataset used for training and testing are 90% and 10% respectively. The HOG is also split accordingly. Then the model is generated based on the data. The values of C and gamma are set appropriately to make a compromise between bias and variance. C is set to 12.5 and gamma is set to 0.5. The Radial Basis Function (RBF) is set as the Kernel. The HOG data and labels are also trained in a similar manner.

## 6.2. Detecting and recognizing the 'No Horn' sign

After training the model, the process of "No Horn" sign detection and recognition is started. A video file containing the "No Horn" is used for the purpose of input. Each video file consists of a series of frames. Before initiating the "No Horn" sign detection system, each frame is preprocessed. In the preprocessing stage first the contrast of the frame is increased. The main purpose of contrast increase is to represent the objects of the frame, more prominently. Then Laplacian of Gaussian is applied to remove the noise from the image. After that image is converted into a binary image.

Then comes the step in which small components are removed from the images. In the real life image there are many irrelevant objects in the binary images such as buildings, wires etc.

The next step is application of mask. In this step a mask is generated using the blue, black and white components of the HSV of the original frame. The mask is applied on the refined binary image in the form of the bitwise AND operation. This keeps only the desired color components. The "No Horn" sign that consists of red, white and black color, remains preserved.

The contour of the "No Horn" traffic sign is circular in shape with a diagonal line. To extract this contour, all the contours are iteratively taken, and each contour is checked to find similarity. To verify if the object is a traffic sign, the Euclidean distance of every perimeter pixel is taken. The distances are normalized using the ratio of each distance to the maximum distance. Then the signature value of the image is calculated. The object is classified as a traffic sign if this signature value is equal to or greater than the threshold value. The "No Horn" sign meets this criteria, therefore it is selected. Then the "No Horn" sign is cropped via a bounding box.

Although the "No Horn" sign has been detected, it has not been classified as a "No Horn" sign. For this purpose, the cropped traffic sign object is fed into the model. The model resizes and straightens the image for proper identification. The HOG of the sample image is calculated and if the HOG value matches the "No Horn" sign the most from the dataset then the sign is predicted as "No Horn" sign.

## 7. Conclusion

Traffic signs are symbols placed on the traffic signs to provide instructions and information to the road users. An efficient automated model for detection and recognition of traffic signs is essential at present. This proposed model designs an automated system that can detect and recognize Bangladeshi traffic signs from the video of the surroundings, based on the information obtained from the dataset through machine learning techniques. SVM classifier is used to classify the detected traffic signs based on their HOG property. The model was trained using a dataset of Bangladeshi traffic sign images collected from the streets. It detects traffic signs with an accuracy of 96.15%, significantly higher than the SVM-based techniques used by other authors and slightly higher than the Neural Network based techniques. Also, the proposed system has managed to achieve 100% precision due to an efficient separation technique based on HSV colour scheme and application of mask. Furthermore, compared to previous works, it is capable of detecting traffic signs from video input, rendering it more suitable for practical applications. Thus this model is efficient enough to assist the road users in detection and recognition of the traffic signs.

## References

- [1] A. de la Escalera and M. A. Salichs, "Road traffic sign detection and classification," *IEEE Transactions On Industrial Electronics*, vol. 44, no. 6, 1997
- [2] "Convention on Road Signs and Signals", Vienna, 1968.
- [3] S. Chakraborty and K. Deb, "Bangladeshi road sign detection based on YCbCr color model and DTBs vector," *1st International Conference on Computer and Information Engineering*, 2015.
- [4] <https://en.wikipedia.org/wiki/YCbCr> [Last Accessed: 10 February, 2020]
- [5] J. D. Zhao, Z. M. Bai, and H. B. Chen, "Research on road traffic sign recognition based on video image," *10th International Conference on Intelligent Computation Technology and Automation*, 2017.
- [6] S. Ardianto, C.-J. Chen, and H.-M. Hang, "Real-time traffic sign recognition using color segmentation and svm," *International Conference on Systems, Signals and Image Processing (IWSSIP)*, 2017.
- [7] C. Rahmad, I. F. Rahmah, R. A. Asmara, and S. Adhisuwigno, "Indonesian traffic sign detection and recognition using color and texture feature extraction and svm classifier," *International Conference on Information and Communications Technology (ICOIACT)*, 2018.
- [8] E. A. Roxas, J. N. Acilo, R. R. P. Vicerra, E. P. Dadios, and A. A. Bandala, "Vision based traffic sign compliance evaluation using convolutional neural network," *IEEE International Conference on Applied System Innovation*, 2018.
- [9] W. Canyong, "Research and application of traffic sign detection and recognition based on deep learning," *International Conference on Robots and Intelligent Systems*, 2018.
- [10] Y. Jia, "Caffe: Convolutional architecture for fast feature embedding," 2018.
- [11] P. Jingzhang, "A study on convolutional neural networks for traffic sign recognition," 2017.
- [12] S. Houben, "Detection of traffic signs in real-world images: the german traffic sign detection benchmark," *International Joint Conference on Neural Networks IEEE*, 2013.

