Turkish Online Journal of Qualitative Inquiry (TOJQI)
Volume 12, Issue 6, July, 2021: 9892-9901

Research Article

Detection of Moving Vehicle in Foggy Environment using Google's Firebase Platform

Sneha Mishra[a], Dileep Kumar Yadav[b], Farah Tabassum[c], Divyansh Kumar[d]

[a,b,c,d] School of Computing Science and Engineering, Galgotias University, Greater Noida
Email: [a]Indiasnehariet@gmail.com, [b]dileep252000@gmail.com, [c]farah.tabassum1999@gmail.com, [d]adivi2214@gmail.comd

Abstract

Now a day, the vehicle detection and tracking in intelligent transportation system is a highly active area. It focuses on real-time situations such as traffic monitoring, suspicious vehicle surveillance, and analysis of traffic flow, avoiding congestions in traffic, avoiding pile-ups and collisions or accidents, especially in a foggy environment. The proposed work also resolves the above problems and delivers solutions for the enhancement of the transportation system and the automobile industry. So, this paperwork investigates a method using Google's firebase platform (as a cloud service) and a background subtraction method for moving vehicle detection and tracking in a foggy environment. Here, the Google's Firebox storage (Pyrebase API) is used for the storage of video that provides authentication and storage services. The moving vehicle is detected using the background subtraction method for considered video followed by post-processing and achieves better results.

Keywords: *Vehicle detection, Google's Firebase, Foggy Environment, Video Surveillance*

Introduction

In today's real world, technology is enhancing day-to-day for computer vision applications in the field of video surveillance for vehicle detection and tracking. For avoiding real-time adverse situations such as traffic congestion, pile-ups, accidents due to ill management of transportation system, the detection and tracking based methods must be improved. For video surveillance in intelligent transportation systems, traffic cameras are installed in chosen positions. The captured information is passed to the control centre of the traffic management system for further analysis and investigations, for such a date Pan Tilt Zoom, Vector cameras, *etc*. may be used for traffic surveillance or detection of motion-oriented vehicles on-road or highway. Apart from fog, in reality, there may be other challenging problems such as – illumination variation, snow, dust, mist, cloudy environment, drift or occlusion, etc. This paperwork mainly focuses on the foggy environment. The main purpose of analysing computer vision is to detect, track and estimate the features of the object directly with the help of computers and then develop a system that reduces human effort. With the enhancement of technology, society is adopting digital video-based surveillance [1, 2, 3, 4]. The detecting and tracking of targets is also a very important application area. Mobile target detection technology distinguishes moving objects and backgrounds and extracts moving targets from video. Such work is used for transportation, industry, and security surveillance [5, 6, 7, 8].
The detection of moving objects is constantly very difficult in a foggy environment. The object detection-based research has offered a variety of traditional methods, including frame differentiation, optical flow, and background extraction to detect moving objects, and are recently got a lot of attention. According to the literature, researchers have done few works for detection and tracking under foggy environments, and remains a very tough task to identify, detect and track the moving vehicle on highway or road. Hence, the proposed work develops a method for such work. The focus is to experiment with such work under Google's firebase cloud platform. It reduces the number of spirits in removing the critical gaps and background in differences between the frames and effectively detect moving objects in the presence of critical background. However, in complicated dynamic situations, the detection is not accurate, as many false positives are classified in most cases. Various pixels may be misclassified due to illumination variations, environmental or weather effects of the background scene.
This article suggests a new direction to the detection of moving vehicles. In this paper, the background subtraction method is applied over videos stored in Firebase Storage, and the static backgrounds show the following new attributes: 1) In the proposed method, frames are considered as a sequential order 2) bilateral blurring removes noise as a pre-processing 3) morphological techniques as post-processing are included in the process; 4) Contour detection, that allows to avail more accurate information.
The main motivation behind this work is problems associated with adverse weather while driving. While the entire car body is fogged up, the windshield and windows are most visible and may create dangerous driving situations. In the thick fog, the driver is unable to see beyond the limits of his vehicle. Driving in such conditions may be very problematic and cause accidents or pile-ups, which affects perceived judgments about speed and distance. The effect is a result of reduced contrast. The rain reduces driver awareness and changes visibility through changes to the headlights, windshield, road itself, and road signs. It also affects traffic flow by reducing road capacity, forcing drivers to reduce their driving speed, slowing down travel time, and increasing the overall risk of accidents [16, 17, 18, 19, 20, 21]. Hence, the proposed method works optimally and detects objects on the screen even in inclement or adverse weather, especially in fog. The moving vehicle detection
This paper is organized and explored in the following sections. Section 2 focuses on the literature work. The remaining part of this manuscript discusses the main techniques and methodologies of the proposed work in Sections 3, 4. The experimental results are presented in Section 5 with the proposed task boundaries. Finally, conclusions and future guidance are discussed in Section 6.

