Classifying and counting the number of vehicles in lane

Dhruv Jagdishkumar Patel Tirth Jitendra Patel
University of Maryland Baltimore County
CMSC 491/691 Final Report
Spring 2021, Group 10

Abstract-
Nowadays, the number of vehicles moving on roads has increased significantly. Due to which managing the traffic and providing the information about the density of vehicles on the particular roads is becoming the essential part. It can be made possible by utilizing the traffic camera and other local cameras. This project classifies the type of vehicle passing through the lane/road and also to count the number of each type of vehicle. It will use object tracking and object classification as its main concept. This project will focus on providing information about which type of vehicle is highly dense on the road.

1. INTRODUCTION

Nowadays, vehicles have become the basic necessity of the people for their daily transport. Due to this, more and more vehicles are running on the roads which increases the density of automobiles on the road. So, keeping track of the number of vehicles that are passing through the lane/roads has become one of the problems. Due to which the detection of the vehicle and its classification became one of the interesting problems of computer vision. As there are many ways to detect and classify the vehicle. One of the effective ways is by implementing the chain of method/processes starting with capturing the traffic camera but in this case, we are going to use recorded video. The first step is to remove the shadow of a vehicle from the image, as the shadow is considered as an important issue because sometimes shadow is considered as the part of the vehicle or sometimes as an individual vehicle which makes it hard to find the precise contour. The process of vehicle detections starts with the implementation of a background subtractor which will remove the part of the image which remains steady every time the frame is captured. After that, applying the threshold method to get the binary distribution of an image, one is the background and the second is a vehicle. As those data still remain with the noise, that is removed by implementing the morphological operations such as closing and erosion. This refined noise-free data will be good to get the precise contour and after implementing the centroid it will be easy to track down the position of the vehicle and detect it. After detecting the vehicle it will be easy to classify its type by getting the area of the contour and then, counting of the vehicle can be done.

2. RELATED WORK

Many works for this problem have been implemented by using Traditional methods as well as Machine Learning Techniques. Most of the traditional methods start with Background Subtraction methods and Thresholding methods, but they vary in the method of implementing the Region of interest(ROI) and noise removal techniques. For example, (R. R. Chandrika et al., 2019) has implemented the erosion method of morphological operations to remove the noise and incorrect failed region and followed by Laplacian operation on image to get the border image of the vehicle. (Anandhalli Mallikarjun et al., 2015) have implemented the K Gaussian distribution in the background subtractor method to improvise the previous versions of the background subtractor method and followed by connected component labeling. (Á. Virginás-Tar et al., 2014) used a Mixture of Gaussians (MOG) technique in Background Subtractor and closing operator in Morphological operation for noise removal from threshold image. In the traditional method, the shadow of vehicles becomes a major issue because many times shadow is considered as an individual vehicle or as a part of the vehicle which detects the wrong contour of the vehicle and results in getting a big box around the vehicle. To overcome this situation (Vargas et al., 2010) have come up with the idea of using reflectance ratio and the edge density to remove the shadow of vehicles. Recently, with the increasing popularity of Convolutional Neural Network(CNN), many experts have implemented CNN in vehicle detection. As in traditional CNN, it's difficult to reduce computational expenses due to fusing the multi-scale feature maps because they are scale-sensitive that's why (Xiaowei Hu et al., 2018) introduce the new technique of CNN which is Scale-insensitive CNN(SINet) for detecting the fast-moving vehicles. Advanced methods have been introduced with time so those methods are also introduced in the Vehicle detection process like instead of CNN, Fast R-CNN is used with the You Look Only Once(YOLO) method for fast vehicle detection as observed by (Huansheng Song et al., 2019). As the traditional methods work on the background subtractor, this process is fast to implement and able to work in real-time but it fails to detect the precise contour of the vehicle which sometimes mis-classifies the vehicles. As mentioned above there are many works in which machine learning techniques are implemented to accurately detect the vehicle. But it makes algorithms heavy in terms of processing time. Thus we are going to use the traditional method and reduce the effect of the shadow which makes it different from other works.

