Vehicle Detection in Fog and Night Condition using Block Matching and Color Enhancement Algorithm

Nivedha M,Pallavi R,Supriya S,Sushma R
Final year B.E Students(UG)
Guided by: Mrs. R Suganya, Assistant Professor Department of ISE, TOCE
The Oxford College of Engineering, Bengaluru ,India

Abstract: Fog is a big reason for road accidents. Images which are captured under bad weather conditions suffer low contrast so as their quality also degrade with the changes in atmosphere. The main reason behind this problem is that, the light capture by the lens is spread by the atmosphere. Fog reduces visibility to less than 1 kilometre. This paper aims to resolve the sight problem faced by car/other automobile drivers when driving in foggy weather condition and at night. For improving the visibility level of an image and reducing fog, various image enhancement methods are used. For improving the visibility level five major steps are used. First step is acquisition process of foggy / dark images. Second is estimation process. Third is enhancement process (improve visibility level, reduce fog). Next process is restoration process (restore enhanced image) while the final is the vehicle detection. The main aim of the paper is to review image enhancement and restoration methods for improving the quality and visibility level of an image which provide clear image in bad weather condition.

Index Terms—restoration process, foggy, estimation process, image enhancement ,automatic braking.

1. INTRODUCTION

The images of outdoor scenes are usually degraded by the turbid medium (e.g., particles and water droplets) in the atmosphere. Haze, fog and smoke are such phenomena due to atmospheric absorption and scattering. Light from the atmosphere and light reflected from an object are scattered by the water droplets, resulting the visibility of the scene to be degraded. The two fundamental phenomena that are consequence of scattering are 'attenuation' and 'airlight'. Fog removal is a difficult task because fog depends on the unknown scene depth map information. Fog effect is the result of distance between camera and object. Hence removal of fog requires the estimation of airlight map or depth map.
The current fog removal method can be divided into two categories:
(a) Image enhancement and
(b) Image restoration.
Image processing is a method to perform some operations on an image, in order to get an enhanced image or to extract some useful information from it. It is a type of signal processing in which input is an image and output may be image or characteristics/features associated with that image. Nowadays, image processing is among rapidly growing technologies. It forms core research area within engineering and computer science disciplines too.

Image processing basically includes the following three steps:

- Importing the image via image acquisition tools.
- Analysing and manipulating the image.
- Output in which result can be altered image or report that is based on image analysis.

There are two types of methods used for image processing namely, analogue and digital image processing. Analogue image processing can be used for the hard copies like printouts and photographs. Image analysts use various fundamentals of interpretation while using these visual techniques. Digital image processing techniques help in manipulation of the digital images by using computers. The three general phases that all types of data have to undergo while using digital technique are pre-processing, enhancement, and display, information extraction. Improving the performance of vehicle-detection in different weather conditions becomes an important issue in vehicle-detection system because in the case of fine weather, the system would achieve good performance but however when it comes to bad weather like foggy environment and night condition the performance of many systems are not appreciable. The proposed method provides image enhancement and fog removal. Because of scarceness of data that are available in the foggy image, quality of image lower. Hence it is very much essential to make the image appropriate to human sight distinctive. Firstly we have to retain true color and striking differences of foggy image and retain the original image to obtain a clear image. Due to the lack of illumination at night, light will be from either other vehicles or from any light providing sources. So the brightness of image will be indistinct and contrast will be literally less, most of the information are not perceivable by human eye. Images color will be the color of the light providing or light causing devices. Image's original color will be misrepresented. So it is very much necessary to enhance the image's characteristics to

make better the execution of vehicle detection. In proposed system we make use of kalman filter for vehicle detection. Block matching algorithm is used for breaking images into fine blocks according to the type of image.

2. LITERATURE REVIEW

As Sensors are sensitive to weather conditions; video cameras could be used to record the traffic information at different weather conditions. We have sophisticated algorithms to analyze the traffic videos in real time and discover information of interest. Although some sensors could be more accurate, they could also be Intrusive and need a higher maintenance cost. We may need to embed weighing sensors in road to measure vehicle feature and classify vehicle size. In video surveillance systems, it is very complicated to extract more number of features from a video. It has been also inferred that more computations are required to calculate background model and to extract the key frames. In this paper, a novel algorithm is implemented which counts and classifies highway vehicles using regression. The algorithm proposed in this paper, starts with preprocessing the Low Quality videos by Removal of Noise using Bi-lateral filtering, followed by color image based Background mask generation using Multi-layer Background subtraction technique[1]. A fast performing kernel is designed which then used to extract the Foreground mask using Mixture of Gaussians. Finally Contour extraction and Cascaded Regression will results the foreground moving objects in the Low quality video.