- [13] S. Guangming, "Visualization and pruning of ssd with the base network vgg-16," *International Conference on Deep Learning Technologies ACM*, 2017.
- [14] L. Wei, "Ssd: Single shot multibox detector," 2015.
- [15] <https://github.com/Tazmilur20/Bangladeshi-traffic-sign-dataset> [Last Accessed: 7 August, 2020]
- [16] <https://www.jetbrains.com/pycharm/> [Last Accessed: 7 August, 2020]
- [17] <https://www.opencv.org/> [Last Accessed: 7 August, 2020]
- [18] <https://www.numpy.org/> [Last Accessed: 7 August, 2020]
- [19] Dip Nandi, A.F.M. Saifuddin Saif, Prottoy Paul, Kazi Md. Zubair, Seemanta Ahmed Shubho, " Traffic Sign Detection based on Color Segmentation of Obscure Image Candidates: A Comprehensive Study", *International Journal of Modern Education and Computer Science*, Vol.10, No.6, pp. 35-46, 2018.
- [20] <http://benchmark.ini.rub.de/> [Last Accessed: 7 August 2020]
- [21] S. S. Gornale, A. K. Babaleshwar, P. L. Yannawar, "Detection and Classification of Signage's from Random Mobile Videos Using Local Binary Patterns", *International Journal of Image, Graphics and Signal Processing*, Vol.10, No.2, pp.52-59, 2018.
- [22] D.M. Filatov, K.V. Ignatiev, E. V. Serykh, "Neural Network System of Traffic Signs Recognition", *2017 XX IEEE International Conference on Soft Computing and Measurements (SCM)*, 24-26 May 2017
- [23] A. Sugiharto, A. Harjoko, "Traffic sign detection based on HOG and PHOG using binary SVM and k-NN", *2016 3rd International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE)*, 19-20 Oct. 2016
- [24] Priyanka Desai, G. R. Kulkarni, "Use of API's for Comparison of Different Product Information under one Roof: Analysis Using SVM", *International Journal of Information Technology and Computer Science*, Vol.10, No.6, pp.11-22, 2018.
- [25] Abbas Hanon. Alasadi, Baidaa M. ALSafy, "Early Detection and Classification of Melanoma Skin Cancer", *International Journal of Information Technology and Computer Science*, vol.7, no.12, pp.67-74, 2015.

### Authors' Profiles



**Nabil Ahmed** earned his Bachelor of Science (B.Sc.) in Computer Science and Engineering from Ahsanullah University of Science and Technology (AUST) in Dhaka, Bangladesh. His core areas of interest are Machine Learning, Computer Vision and Data Structures.



**Sifat Rabbi** earned his Bachelor of Science (B.Sc.) in Computer Science and Engineering from Ahsanullah University of Science and Technology (AUST) in Dhaka, Bangladesh. His core areas of interest are Competitive Programming, Machine Learning and Computer Vision.



**Md. Tazmilur Rahman** earned his Bachelor of Science (B.Sc.) in Computer Science and Engineering from Ahsanullah University of Science and Technology (AUST) in Dhaka, Bangladesh. His core areas of interest are Machine Learning, Computer Vision and Data Structures.



**Md. Rubel Mia** earned his Bachelor of Science (B.Sc.) in Computer Science and Engineering from Ahsanullah University of Science and Technology (AUST) in Dhaka, Bangladesh. His core areas of interest are Competitive Programming, Machine Learning and Computer Vision.



**Md. Masudur Rahman** is a lecturer of Ahsanullah University of Science and Technology (AUST) in Dhaka, Bangladesh. He earned his Master of Science in Software Engineering (MSSE) and Bachelor of Science in Software Engineering (BSSE) from the Institute of Information Technology (IIT), University of Dhaka, Bangladesh. He has the experiences of working both in industry and academia. He worked as a Software Engineer in KAZ Software Limited (Dhaka, Bangladesh). His core areas of interest are machine learning, information retrieval, code smell and refactoring, software security, and software engineering.

**How to cite this paper:** Nabil Ahmed, Sifat Rabbi, Tazmilur Rahman, Rubel Mia, Masudur Rahman, "Traffic Sign Detection and Recognition Model Using Support Vector Machine and Histogram of Oriented Gradient", International Journal of Information Technology and Computer Science(IJITCS), Vol.13, No.3, pp.61-73, 2021. DOI: 10.5815/ijitcs.2021.03.05