3. METHODOLOGY

The project is divided into three main tasks including Cleaning of Frame, Vehicle Detection and Vehicle Counting and Classification. The project starts with the utilization of the Traffic camera or Recorded video(we have captured some video footage for initial references from the Baltimore Highway I-695 which passes near UMBC). After that two consecutive frames are captured from the video, then implementing the absolute difference method between two consecutive frames to calculate the absolute difference between two consecutive frames. That result is then converted into grayscale for reducing the calculation that might have happened more in RGB format. As Figure 3.(b) shows the grayscale conversion of the previous result. After that Gaussian Blur method is implemented on a grayscale frame. Figure 3.(c) represents the gaussian blur output of grayscale image, as a gaussian blur method is implemented to remove the Gaussian (random) noise from the frame. Then to convert the frame into binary bifurcation of frame(background and vehicle), for that thresholding method is applied. As Figure 3.(d) represent the outcome of the Gaussian Blur frame after applying threshold method.Thresholding is used to unified the gradient color effect available in the previous frame into two extreme parts(black and white) but still, that image will have some noise data for example, as we see many black colored spots inside the white colored section in Figure 3.(e). So, it can be removed by implementing the Morphological operation such as close and erosion. Morphological operations include various sub-operations like erosion, dilation, opening, closing, Morphological Gradient, Top hat, and Black hat which are used to remove the noise. Specifically we are applying the dilation method of morphological operations. In the dilation method, the border is dilated/expanded in Figure 3.(e) shows the output of the dilation method. Finding the contour of the vehicle in the frame is the important part, to map the location as well as defining the shape of that vehicle and to get the precise flow and location of the vehicle. The contour is created by making a box around the detected object and getting its center is the process for finding the centroid of that bounding box and all this process is done on the frame of the video. After that, we get the detected vehicle by getting the contour, and by using its centroid, tracking of the detected vehicle can be done. By the size of the contour, classification of the vehicle is made, and then finally counting of the vehicle can be done by creating a line and finding the number of centroids which passes through that line.

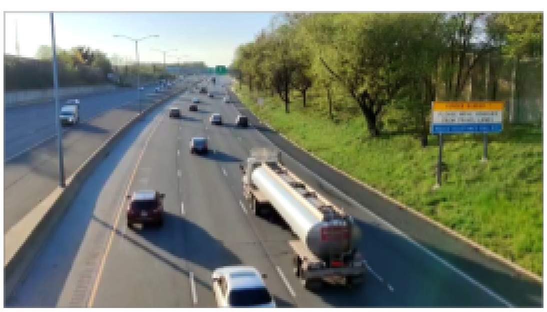

Figure 3.(a)
Original frame that is captured from the video

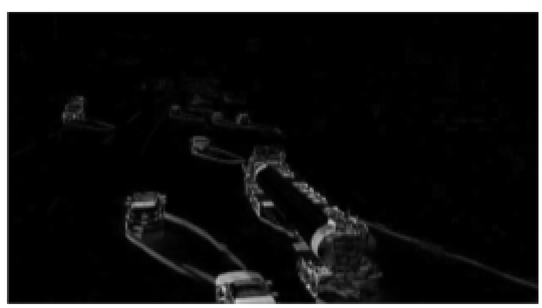

Figure 3.(b)
Greyscale frame obtained after applying the greyscale method on frame