A vehicle detection and counting system plays an important role in an intelligent transportation system, especially for traffic management. This paper proposes a video-based method for vehicle detection and counting system based on computer vision technology. The proposed method uses background subtraction technique[2] to find foreground objects in a video sequence. In order to detect moving vehicles more accurately, several computer vision techniques, including thresholding, hole filling and adaptive morphology operations, are then applied. Finally, vehicle counting is done by using a virtual detection zone. Experimental results show that the accuracy of the proposed vehicle counting system is around 96%.

Intelligent transportation systems have received a lot of attention in the last decades. Vehicle detection is the key task in this area and vehicle counting and classification are two important applications. In this study, the authors proposed a vehicle detection method which selects vehicles using an active basis model and verifies them according to their reflection symmetry. Then, they count and classify them by extracting two features: vehicle length in the corresponding time-spatial image and the correlation computed from the grey-level co-occurrence matrix of the vehicle image within its bounding box. A random forest is trained to classify vehicles into three categories: small (e.g. car), medium (e.g. van) and large (e.g. bus and truck). The proposed method is evaluated using a dataset including seven video streams which contain common highway challenges such as different lighting conditions, various weather conditions, camera vibration and image blurring[3]. Experimental results show the good performance of the proposed method and its efficiency for use in traffic monitoring systems during the day (in the presence of shadows), night and all seasons of the year.

Estimating the number of vehicles present in traffic video sequences is a common task in applications such as active traffic management and automated route planning. There exist several vehicle counting methods such as Particle Filtering or Headlight Detection, among others. Although Principal Component Pursuit (PCP) is considered to be the state-of-the-art for video background modeling, it has not been previously exploited for this task. This is mainly because most of the existing PCP algorithms are batch methods and have a high computational cost that makes them unsuitable for real-time vehicle counting. In this paper, we propose to use a novel incremental PCP-based algorithm[4] to estimate the number of vehicles present in top-view traffic video sequences in real-time. We test our method against several challenging datasets, achieving results that compare favorably with state-of-the-art methods in performance and speed: an average accuracy of 98% when counting vehicles passing through a virtual door, 91% when estimating the total number of vehicles present in the scene, and up to 26 fps in processing time.
Vehicle detection has been applied in many fields, such as intelligent transportation, video surveillance, driving assistance system and so on. In the case of fine weather, the state-of-the-art vehicle-detection systems may achieve good performance. However, the performance has a substantial decline in bad weathers, such as fog, night and so on. Therefore, improving the performance of vehicle-detection systems in different weather conditions becomes an important issue in vehicle-detection system[5]. In the fog or night, the quality of the images is reduced. In this paper, we propose some algorithms of image defogging and color enhancement in order to improve the performance of vehicle detection. The result of vehicle detection get much better after image processing in bad weathers.

3. METHODOLOGY

A. IMAGE ACQUISITION MODULE:

The program performs pre-processing of the image.

Step 1: Loading Image

Internal storage consists of number of images such as night, fog and normal images loaded from different sources. Few of these images from the internal storage are sent to the database.

Step 2: Extract RGB Component

Here the extraction of red, green, blue channels of a color image is been done.

1. Obtain the Red color plane of the image and display the red index of the image.
2. Obtain the Green color plane of the image and display the green index of the image.
3. Obtain the Blue color plane of the image and display the blue index of the image.

B. FEATURE EXTRACTION MODULE:

Features like mean and amount of white pixels in the image is extracted which helps in classification of images.

The program performs feature extraction of the image to identify if image is too foggy or less.

If the input image is colored, it should be converted to grey scale to perform computation on the grey scale image. The original version of the operator labels the image pixels by thresholding the 3 × 3 neighborhood of each pixel with the center value and summing the thresholded values. For this module input is gray scale image and output is features of the image.

- **BLOCK MATCHING ALGORITHM**

Step1: A block matching algorithm involves dividing the current frame of a video into macroblocks and comparing each of the macroblocks with a corresponding block and its adjacent neighbors in a nearby frame of the video (sometimes just the previous one).

Step2: A vector is created that models the movement of a macroblock from one location to another. This movement, calculated for all the macroblocks comprising a frame, constitutes the motion estimated in a frame.

