Speed Breakers, Road Marking Detection and Recognition Using Image Processing Techniques

Martins E. Irhebhude, Oladimeji A. Adeyemi, Adeola Kolawole
Department of Computer Science, Nigerian Defence Academy, Kaduna
mirhebhude@nda.edu.ng; oladimejiadeyemi@gmail.com; deolakolawole@yahoo.com

ABSTRACT

This paper presents a image processing technique for speed breaker, road marking detection and recognition. An Optical Character Recognition (OCR) algorithm was used to recognize traffic signs such as "STOP" markings and a Hough transform was used to detect line markings which serves as a pre-processing stage to determine when the proposed technique does OCR or speed breaker recognition. The stopline inclusion serves as a pre-processing stage that tells the system when to perform stop marking recognition or speed breaker recognition. Image processing techniques was used for the processing of features from the images. Local Binary Pattern (LBP) was extracted as features and employed to train the Support Vector Machine (SVM) classifier for speed breaker recognition. Experimental results shows 79%, 100% "STOP" sign and speed breaker recognitions respectively. The proposed system goes very well for the roads which are constructed with proper painting irrespective of their dimension.

Keywords: Speed breaker, Recognition, Intelligent transport system, Hough line transform, Optical Character Recognition, Local Binary Pattern.

1 Introduction

An intelligent transport system (ITS) is a global phenomenon, attracting worldwide interest from transportation professionals, the automotive industry, and political decision makers [1]. ITS provide transport solutions by utilizing state-of-the-art information and telecommunications technologies. It is an integrated system of people, roads and vehicles, designed to significantly contribute to improve road safety, efficiency and comfort, as well as environmental conservation through realization of smoother traffic by relieving traffic congestion according to Andersen and Sutcliffe [1]. Early work in ITS was carried out by the Japanese in the 1980s. At that time, the Japanese had not coined a specific name for ITS as it was considered part of traffic control. They later referred to this work as the Japanese Intelligent Vehicle System (IVS) programme as reported by Andersen and Sutcliffe [1]. Sustainable and efficient transport systems are requirements for economic well-being. ITS holds the promise of sustainability which presents the opportunity for better management of existing resources and infrastructure, through the provision of information to travelers and transportation planning professionals and offers new control possibilities.

Driver Assistance System (DAS) is an important module in ITS. The system is developed to alert a driver or to interact directly on the vehicle for safety and better driving. DAS includes Driver drowsiness detection, Adaptive Cruise Control (ACC), Lane departure warning system, Traffic sign recognition, Wrong-way driving warning, automotive navigation system, and so on [2].

The purpose of Advanced Driver Assistance Systems (ADAS) is that driver error will be reduced or even eliminated, and efficiency in traffic and transport is enhanced. The benefits of ADAS implementations are potentially considerable because of a significant decrease in human suffering, economical cost and pollution [3]. However, there are also potential problems to be expected, since the task of driving an ordinary motor vehicle is changing in nature, in the direction of supervising an or partly automated moving vehicle [3]. The main focused of this work is on obstacle detection on the road i.e. speed breakers.

Speed bumps or speed breakers are designed on road to avoid or reduce over speed. Speed Breakers are traffic handling tool which are used to slow down vehicles speed [4]. Authors stated further that they are called Speed Humps, Speed Bumps, Speed Ramps, Speed Cushions and Speed tables. They are designed to be driven over at a predetermined comfortable speed, while causing exceeding discomfort at higher speeds. The reduction in average vehicular speed significantly improves the safety of people in the neighboring areas [5]. Speed breakers can have different heights, lengths, spacing, signs, etc. In fact, no particular design is suitable for all the types of vehicles using the road. Where speed breakers are permitted to be installed, provide visual and tactile stimuli which alert drivers and cause them to slow down. Speed breakers are typically placed on residential roads. According to survey, speed breakers have predominantly reduced casualty crash frequencies, fatal crashes and pedestrian crashes [5].

According to Rahayu et al [6] speed breaker detection is important to community in order to save the car and human life. Authors in [2] suggested the need to develop a system that services the end user driver using image processing algorithms such as Gaussian filtering, Median filtering and Connected Component Approach etc. The state-of-art algorithm suggested by Devapriya et al in [2] fails on datasets that includes stop markings. Hence there is a need for an algorithm that provides a solution to the state-of-art algorithm problem. In this work, speed breakers with "STOP" traffic marking will be detected and recognized using image processing algorithms such as Local Binary Patten (LBP) and Optical Character Recognition (OCR) with template matching [7].

For clearness of presentation, this article is separated into various segments; immediately following this section is section 2, which reviews existing literatures, section 3 explains the proposed methods for speed breaker detection and recognition, section 4 shows the results of the various experiments performed and performance evaluation and section 5 gives conclusion and future work.

2   Literature Review

Smart cities are faced with the challenges of traffic management and maintenance [8]. Road surface monitoring is currently performed by humans, but the road surface condition is one of the main indicators of road quality, and it may drastically affect fuel consumption and the safety of both drivers and pedestrians. Authors further suggested solution that uses a gyro, an accelerometer, and a GPS sensor mounted on a car. The methodology they suggested had the potentials to detect speed bumps in quasi real-time conditions, and can be used to construct a real-time surface monitoring system.