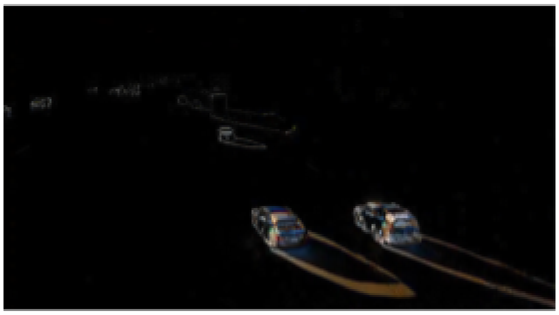

Figure 3.(c)
Gaussian Blur frame

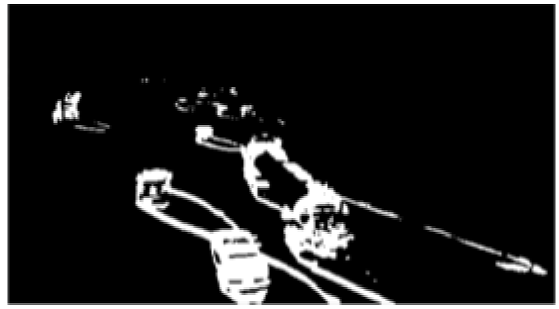

Figure 3.(d)
Threshold output of Gaussian blur frame

Figure 3.(e)
Output obtained after applying the dilation method on threshold frame

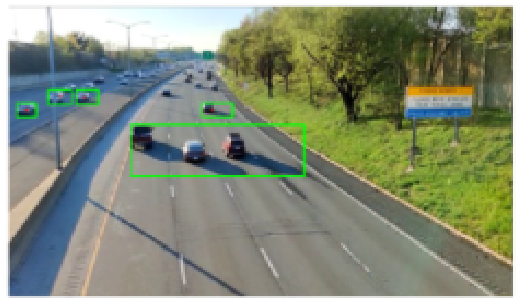

Figure 3.(f)
This is the error with miss calculated the shadow as the part of the vehicles

But this method leaves space for a lot of errors. One such major error is the distorted object detection resulting from the shadows which overlap the vehicles as shown in figure 3.(f). For that, we are going to use the shadow removal technique that was shown by (Vargas et al., 2010) to remove the shadow of the vehicle with some modification. The method suggests using sigma-delta background subtraction to get the background model but we will be going with the one proposed above. The quotient image is typically calculated as the quotient between the current frame I and the background model M or the inverse as you can see in the equation (1) below.

$$C_1(x) = \frac{I(i,j)}{M(i,j)} \quad ; \quad C_2(x) = \frac{M(i,j)}{I(i,j)} \quad \forall\, pixel(i,j) \tag{1}$$

$$C_3(i,j) = \frac{I(i,j) - M(i,j)}{I(i,j) + M(i,j)} \quad \in [-1, 1] \tag{2}$$

Here I is the current frame shown in figure 3.(g) and M is the background Model as shown in figure 3.(h). The quotient image amplifies the intensity of the surfaces where the shadows or reflections are cast. Note here that both C_1 and C_2 are unbounded. Range of C_1 is between 0 and 1 while range of C_2 is between 1 to ∞. This may cause problems in computation. So we take an updated version of calculating the quotient image from (S. Nayar et al., 1996) which is shown in equation (2). We now calculate the quotient gradient from the quotient image using a sobel filter vertically and horizontally as shown in Figure 3.(i). Then morphological operations are used to remove noise from the image. For that adaptive erosion will be used. We get output as shown in figure 3.(j). Then we compute the edge density image. For this we get the edge image by binarization of the gradient image and then perform convolution smoothing operation. But this process will result in several disjoint blobs as shown in figure 3.(k). For merging the blobs, seeded region growing will be performed on the edge density image. But not all of the blobs will be merged so finally we use a blob clustering algorithm to merge these blobs. Blob clustering algorithm is based on the simple concept of minimum distance: whenever the distance between two blobs is below a given threshold, they are considered to belong to the same object and they are labelled as connected blobs. So we now get the image without the shadows with vehicles intact as given in figure 3.(l). Now we are going to test our method on various video clips to find out the outcomes. We test our method for sunny and cloudy weather. Also we test for noon, morning, evening and night conditions. This sums up our qualitative analysis. For quantitative analysis we test our method for heavy and light traffic. We measure our accuracy in two ways. We measure accuracy by comparing the number of vehicles passed and the actual number of vehicles which have passed. But this result has been observed to be false as for some instances a vehicle is not counted and for some instances a single vehicle is counted as double. So we also count the number of errors or misclassified instances for a clip to compute accuracy of our method. The results have been shown in table 4.1.