Step3: The search area for a good macroblock match is decided by the 'search parameter', p, where p is the number of pixels on all four sides of the corresponding macro-block in the previous frame. The search parameter is a measure of motion. The larger the value of p, larger is the potential motion and the possibility for finding a good match. A full search of all potential blocks however is a computationally expensive task. Typical inputs are a macroblock of size 16 pixels and a search area of p = 7 pixels.

C. MACHINE LEARNING:

This module accepts the test image as input and matches it to one of the classes present in the training dataset, based on the feature values extracted from the input image. It classifies each row in test and returns the predicted class level group. Test must have the same number of columns as the data used to train the classifier in neural network, label sec indicates the group to which each row of test is assigned. Based on that the result is assigned to test variable. For this module, input the test image and trained dataset features and output is category of input image.

D. DENOISING MODULE:

Input will be gray scale or color image. The program performs feature denoising of the image so as to convert foggy image into better visibility. Output will be defogged image.

E. IMAGE ENHANCEMENT MODULE:

The image qualities of captured outdoor scenes are usually degraded due to bad weather such as fog, haze, smog, cloud and rain. Bad weather reduces visibility and contrast of the scene. The program performs image enhancement for better visibility. Input is gray scale or color imageand output is enhanced image.

IMAGE ENHANCEMENT ALGORITHM

Algorithm 1 Adaptive Low-light Image Enhancement

Input: A low-light image I.
Output: An enhanced image E.

1: Apply superpixel segmentation on I, calculate α_{p_i} by Eq (1);
2: Reverse the image I to get the image R;
3: Apply BM3D filter on R in two scales to get $b^{fine}(R)$ and $b^{coarse}(R)$, then combine them as Eq (3) to get the base layer $b(R)$;
4: Apply first order differential on R to generate the noised detail layer $d_0(R)$;
5: Using the structural filter to smooth $d_0(R)$ to generate the noise-free detail layer $d(R)$;
6: Adaptively combine the noise-free detail layer $d(R)$ and the base layer $b(R)$ to obtain R' according the parameter α by Eq (2);
7: Estimate the global atmosphere light A using R';
8: Calculate the enhancement parameter $\omega(x)$ for each pixel as Eq (6);
9: Estimate the transmission parameter $t(x)$ as Eq (5);
10: Update $t(x)$ by using $P(x)t(x)$;
11: Generate the dehazed image J by Eq (9) using R';
12: Generate the final output E by reverse image J.
13: **return** E;

We firstly utilize the superpixel method to split the low-light image I into patches. For each patch we use the following method to determine the smoothing degree, assuming the noise is additive white Gaussian noise (AWGN). We use σpi to denote the standard deviation and gpi to denote the local gradients of a superpixel pi . Experimental observations show that the gpi in the flat patch increases greatly when AWGN is added into a clear image. Whereas, the gpi does not change a lot in the textural patch. On the other hand, for the normalized image in the range [0, 1], the patch standard deviation σpi varies in an order of magnitude. So we consider the normalized ratio αpi between σpi and gpi to measure the patch noise-texture level as follows,

$$\alpha_{p_i} = \frac{\sigma_{p_i}}{g_{p_i}}. \qquad (1)$$

Based on the measurement of noise-texture level of each patch, we can adaptively apply our denoising algorithm. To facilitate denoising and utilize the dark channel prior dehazing algorithm for contrast enhancement, we invert the input image I using R = 255−I. Enlightened by the unsharp masking filter , we define the the denoised R as R0 , and R0 is obtained by the weighted combination of the base layer and the denoised detail layer of R.

$$R' = \alpha \cdot d(R) + b(R), \qquad (2)$$

where d(R) and b(R) denote the noise-free detail layer and the base layer of R respectively. For a patch with small α, we add few details to constrain the noise degree. While, for a patch with large α, we add more details to the base layer. One good technique to get the base layer of an image is to smooth it using the BM3D filter , which can effectively attenuate AWGN. We utilize the noise-texture level coeffi- cient α as a weight in generating the base layer.