Due to the bad roads that have caused damages to vehicles and also lead to accident, authors in [9], proposes a system that provides a cost-effective solution to timely alerting to the driver regarding potholes or hump. The proposed system was divided into three sub-units: the sensing sub-unit, the server sub-unit and the user sub-unit. At the sensing sub-unit, an ultrasonic sensor is used to detect potholes and humps, whose location co-ordinates are retrieved by the GPS receiver. This data is stored to the database, which is the sever sub-unit. At the user sub-unit, a hardware module is set-up that provides timely alerts to the drivers regarding potholes and humps.

In [10] a system that detects speed bumps using color segmentation and knowledge base object detection was proposed.

The work done by Afrin et al [11] was to help in preventing accident caused by unmarked speed breakers. They then came up with a system that facilitates autonomous speed breaker data collection, dynamic speed breaker detection and warning generation for the on-road drivers. Their system also incorporates real-time tracking of driver, vehicle and timing information for speed breaker rule violations. The system they proposed outperformed the state-of-the-art works in terms of response time and accuracy.

As reported in [12] a summarized paper on road condition detection using dedicated sensors and smart phones was presented. In the same paper authors highlighted the disadvantage of using smart phones and GPS for speed bump detection. The drawbacks of the using smartphone system are vibration patterns of sensor data, benign events, GPS error, network overload, delay and battery draining. One of the most common methodologies for speed bump detection is using smartphone; the problem arises because of its hard code nature. It is so called, since the detection of speed bump is based on the previous history not based on current scenario so it is unfit for real time scenario.

BLOB analysis technique was used to detect the speed breaker in a given image [13]. Camera fixed in the vehicle was used to capture the image of the road and it was analyzed in real time to detect the presence of speed breaker. This technique demonstrated on painted speed breaker. The technique used by authors proved to be effective because it was designed to alert the driver before the vehicle hit the speed breaker.

Authors in [14] used image processing techniques to detect t h e speed breakers that lie along the roads and to hint the driver to carry on with velocity that makes for vehicle comfort. They came up with a conclusion that using image processing proves to be more effective and precise than using a sensor to detect the speed breakers.

According to Rahayu et al [6], speed breaker detection is important to community in order to save the car and human life. The most critical thing is preventing remedy leading to the cause of accident. Because many speed bumps are constructed without proper permission. Not notifying of speed bump over high speed is harmful for patients in transit, pregnant women, rapid wear and tear and damage to vehicles [2]. They went further to suggest the need to develop a system that services the end user driver using image processing algorithms such as Gaussian filtering, Median filtering and Connected Component Approach. The state-of-art algorithm suggested by Author works fails on datasets that includes stop signs markings. Hence there is a need for an algorithm that provides a solution to the state-of-art algorithm problem. We are basing our research on the methodology proposed by the authors; however, we went further by performing classification of the detected bump and included an OCR algorithm as a way of providing an alternative marking for speed reduction.

2.1 Optical Character Recognition

Optical character recognition (OCR) is the mechanical or electronic translation of images of hand-written or printed text into machine-editable text [15]. According to Mithe et al [16] it is a technology that enables you to convert different types of documents such as scanned paper documents, PDF files or images captured by a digital camera into editable and searchable form. Character recognition system has received considerable attention in recent years due to the tremendous need for digitization of printed documents [17].

The OCR based system consists of image acquisition, image pre-processing, image segmentation, feature extraction, post-processing, and template-matching and correlation techniques [18] [19]. Images captured by a digital camera differ from scanned documents or image. They often have defects such as distortion at the edges and dimmed light, making it difficult for most OCR applications, to correctly recognize the text [16]. In light of that, [20] used OCR to detect text-based traffic signs but in other to improve the accuracy of recognition, the OCR results from several frames were combined together by matching individual words through frames and using a weighted histogram of results, Similarly, authors in [16] used "tesseract" an open source OCR engine to perform the character recognition because of its widespread approbation, its extensibility and flexibility, its community of active developers, and the fact that it "just works" out of the box. Due to the inconsistencies on the fonts used in the marks, OCR template-matching is picked for the proposed system methodology.

2.2   Local Binary Pattern

Local Binary Pattern (LBP) is a non-parametric descriptor, which efficiently labels the pixels of an image with decimal numbers that encode the local structure around each pixel of an image [21] [22]. Perhaps the most important property of the LBP operator in real-world applications is its invariance against monotonic gray level changes caused, for example, by illumination variations. Another equally important property is its computational simplicity, which makes it possible to analyze images in challenging real-time settings [23]. Each pixel ($i.e.$ $g^1, g^2, …, g^8$) is compared with its eight neighbors (see equation 1) by subtracting the center pixel value; the results; if negative, are encoded as 0, and the otherwise 1 (see equation 2). For each given pixel, a binary number is obtained by concatenating all these binary values in a clockwise direction, which starts from the one of its top-left neighbor (see equation 3). The corresponding decimal value of the generated binary number is then used for labeling the given pixel.