Related Work

There are great approaches in the literature for detecting and tracking moving objects in a video stream. Motion-based information plays a very important role in the recognition process. Various approaches and algorithms have been developed on object detection in the literature. Some of them are discussed in this paper.
Li *et. al.* [4] have proposed a method for the moving object in the video that has been detected by the cloud server. It also shows the computing ability of the suggested cloud platform and enables the utilization of the available resources. It also avoids having to rely on mobile devices for the limited processing capacity.
Tuli *et. al.* [5] suggested a framework to deploy deep learning-based applications to deliver better service quality in fog-cloud environments to harness edge and cloud resources. The authors also developed a framework, (*i.e.* EdgeLens) that adapts to user requirements and depicts high accuracy through experimental analysis or low latency modes of services.
Gruyer *et. al.* [6] focuses on real-time natural issues such as fog, snow, haze, rain, or sun glare and these are very dangerous for drivers. For visibility problems, the driver faces problems while driving. This paper develops a method to improve visibility in adverse weather conditions.

Singh *et. al.* [7] has investigated a method for detection of moving vehicle in foggy environment. This work also applied a LiDAR to measure the distance from the front vehicle and raise a warning message according to the measured distance.
Yadav *et. al.* [9] has developed a Kullack-Leibler Divergence-based method for moving object detection in a thermal environment and demonstrated better performance against considered peer methods.
Lochabh*et. al.* [10] has given a theoretical application-oriented concept for object detection in the video in a Cloud Environment. This paperwork also focuses on the real-time-based problematic challenges.
S. Yadav et. al. [11] also developed a method for enhancing services in the transportation system. This work focused to automate the Indian transportation system through intelligent searching and retrieving mechanisms in amazon's elastic compute service.
Zhang et.al. [29] has developed a Moving- confidence- assisted Matrix decomposition (MCMD) model that has focused on foreground regularization and configured to assist real-time moving objects with the moving confidence score evaluated from optical flow methods. This work developed batch processing and online solutions over videos by SkySat and Jilin-1.
Chaojun et. al. [30] has proposed a vehicle model detection to improve the accuracy of recognition and vehicle model's speed employing an improved YOLOv3 model. The dataset – BIT- vehicle resulted in the depiction of improved recognition speed.
Xiaosong [31] improved the method of tracking to detect the increased number of counts for speedy vehicles. This work was based on pairing results over distinct atmospheric conditions like dusty and foggy weather. Similarly, Yang *et. al.* [33] and Ao *et. al.* [34] detects moving vehicles from satellite videos that were captured by moving satellite platforms.

Proposed Method for Moving Vehicle Detection

In this section, the proposed work explored the work carried out for the development of the investigated working model; this work also focuses on the proposed background subtraction (BGS) method that develops a background independently of each pixel relative to the previous pixel value. Here, the bilateral blur filter is applied for improving the model. To ensure uniformity, frames are drawn and converted to grayscale levels. Frame differencing stores different coordinates in grayscale frames to detect irregularities or intensity variations in two frames. The bilateral blur is used to smudge the frame for saturation and dilation that makes it easier to find contour areas. All of this acts as normalization for more accurate motion results. This work developed a frame difference scheme using a background subtraction technique for the detection of moving vehicles in a foggy environment. Here, input is provided through the cloud environment, and output is delivered to the cloud environment. This work is presented in five distinct sections:

- To compute differences between two adjacent frames.
- To change the moving object to grayscale and construct the background.
- Apply blurring and smoothening of the edge then remove noise.
- Use BGS technique and classify the pixels a suitable threshold.
- To apply the morphological operators to improve the quality of evidence.