Figure 3.(g)
Image before extracting the background

Figure 3.(h)
Background Image

Figure 3.(i)
Quotient gradient image

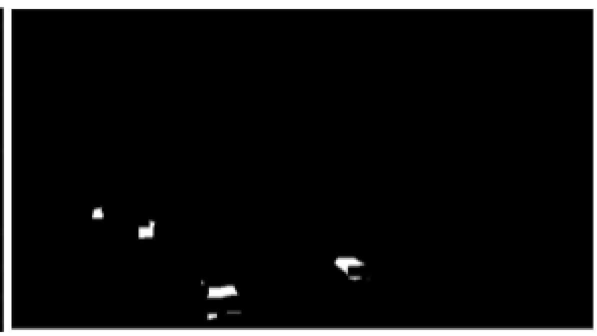

Figure 3.(j)
After applying erosion morphological operation

Figure 3.(k)
Convolutional Smoothing Operation

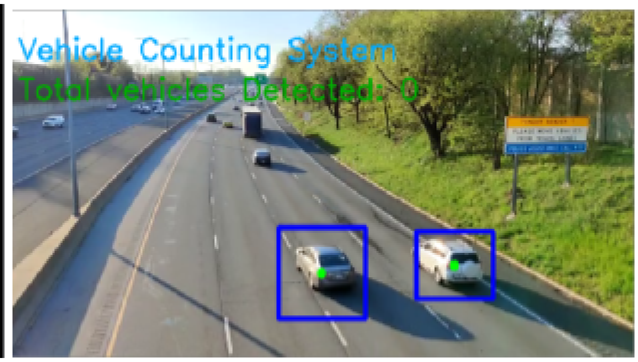

Figure 3.(l)
Output after applying shadow removal technique

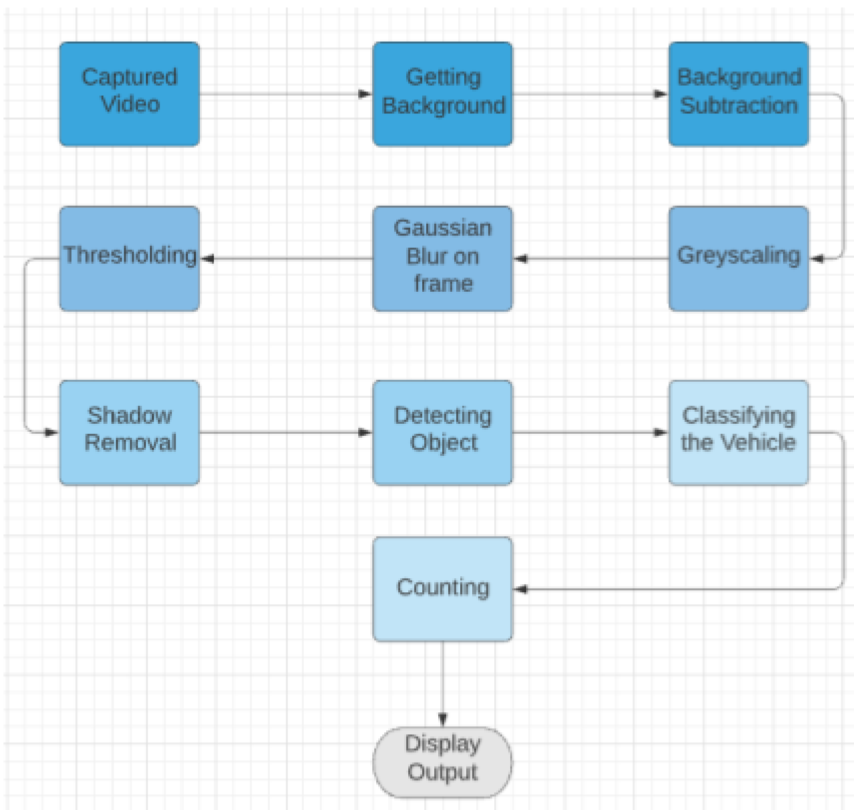

Figure 3.(m)
Workflow Diagram

Figure 3.(m) represents the workflow diagram of the new shadow removal methodology.