LBP can be described as follows:

Pixel neighborhood:

$$\begin{pmatrix} g8 & g1 & g2 \\ g7 & gc & g3 \\ g6 & g5 & g4 \end{pmatrix} \tag{1}$$

Thresholding:

$$\begin{pmatrix} s(g8-gc) & s(g1-gc) & s(g2-gc) \\ s(g7-gc) & & s(g3-gc) \\ s(g6-gc) & s(g5-gc) & s(g4-gc) \end{pmatrix} \text{s(x)} = \begin{cases} 1, x \geq 0 \\ 0, x \geq 0 \end{cases} \tag{2}$$

LBP for pixel:

$$LBP = \sum_{P-0}^{P-1} s(g_P - g_c)2^P \tag{3}$$

2.3   Support Vector Machine

According to Pupale [24] Support Vector Machine (SVM) is a linear model for classification and regression problems. It can solve linear and non-linear problems and work well for many practical problems. SVM can be used for binary and multi classification tasks [22]. The idea of SVM is simple: The
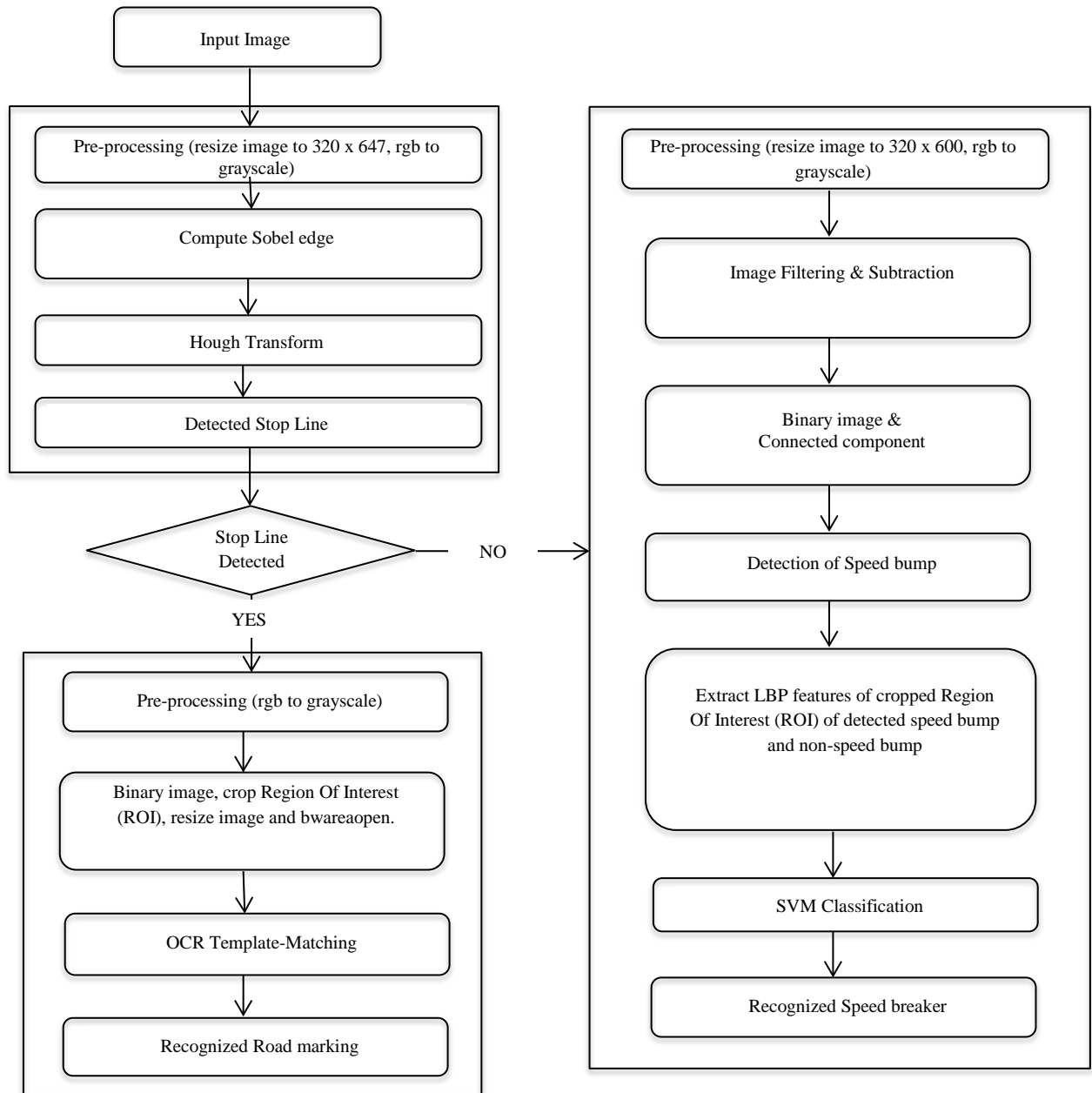
algorithm creates a line or a hyper plane which separates the data into classes. SVM [25] were originally designed for binary classification. How to effectively extend it for multiclass classification is still an ongoing research issue. Currently there are two types of approaches for multiclass SVM. One is by constructing and combining several binary classifiers while the other is by directly considering all data in one optimization formulation. Up to now there are still no comparisons which cover most of these methods. The formulation to solve multiclass SVM problems in one step has variables proportional to the number of classes. Therefore, for multiclass SVM methods, either several binary classifiers have to be constructed or a larger optimization problem is needed [26]. SVM classifier was picked in light of its popularity and ease of usage.