The outcome depicts the better detection quality of the moving vehicles in a foggy environment. The basic working model of the suggested work is shown in *Figure-1* and Similarly, *Figure-2* depicts its working with Firebase Google's cloud platform.

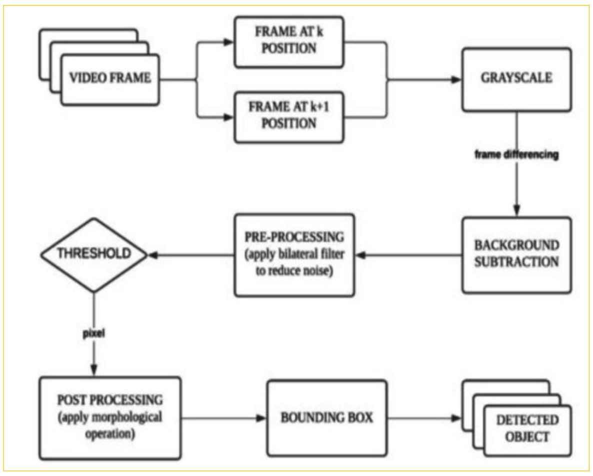


Figure 1. The basic workflow of the proposed method.

Steps and Procedure

The main procedure is performed using following steps:
A. Open the application: Here, the suggested task is initiated.
B. Cloud Data Connectivity: This step is experimented by providing the connectivity through Google's Firebase platform for the cloud environment. The forthcoming sections also reveal a brief description of the platform, tools, services, and methods used to accomplish this work.

1. Firebase

The Firebase [11, 17, 18] platform provided by the Google, accelerates application development through its inbuilt services. It offers BaaS or backend as a service, so Firebase handles cloud infrastructure and all backend requirements. The main benefits of the firebase platform are to allow users to grow and deploy their applications faster. The Firebase is also enables hosting and has APIs for machine learning tasks such as predictive text, image captions, *etc*. The working system design using the firebase platform is given in fig.2.