$$b(R) = \alpha \cdot b^{fine}(R) + (1-\alpha) \cdot b^{coarse}(R), \qquad (3)$$

where b f ine(R) and b coarse(R) respectively denote the smoothed result of the BM3D filter using a parameter half smaller and twice greater than the mean of the local standard deviation σpi of the observed image I. To obtain the detail layer d0(R), we simply calculate the first order differential of the inverted image R. We find that the random noise tends to fuse with texture in the detail layer d0(R). So it is necessary to choose an appropriate algorithm to smooth the detail layer, while retaining useful texture we apply the structure smooth to the detail layer d0(R) to obtain a smooth and texture preserved result d(R). Finally, we adaptively add the smoothed detail layer d(R) back to the base layer b(R) to get a noise-free and texture preserved image R0 , as Equation (2).
Since the noise-free image R0 is similar to the hazy image, we utilize efficient haze removal method to enhance its contrast. The algorithm is based on,

$$R' = t \cdot J + (1-t) \cdot A, \qquad (4)$$

where A is the global atmospheric light. J is the intensity of the original objects or scene without hazy depravation. t describes the percent of the light emitted from the objects or scene that reaches the camera. t is estimated using,

$$t(x) = 1 - \omega \cdot \min_{c \in \{r,g,b\}} \left(\min_{y \in \Omega(x)} \left(\frac{R'^c(y)}{A^c} \right) \right), \qquad (5)$$

where Ω(x) is a local block centered at pixel x and the block size is 3 × 3 I . ω is a weight coefficient, which is 0.8 , to control the enhance degree.

$$\omega(x) = \left(1 - 10^{-\sqrt{\frac{\sum_{c \in \{r,g,b\}} (255 - I(x))^c}{3}}}\right)^2, \qquad (6)$$

where I is the intensity of the input low-light image, and c denotes the color channels. By applying the Equation (6), the weight coefficient ω is reduced when the pixel x is bright, and increased when the pixel x is dark. This adaptive adjustment can efficiently alleviate over-enhancement and under enhancement. We utilize the following process to estimate global atmosphere light A. To avoid the negative influence of random texture, we first smooth the R0 with a 5 × 5 average filter, then we select the pixels whose minimum intensities in all color (RGB) channels are the 2% highest of all the pixels in the image. Among these pixels, we choose the pixel whose sum of RGB values is the highest. The RGB values of this selected pixel are used to represent the RGB values of the atmosphere A. Thus, according to Equation (4), we can recover the J by,

$$J = \frac{R' - A}{t} + A. \qquad (7)$$

However, direct using of Equation (7) might lead to underenhancement for dark areas. To further optimize t, we introduce a multiplier P into Equation (7), and through extensive experiments, we find that P can be set as,

$$P = \begin{cases} 2t, & 0 < t < 0.5 \\ 1, & 0.5 < t < 1 \end{cases}, \qquad (8)$$

then the recovery equation becomes,

$$J = \frac{R' - A}{t \cdot P} + A. \qquad (9)$$

F. VEHICLE DETECTION MODULE:

The program performs vehicle detection . For this module, we input a video and image is detected as output.