3   Proposed System Methodology

This section introduces the novel technique for speed breaker, road marking detection and recognition, presenting the functionality of each module. As proposed in [2], a methodology that involves 4 stages detected speed bump. Adopting similar method, on our dataset; it failed due to addition road marking "STOP". In the light of this, we will include the OCR algorithm to enable us do a matching on the "STOP" marking. The modified algorithm with an inclusion of OCR and a stop line detection algorithm i.e. Hough transform so as to aid the recognition of the stop marking and also detect stop line before the detection and recognition of the speed breaker (see figure 2).

Figure 1: Sample image from custom dataset of dimension 3264 x 2448

```
                    ┌─────────────────┐
                    │   Input Image   │
                    └─────────────────┘

┌──────────────────────────────────┐    ┌──────────────────────────────────┐
│ Pre-processing (resize image to   │    │ Pre-processing (resize image to   │
│ 320 x 647, rgb to grayscale)      │    │ 320 x 600, rgb to grayscale)      │
│                                    │    │                                    │
│        Compute Sobel edge          │    │     Image Filtering & Subtraction  │
│                                    │    │                                    │
│         Hough Transform            │    │        Binary image &              │
│                                    │    │      Connected component           │
│        Detected Stop Line          │    │                                    │
└──────────────────────────────────┘    │      Detection of Speed bump       │
                                          │                                    │
           ◇ Stop Line Detected ◇  NO →  │  Extract LBP features of cropped   │
                   │ YES                  │  Region Of Interest (ROI) of       │
                                          │  detected speed bump and           │
┌──────────────────────────────────┐    │  non-speed bump                    │
│  Pre-processing (rgb to grayscale) │    │                                    │
│                                    │    │        SVM Classification          │
│  Binary image, crop Region Of      │    │                                    │
│  Interest (ROI), resize image and  │    │     Recognized Speed breaker       │
│  bwareaopen.                       │    └──────────────────────────────────┘
│                                    │
│       OCR Template-Matching        │
│                                    │
│     Recognized Road marking        │
└──────────────────────────────────┘
```

Figure 2: An overview of the proposed system method.

The analysis of the performance of the proposed system for speed breaker detection and recognition was conducted on datasets gathered from a stationary camera of 3264 x 2448 pixel resolution.

3.1 Stop Line Detection

The captured image first undergoes the pre-processing stage. According to Gonzalez and Woods [27] pre-processing is an important procedure which helps to remove the unwanted information like noise and strengthens the required information in an image. In projected method the preprocessing involves; Resize to standard size (320 x 647 pixels) and RGB to Gray scale conversion. A Sobel edge detector is then used to estimate the edges of input image after which Hough transform is applied to detect horizontal lines [28]. The stop line is then detected by using MATLAB function imcrop to crop out points from the horizontal lines that falls in between the stop marking and the painted speed breaker. Imcrop copies a four-element position vector ([xmin ymin width height]) to the clipboard. We then represent the first point as the xmin and last point as the ymin, the width and height is the length and breadth of