4. RESULTS

We have implemented our working model on different recorded videos which we have captured from Highway I-695 that passes nearby UMBC and also included a few video datasets from the internet(Gustavo Velasco-Hernández et al.). First we test our method on various day conditions such as day, night, evening as well as cloudy and sunny weather without taking shadows into account. We took different traffic considerations into account such as low, medium and high traffic. Table 4.1 displays the results we obtained. We can see that the highest accuracy was observed during the cloudy day. This was the ideal condition. It had no interruption of shadows and the traffic was mild. We observed the highest accuracy of 92%. But the average was around 89% for cloudy weather. This is observed due to varying traffic conditions. Next we tested on a bright sunny day where we got lower accuracy. Shadows distort the foreground objects and break the systems. Highest was observed around 73%. And during the Night time accuracy lowered even more. It was observed to be around 64%. But the night time accuracy count could be significantly improved by placing the camera on the reverse side as shown in figure 4.(a) and figure 4.(b). There was drastic improvement observed which could be noted from table 4.1. Now we test the effectiveness of our method after removing the shadow and see the improvements. In cloudy conditions no significant changes were observed. But for sunny and night conditions there were few changes observed which is mentioned in table 4.1.

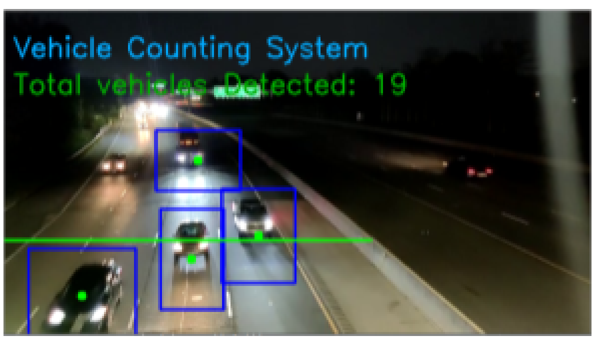

Figure 4.(a)
Camera placed at the front which results in low accuracy during the night.

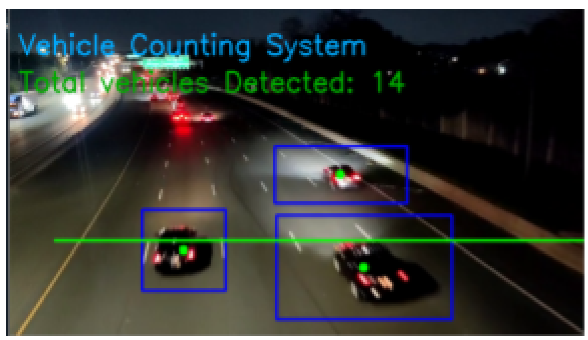

Figure 4.(b)
Camera placed at the behind to improve the accuracy during the nighttime

Table 4.1 Evaluation metrics of our method(Best results for each case)

Condition	**Vehicle Detected Count**	**Original Count**	**Error count (approx)**	**Accuracy**
Cloudy Day	87	90	7	92.22 %

Sunny	152	160	42	73.75 %
After Implementing Shadow Removal (Sunny)	54	57	9	84.21%
Night (front side)	114	106	38	64.15 %
Night (Back side)	116	120	16	86.67 %
After Implementing Shadow Removal (Night)	92	98	10	89.79%

One thing which is worth noting is that the detected vehicle boundary which originally covered both the vehicle and the shadow is significantly reduced. As we can see from figure 4.(c) the detected vehicle before shadow removal takes too much area. Even though boxes are separate and it counts them differently it takes shadow into consideration and will create problems if vehicles are near enough. Figure 4.(e) shows this case where vehicles are near so the box covers all the vehicles as one object. Now after removing shadow the detected object is precise and does not cover shadow. It can be seen from figure 4.(d) and 4.(f) that each vehicle is detected separately and even though shadow is present it is not counted as the part of the object.