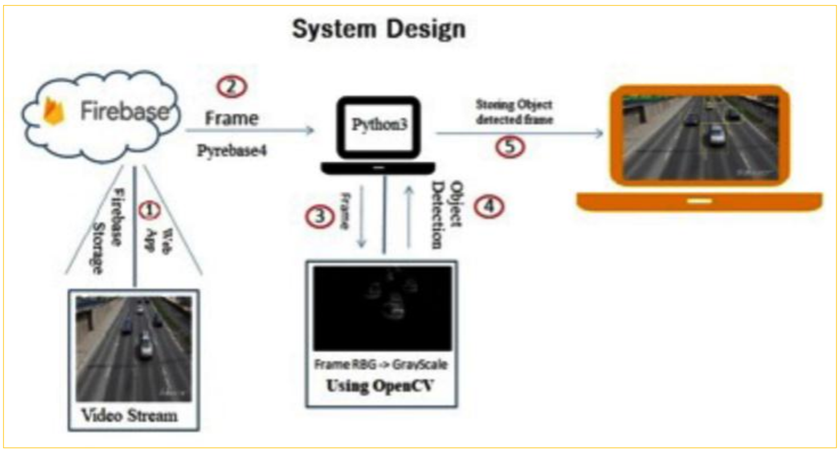


Figure 2. System design to demonstrate the connection of Firebase to PyCharm

2. Firebase console

To work with Google's Firebase, the user/ programmer must have a Google account then open the Firebase console, register, and provide the app with Firebase. It has the following project environment according to the requirements: (i) iOS for Apple, (ii) Android, and (iii) Network. In the Firebase console click on the web setup link present on the Auth tab to get the authDomain, databaseUrl, apiKey, and storageBucket variables needed to link it with the database.

```
{
"apiKey": "AIzaSyBqDOtkV9ue802-9KonKcdvJRyuL1xzbhI",
"authDomain":"object-detection-project-ce0b4.firebaseapp.com",
"projectId":"object-detection-project-ce0b4",
"databaseURL":"gs://object-detection-project-ce0b4.appspot.com",
"storageBucket":"object-detection-project-ce0b4.appspot.com",
"messagingSenderId":"1050244789749",
"appId":"1:1050244789749:web:1b516c54ec2ef3a8025cbc",
"measurementId":"G-QCL2WVLCQC"
}
```

3. Firebase Storage - Pyrebase

After adding a new project in the Firebase console, it is necessary to import Pyrebase services to the application for using Firebase. The interface of Firebase REST API is Pyrebase i.e. one can use Python to manipulate your Firebase database. The pip is required to install Pyrebase and its dependencies through a package manager to install and manage inbuilt packages of Python, which are acquired from a third-party depository. The Pyrebase apps can use many Firebase services:

1. firebase.auth() for Authentication purpose
2. firebase.database() for Database services and management
3. firebase.storage() for Storage management

The architectural diagram of firebase (Google's) platform for cloud storage services is depicted in fig. 3.

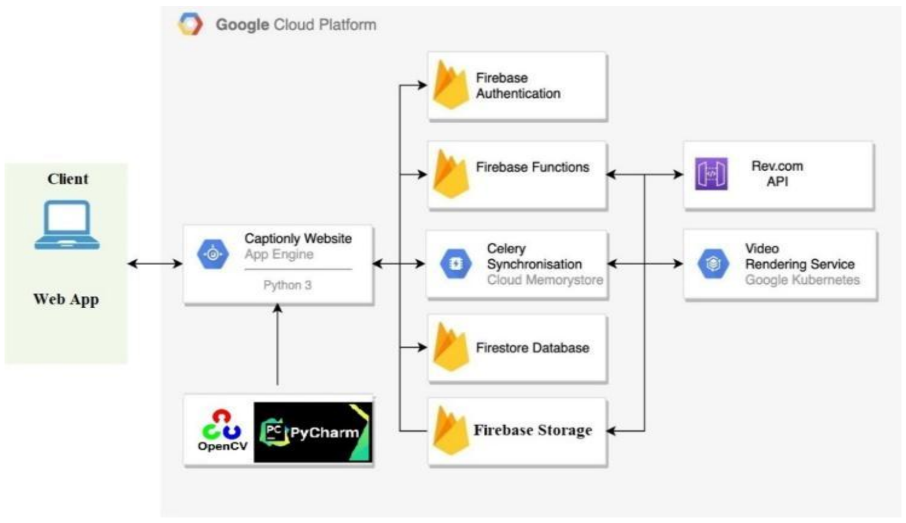


Figure 3. Architectural Diagram of Cloud (Firebase)
[Source: https://blog.smartandnimble.com/content/images/2019/11/captionly_architecture.jpg]

4. Firebase Storage - Read video through frames

The cloud storage provides additional protection for Google's Firebase when uploading and downloading files from the Firebase app, irrespective of network quality. Here, SDK is used to accumulate video, images, audios, or other content generated by the user, the Google cloud storage is used to store and retrieve the same data on the server.

With the help of the root *or* child method, the paths may be built to data with the storage service and the download method uses the path to the saved database file.

C. OpenCV

Here, an OpenCV (an open-source library for computer vision, machine learning, and image processing) plays a vital role in real-time environment in this system. It can be used to refine videos and images to detect object, features, *or* even the handwriting of a person. To perform object detection, the following modules must be imported.

1. import cv2 - python library for solving computer vision problems.
2. import imutils - a set of functions to facilitate basic image processing operations and viewing Matplotlib based outcomes.
3. import time - useful for knowing and analysing each object found in the video.