the cropped region rectangle. The step taken is demonstrated by the experimental results presented in figure 3.
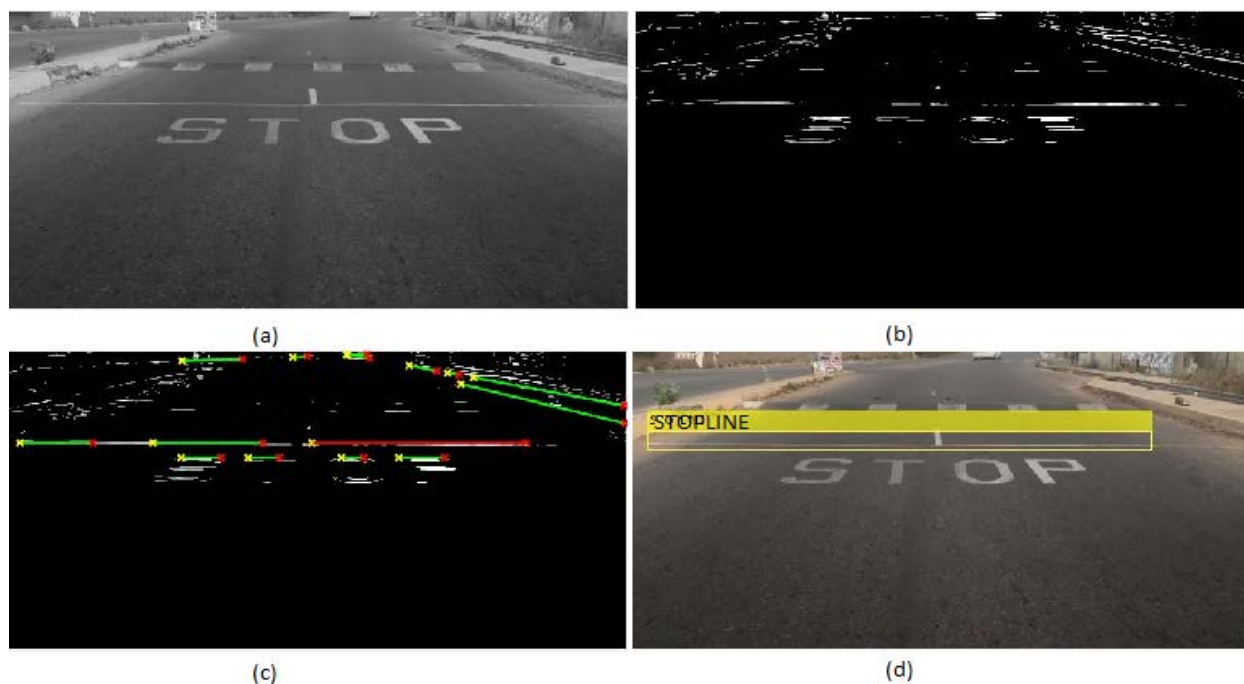
Figure 3: Diagram shows (a) pre-processed image (b) after edge detection (c) after hough transform (d) detected stop line.

3.2 Stop Marking Recognition

The detected stop line determines this stage i.e. if it's detected (stop line), stop marking is then perform on the input image. The first phase here, is to pre-process the input image i.e RGB to Gray scale conversion. We further process the image by converting it to binary using MATLAB functions graythresh,and imbinarize. The Region of Interest (ROI) in the binary image is then cropped and resized to 240 x 420 pixels maintaining aspect ratio, before applying bwareaopen value of 50. An OCR template-matching algorithm is then used on the resultant image to recognize the stop marking. The experimental results are presented in figure 4 below.
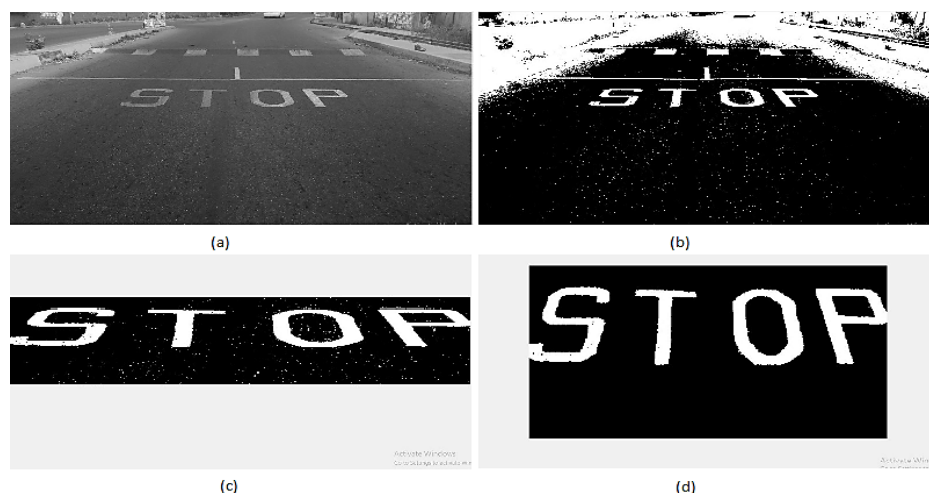
Figure 4: Diagram shows (a) pre-processed image (b) after binary conversion (c) after cropped ROI (d) after resizing.