Figure 4.(c)
Vehicle detection before applying Shadow removal technique

Figure 4.(d)
Vehicle detection after applying shadow removal technique

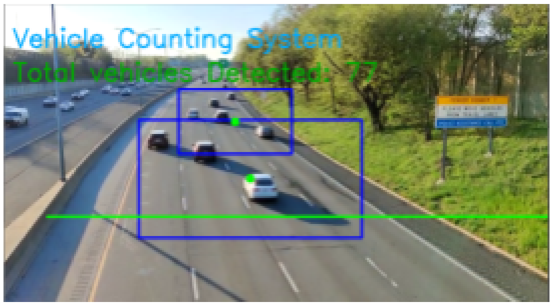

Figure 4.(e)
Vehicle detection before applying Shadow removal technique

Figure 4.(f)
Vehicle detection after applying shadow removal technique

Figure 4.(g)
Break down of image into several blobs

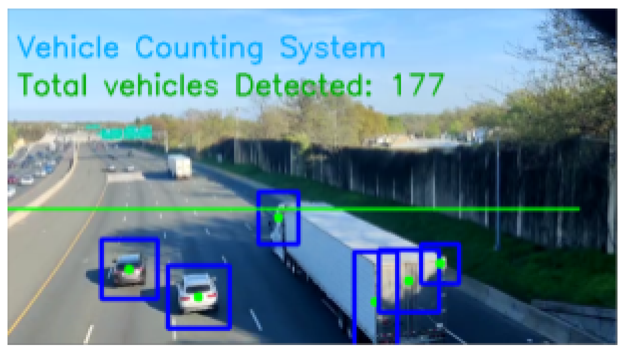

Figure 4.(h)
Multiple object detected when large vehicle passes by

Now the region which we want is only surrounding the vehicle and is not taking shadow into consideration. So now we actually get different vehicles instead of combined vehicles. This should lead to an increased accuracy but it is not the case. After the erosion the image breaks into several blobs as shown in figure 4.(g). So for this seeded region growing algorithm followed by blob clustering is used in the methodology to join the disjoint blobs. But due to not being able to implement these steps our system fails. Dilation is used to minimize the error but it works only for small vehicles. Larger vehicles are still disjointed into several blobs as shown in figure 4.(h). There are two boxes surrounding the vehicle where it should have been one. Sometimes more than two objects are detected instead of one due to this. So we remove the shadow but a new problem comes up. And the results before and after implementing shadow removal can be seen in table 4.1.

5. CONCLUSION

Shadow removal in the real world conditions is a major problem, and that too by using traditional techniques makes it even more challenging. There are many papers which rely on color properties to remove the shadow but these techniques make the algorithm heavy. So to make the method light weight and be useful in real time scenarios we go with the grayscale approach. Our approach is based on getting the background image and by processing it with every frame to get a threshold image. Then the shadows are dealt with using adaptive erosion. But here we break the objects in multiple parts which is now a problem for us. We test the method in various conditions as described in the result section above. A success rate over 78 % is obtained in all tested conditions and this is also when we have yet to implement our method completely. Room for improvement and future work is discussed below.

6. DISCUSSION

Even though we have dealt with the shadows which was our main obstacle, we came across a new one which is unresolved for now. When we perform the erosion on the video frame the object which we want to detect and classify breaks into multiple parts which we then try to join by dilation. But the large objects break in multiple pieces and are also separated by a large margin. We believe a seeded region growing algorithm followed by blob clustering as discussed in methodology helps to resolve this by merging the nearby blobs, but we were not able to complete it. So whenever a large vehicle passes the system will count it as multiple objects. One more thing to note is our system is not very robust. It relies on camera positioning too much. We have to manually set parameters for different videos due to change in the camera positioning. And this problem is for almost all the traditional methods which are used for vehicle detection and counting. Here machine learning techniques definitely take the upper hand. So further research would be required to develop the traditional method which is robust and can be used without too much manual interference.