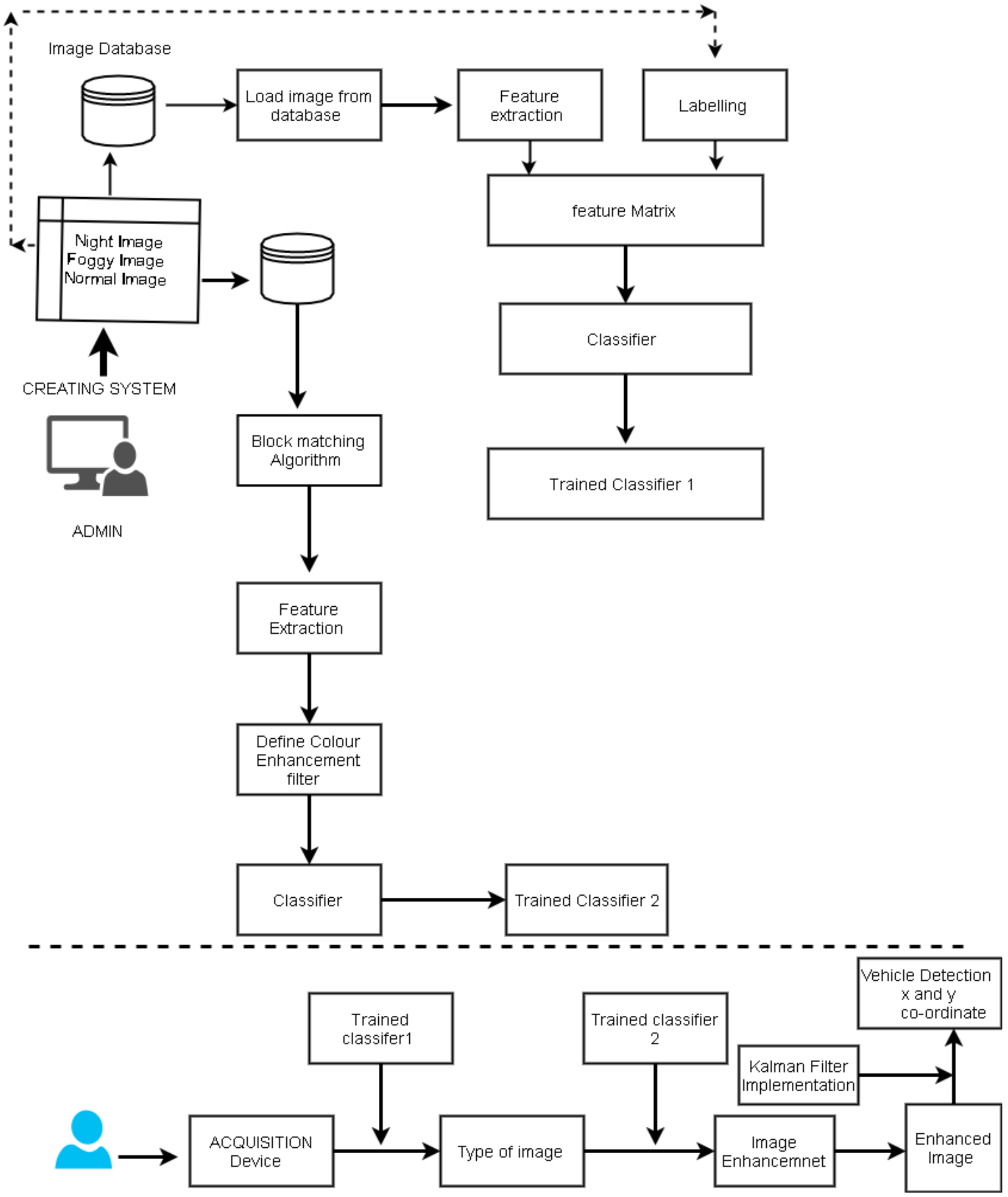

Figure 1: Architecture diagram of vehicle detection

EXPERIMENTAL RESULTS

The figure shown below gives the step by step operational output for our proposed system.

Defogging: The figure 2(a) and 2(b) demonstrates fog removal using the defog function. the picture on the left is taken on a foggy day with the defog function off. the picture on the right is the same scene taken with the defog function on. both pictures were captured with a camera. the defog feature adjusts contrast, color, and sharpness.

Figure 2(a): defogged image1

Figure 2(b): defogged image2

Color Enhancement: The purpose of color enhancement is to get finer details of an image and highlight the useful information. During the poor illumination condition the image appear darker or with lower contrast such image needs to be enhanced. Examples are shown below in figure 3(a) and 3(b).

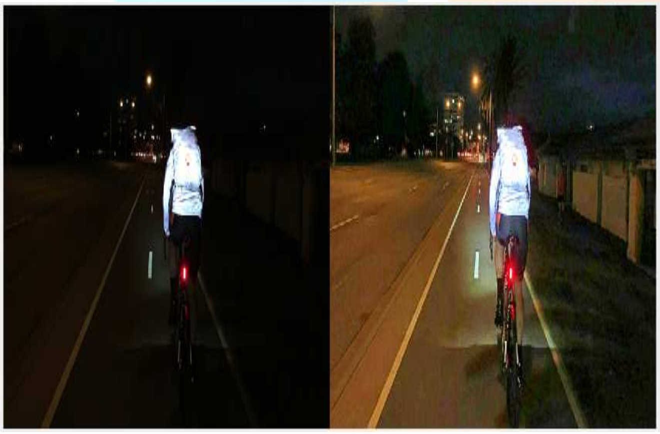

Figure 3(a): color enhancement image1

Figure 3(b): color enhancement image2

Vehicle Detection: The detection of moving objects regions of change in the same image sequence which captured at different intervals,which also includes includes moving or motion vehicle detection and segmentation approach.

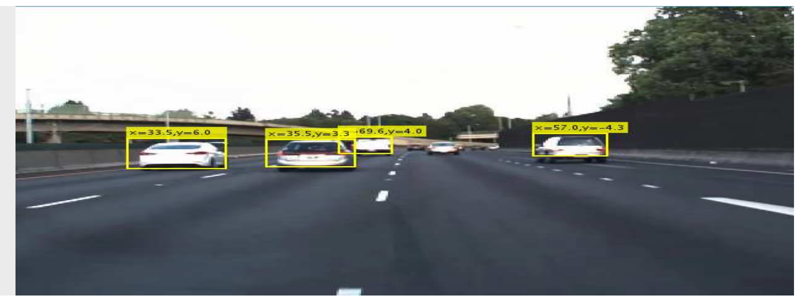

Figure 4: Vehicle detection

CONCLUSION

In the proposed system we choose algorithms for defogging the image, color enhancement prior to vehicle detection. The algorithm used for defogging will make the image's quality better. Color enhancement algorithm allows to increase visibility nature of scene.. This makes the image more appropriate to human sight. The proposed system can enhance the performance of vehicle detection and vehicle visibility in bad weathers.