3.3 Speed Breaker Detection and Recognition

The first stage is preprocessing which prepare the input image before doing the actual process. This is done by resizing the input image to 320 x 600 pixels and also converting it to gray scale image. For the resultant image we apply Gaussian and Median filtering followed by image subtraction. The significance of a low pass Gaussian filtering is to remove noise by blurring the image and remove the high frequency component of the image. Here the environmental noise like sand particle on the road, uneven road conditions is eliminated by the influence of Gaussian filtering. For the Gaussian output we apply median filtering of size 31 × 31. Among the 961 value it choose the median value, the size of the filter is assumed high for easy computation and at the same time we ensure to maintain the required information. The resulting output is added with a number 120 to perform Binary image. The subtracted image is converted to binary image by a simple technique named thresholding [27]. At the end of this stage the image are seen as binary image with highlighting the white region and removing the unwanted region as showed in figure 5c. By viewing the result, there is a possibility of highlight some noise. To remove such noise we move on to next stage called connected component approach. At this stage we apply area open operation of connected component method to stay back with area above a threshold value and remove other noisy region, and from the connected component approach speed bump parameter are computed. In order to recognize and classify speed breakers, we extracted LBP features of cropped ROI of detected speed bump and non-speed bump. The feature set formed the attribute for the classification of speed breakers. SVM was used to classify the speed breakers. The experimental results are presented in figure 5 below.
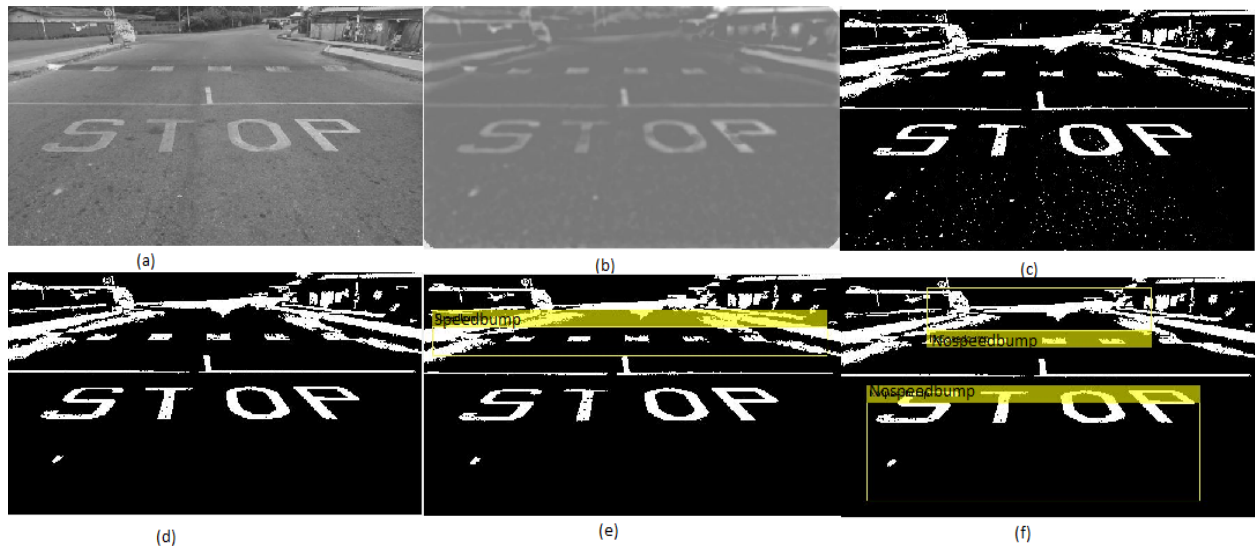

Figure 5: Diagram shows (a) pre-processed image (b) after Image Filtering & Subtraction (c) after Binary image (d) after connected component (e) the detected region of speed bump (f) the detected region of non-speed bump.

4 Experimental Analysis, Results and Discussion

A number of experiments were conducted to evaluate the performance of the proposed algorithm in speed breaker detection and recognition. The experiments were conducted on images retrieved from custom dataset. The dataset consist of 60 images of size 3264 x 2448 pixel, which was taken from a stationary camera of 8mega pixels at daytime. The camera was positioned in the center of the road, at each location used for data gathering. For the classification of the speed breaker, the set of input-output sample pairs that are used for the training of the classifier can be represented as,

$$(x_i, y_j), (x_i, y_j), \ldots, (x_k, y_k) \tag{4}$$

where the input x_i denotes the feature vector extracted from image I and the output y_j is the class label. Since we are categorizing into speed breaker and non-speed breaker, the class label y_j encodes the speed breaker and non-speed breaker; while the extracted feature x_i encodes LBP features respectively. The dataset consisted of approximately 60 images, the region with the detected speed breakers and non-speed breakers were used for the purpose of training and testing (see figure 5).

Figure 6: Diagram shows validated result of the detected region of (a) speed bump (b) non-speed bump.

After carrying out a number of experiments to evaluate the performance of the proposed algorithm. The stop marking recognition has an accuracy of 79.07% due to the inconsistencies on the fonts used in the marks. We obtained the result by calculating the number of characters correctly recognized and the number of predicted characters from the 10 randomly selected images (see table 1).

The accuracy of performance was defined as:

$$\frac{CC}{PC} \times 100 \tag{5}$$

Where CC; Correct Characters and PC; Predicted Characters.

Table 1: Detailed information about result obtained from OCR.