REFERENCES

Anandhalli, Mallikarjun & Baligar, Vishwanath. (2015). Improvised approach using background subtraction for vehicle detection. Souvenir of the 2015 IEEE International Advance Computing Conference, IACC 2015. 303-308. 10.1109/IADCC.2015.7154719.

A. Prati, I. Mikic, M. Trivedi, R. Cucchiara, “Detecting moving shadows: algorithm and evaluation”, IEEE Trans. Pattern Anal. Mach., Vol. 25, Iss. 3, pp. 918–923, 2003.

Á. Virginás-Tar, M. Baba, V. Gui, D. Pescaru and I. Jian, "Vehicle counting and classification for traffic surveillance using Wireless Video Sensor Networks," 2014 22nd Telecommunications Forum Telfor (TELFOR), Belgrade, Serbia, 2014, pp. 1019-1022, doi: 10.1109/TELFOR.2014.7034580.

E. Salvador, A. Cavallaro, and T. Ebrahimi, “Cast shadow segmentation using invariant color features”, Computer Vision and Image Understanding, Vol. 95, pp. 238–259, 2004

Gustavo Velasco-Hernández, traffic-surveillance-dataset, “https://github.com/gustavovelascoh/traffic-surveillance-dataset”

Huansheng Song, Haoxiang Liang, Huaiyu Li, H. *et al.* Vision-based vehicle detection and counting system using deep learning in highway scenes. *Eur. Transp. Res. Rev.* 11, 51 (2019).

J. C. Silveira Jacques Jr., C. Rosito Jung, and S. Raupp Musse, “Background Subtraction and Shadow Detection in Grayscale Video Sequences”, 18th Brazilian Symposium on Computer Graphics and Image Processing, SIBGRAPI 2005, 2005, pp. 189-196

J. Yao, Z. M. Zhang, “Hierarchical shadow detection for color aerial images”. Comp. Vision & Im. Underst., Vol. 102, pp. 60–69, 2006

R. Cucchiara, C. Grana, M. Piccardi, A. Prati, “Detecting objects, shadows and ghosts in video streams by exploiting color and motion information”, Proc. 11th International Conference on Image Analysis and processing, 26-28, 2001, pp. 360-365.

R. R. Chandrika, N. S. G. Ganesh, A. Mummoorthy and K. M. K. Raghunath, "Vehicle Detection and Classification using Image processing," 2019 International Conference on Emerging Trends in Science and Engineering (ICESE), Hyderabad, India, 2019, pp. 1-6, doi: 10.1109/ICESE46178.2019.9194678.

S. Nadimi, B. Bhanu, “Moving shadow detection using a physics based approach”, Proc. IEEE Int. Conf. Pattern Recognition, Vol. 2, 2002, pp. 701–704.

S. Nayar and R. Bolle, “Reflectance based object recognition”, International Journal of Computer Vision, Vol. 17, no. 3, pp. 219– 240, 1996

Toral, S.L. & Vargas, Manuel & Barrero, Federico & Ortega, M.G.. (2009). Improved sigma-delta background estimation for vehicle detection. Electronics Letters. 45. 32 - 34. 10.1049/el:20092212.

Vargas, Manuel & Toral, S.L. & Milla, J.M. & Barrero, Federico. (2010). A shadow removal algorithm for vehicle detection based on reflectance ratio and edge density. IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC. 1123 - 1128. 10.1109/ITSC.2010.5625112.

Xiaowei Hu, Xuemiao Xu, Yongjie Xiao, Hao Chen, Shengfeng He, Jing Qin, Pheng-Ann Heng, 2018, ”SINet: A Scale-insensitive Convolutional Neural Network for Fast Vehicle Detection,” arXiv:1804.00433v2 [cs.CV] 16 May 2018