FUTURE WORK

The system proposes a method which is able to detect vehicle but in future scope the vehicle braking system could be employed so as to provide a better ADAS (advanced driver-assistance system) system which is capable of not only detecting the vehicle but also providing braking assistance. The system can also be integrated with hardware model such that a real time model can be demonstrated with the existing algorithm.

REFERENCES

[1] B. G. Rajagopal, N. Vishakraj, N. U. Kumar and P. Jothivenkatesh, " Vision-based system for counting of moving vehicles in different weather conditions," 2017 International conference of Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, 2017,pp.86-91.doi: 10.1109/ICECA.2017.8203649.

[2] Ohnishi, "A computer vision based vehicle detection and counting system," 2016 8th International Conference on Knowledge and Smart Technology (KST), Chiangmai, 2016, pp.224-227.doi: 10.1109/KST.2016.7440510.

[3] S. Kamkar and R. Safabakhsh, "Vehicle detection, counting and classification in various conditions," in IET Intelligent Transport Systems, vol. 10, no. 6, pp. 406-413,82016.
doi: 10.1049/iet-its.2015.0157.

[4] J. Quesada and P. Rodriguez, "Automatic vehicle counting method based on principal component pursuit background modeling," 2016 IEEE International Conference on Image Processing(ICIP),Phoenix,AZ,2016,pp.3822-3826.doi: 10.1109/ICIP.2016.7533075.

[5] Xuerui Dai, Xue Yuan, Jing Zhang, Liping Zhang, "Improving the performance of vehicle detection system in bad weathers," 2016 IEEE.

[6] Kamkar, S., Safabakhsh, R.: 'Vehicle detection, counting and classification in various conditions', IET Intell. Transp. Syst., 2016, 10, (6), pp. 406–413.

[7] G. N. Swamy and S. Srilekha, "Vehicle detection and counting based on color space model," 2015 International Conference on Communication.

[8] P. Valiere, L. Khoudour, A. Crouzil and D. N. T. Cong, "Robust vehicle counting with severe shadows and occlusions," 2015 International Conference on Image Processing Theory, Tools and Applications (IPTA), Orleans, 2015, pp. 99-104.unications and Signal Processing(ICCSP),Melmaruvathur,2015,pp.0447-0450.doi: 10.1109/ICCSP.2015.7322928.

[9] Liu, Y., Tian, B., Chen, S., et al.: 'A survey of vision-based vehicle detection and tracking techniques in ITS'. IEEE Int. Conf. Vehicular Electronics and Safety (ICVES), Dongguan, China, July 2013, pp. 72–77.

[10] Li, Y., Wang, F.Y., Li, B., et al.: 'A multi-scale model integrating multiple features for vehicle detection'. Int. IEEE Conf. Intelligent Transportation Systems (ITSC), The Hague, The Netherlands, May 2013, pp. 399–403.

[11] Held, D., Levinson, J., Thrun, S.: 'A probabilistic framework for car detection in images using context and scale'. IEEE Int. Conf. Robotics and Automation (ICRA), Saint Paul, USA, June 2012, pp. 1628–1634.

[12]Wu, K., Xu, T., Zhang, H.: 'Overview of video-based vehicle detection technologies'. 6th Int. Conf. Computer Science & Education (ICCSE), Singapore, Singapore, August 2011, pp. 821–825.

[13] Atiq, H.M., Farooq, U., Ibrahim, R., et al.: 'Vehicle detection and shape recognition using optical sensors: a review'. Second Int. Conf. Machine Learning and Computing (ICMLC), Bangalore, India, February 2010, pp. 223–227.

[14] Tsai, L.W., Hsieh, J.W., Fan, K.C.: 'Vehicle detection using normalized color and edge map', IEEE Trans. Image Process., 2007, 16, (3), pp. 850–864.

[15] Jia, Y., Zhang, C.: 'Front-view vehicle detection by Markov chain Monte Carlo method', Pattern Recognit., 2009, 42, (3), pp. 313–321.