Serial No.	Characters Present	Predicted characters	Correct characters
1	4	4	4
2	4	4	4
3	4	4	4
4	4	4	4
5	4	4	4
6	4	4	4
7	4	4	4
8	4	4	1
9	4	5	3
10	4	6	2
TOTAL	40	43	34

The classification accuracy rate obtained after using the LBP feature extractor and SVM was 100%. The confusion matrix (Figure 7) and (Table 2) explains the level of accuracy graphically. Where class 1; speed breaker and class 2; non speed breaker.
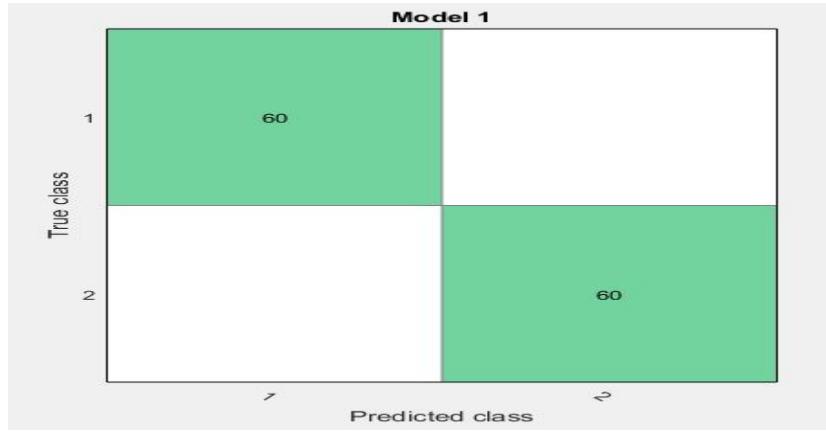
Figure 7: Confusion matrix for LBP feature extractor.

Table 2: Confusion matrix for LBP feature extractor

	Feature	Class 1	Class 2	Correct Prediction		Wrong Prediction	
				Class 1	Class 2	Class 1	Class 2
1	LBP	60	60	60	60	-	-

For the purpose of detailed analysis of the performance of the proposed approach, the classification performance is evaluated using the receiver operating characteristic (ROC) curve (figure 8) [22]
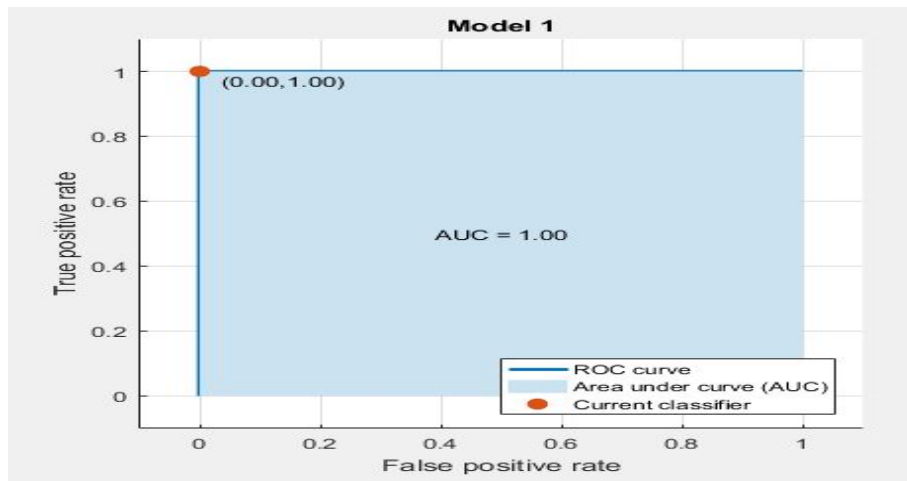

Figure 8: ROC curve for LBP feature extractor.

From the ROC plot (figure 8), we obtained a perfect result due to the fact that our ROC curve as no misclassified points and its denoted by a right angle to the top left of the plot.

The state-of-art algorithm suggested by Devapriya et al in [2] was based on detecting speed bump (figure 9) which aids us to propose a system that works on datasets that includes stop markings.

Figure 9: State-of-art algorithm suggested by Devapriya et al in [2].

5 Conclusion and Future Work

This paper proposed an image processing technique for speed breaker detection and classification with the inclusion of OCR for stop marking and a stop line detection in daytime images. The stopline inclusion serves as a pre-processing stage that tells the system when to perform stop marking recognition or speed breaker detection and recognition. The features for the classification were extracted by applying LBP extractor on images. SVM was used to trained the features obtained from the custom datasets (speed breakers and non-speed breakers). The proposed method were implemented and the experimental results showed 79.97% accuracy for OCR and 100% accuracy for LBP. The future scope of the proposed work is recognition of speed breakers in night vision.

REFERENCES

[1]   Andersen, J., & Sutcliffe, S. (2000). Intelligent Transport Systems (ITS) - An Overview. *IFAC Proceedings Volumes, 33*(18), 99-106.

[2]   Devapriya, W., Nelson Kennedy Babu, C. and Srihari, T. (2016) Real Time Speed Bump Detection Using Gaussian Filtering and Connected Component Approach. *Circuits and Systems*, **7**, 2168-2175.

[3]   Brookhuis, K. A., Waard, D. D., & Janssen, W. H. (2001). Behavioural impacts of Advanced Driver Assistance Systems–an overview. *EJTIR, 3*(1), 245-253.

[4]   Devapriya, W., Babu, C. N., & Srihari, T. (2015). Advance Driver Assistance System (ADAS) - Speed bump detection. *2015 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC),* 478-483.

[5]   Reddy, V. K., & Nagesh, B. S. (2016). Smart Phone Based Speed Breaker Early Warning System. *International Journal of Computer Science and Information Security (IJCSIS), 14*, 20-25.

[6]   Rahayu, E., Faizal, M., & Ani, Z. C. (2016). A Vision Based Speed Breaker Detection for Early Warning Notification. 43-47.

[7]   Optical Character Recognition (OCR) - File Exchange - MATLAB Central. Retrieved October 18, 2018, from https://www.mathworks.com/matlabcentral/fileexchange/18169-optical-character-recognition-ocr

[8]   Celaya-Padilla, J., Galván-Tejada, C., López-Monteagudo, F., Alonso-González, O., Moreno-Báez, A., Martínez-Torteya, A., . . . Gamboa-Rosales, H. (2018). Speed Bump Detection Using Accelerometric Features: A Genetic Algorithm Approach. *Sensors, 18*(2), 443.

[9]   P, M., Singh, S., Shukla, S., & Krishnan, U. (2017). detection of humps and potholes on roads and notifying the same to the drivers. *International Journal of Management and Applied Science, 3*(1), 130-133.

[10]   Danti, A., Dr, Kulkarni, J. Y., Smt, & Hiremath, P. S., Dr. (2013). A Technique for Bump Detection in Indian Road Images Using Color Segmentation and Knowledge Base Object Detection. *International Journal of Scientific & Engineering Research, 4*(8).

[11]   Afrin, M., Mahmud, M. R., & Razzaque, M. A. (2015). Real time detection of speed breakers and warning system for on-road drivers. *2015 IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE),* 495-498.

[12]   Chugh, G., Bansal, D. and Sofat, S. (2014) Road Condition Detection Using Smartphone Sensors: A Survey. *International Journal of Electronic and Electrical Engineering*, **7**, 595-601.

[13]   Manikandan , B., & Bharathi, M. (2018). SPEED BREAKER DETECTION USING BLOB ANALYSIS. *International Journal of Pure and Applied Mathematics*, *118*(20), 3671–3677.

[14]   Anusha, K. & Sirisha, K. (2011) "Breaking the Speed Breakers Using Image   Processing" Retrieved from https://www.scribd.com/document/54957257/Breaking-the-Speed-Breakers-Using-Image-Processing-1-2

[15]   Sagar, B. M., G, Shobha., & P. Ramakanth Kumar. (2008). OCR for printed Kannada text to Machine editable format using Database approach. *WSEAS Transactions on Computers*, *7*(6), 766–769.

[16]   Mithe, R., Indalkar, S., & Divekar, N. (2013). Optical Character Recognition. *International Journal of Recent Technology and Engineering*, *2*(1), 72–75.

[17]   Sadasivan, A. K., & Senthilkumar, T. (2012). Automatic Character Recognition in Complex Images. *Procedia Engineering*, *30*, 218–225.

[18]   Tiwari, S., Mishra, S., Bhatia, P., & Yadav, P. K. (2013). Optical Character Recognition using MATLAB. *International Journal of Advanced Research in Electronics and Communication Engineering*, *2*(5), 579–582.

[19]   Singla, S., & Yadav, R. (2014). Optical Character Recognition Based Speech Synthesis System Using LabVIEW. *Journal of Applied Research and Technology*, *12*(5), 919–926.

[20]   Greenhalgh, J., & Mirmehdi, M. (2015). Recognizing Text-Based Traffic Signs. *IEEE Transactions on Intelligent Transportation Systems*, *16*(3), 1360–1369.

[21]   Huang, D., Shan, C., Ardabilian, M., Wang, Y., & Chen, L. (2011). Local Binary Patterns and Its Application to Facial Image Analysis: A Survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *41*(6), 765–781.

[22]   Irhebhude, M. E., Nawahda, A., & Edirisinghe, E. A. (2016). View invariant vehicle type recognition and counting system using multiple features. *International Journal of Computer Vision and Signal Processing, 6*(1), 20-32.

[23]   Brahnam, S., Jain, L. C., Lumini, A., & Nanni, L. (2013). Introduction to Local Binary Patterns: New Variants and Applications. *Local Binary Patterns: New Variants and Applications Studies in Computational Intelligence*, 1–13.

[24]   Pupale, R. (2019, February 11). Support Vector Machines(SVM) - An Overview. Retrieved August 9, 2019, from https://towardsdatascience.com/https-medium-com-pupalerushikesh-svm-f4b42800e989

[25]   Cortes, C., & Vapnik, V. (1995). Support-Vector Networks. *Machine Learning*, *20*, 273–297.

[26]   Hsu, C.-W., & Lin, C.-J. (2002). A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, *13*(2), 415–425.

[27]   Gonzalez, C. & Woods, R.E. (2013) Digital Image Processing, Person. 3rd Edition, Person, New Delhi, 635, 738.

[28]   Marita, T., Negru, M., Danescu, R., & Nedevschi, S. (2011). Stop-line detection and localization method for intersection scenarios. *2011 IEEE 7th International Conference on Intelligent Computer Communication and Processing*, 293–298.