D. Grayscale

Here, the grayscale format is converted from other RGB spaces. It represents shades of black and white from 0 to 255 that contain 8-bit/pixel (bpp) data. This reduces the size and complexity of the model and is important because other algorithms are adapted to work only with grayscale images.

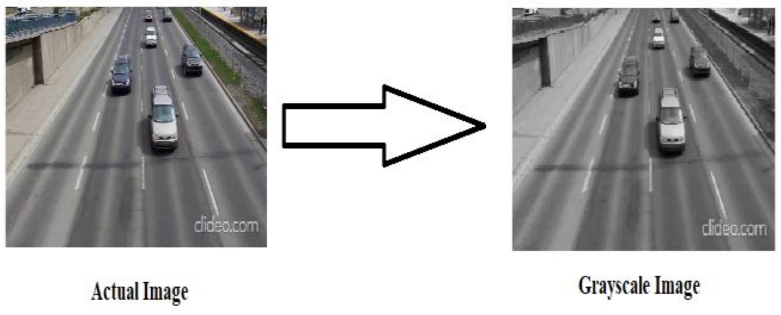


Figure 4(i). Result of the execution of grayscale

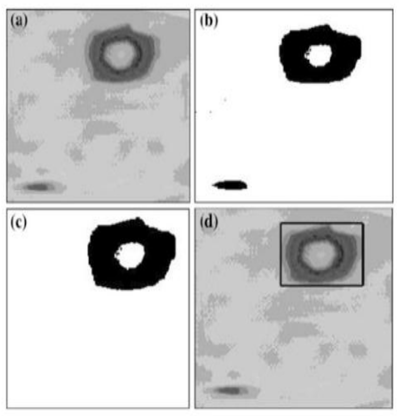


Figure 4(ii). Detection result. (a) Grayscale image; (b) Grayscale threshold image; (c) Grayscale threshold image after clutter removal; (d) Grayscale image with object outlined [2]

E. Background Subtraction

The background subtraction method allows detecting the moving objects *or* vehicles from video captured by a fixed/ static camera. The object detection in this approach would be based upon the identification of changes between sequences of frames. The background model works as a reference here. The reference and identified changes on the new frame are used to detect the movement of objects. At the time of object detection, the state is static *or* moving. This method compares pixel differences between the adjacent frames to identify the state (static or moving) of the object.

In this section, the cv2.absdiff() function is applied to determine the absolute difference between the pixels of two frames. It is used to extract pixels from moving objects. To use cv2.absdiff, first convert the video frame to grayscale. The frame differencing function is applied to compute frame difference mentioned in the given Eq. (1) and detects objects between two frames.

$$\boldsymbol{Fd_{(k,k+1)} = F_{k+1} - F_k} \qquad (1)$$

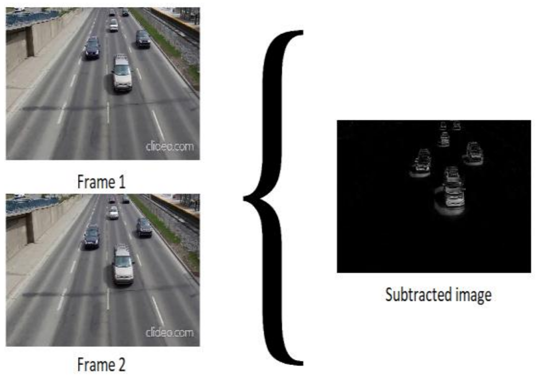


Figure 5. Result of frame differentiation dependent on picture successions

F. Blurring – Bilateral Filter

For accomplishing the blur of frame, various low-pass filters are used. A bilateral filter [31] is a non-linear image-smoothing filter that preserves edges and reduces noise. It replaces each pixel's intensity with a weighted average of neighbour pixel's intensity value. Similar to the Gaussian filter, the bilateral filter also considers adjacent pixels with a set weight. This weight is made up of two sections, the first of which is the same as the one utilized by the Gaussian filter. The second factor considers the intensity difference between the neighbours and the estimated pixels [28].

$$\boldsymbol{BF = \frac{1}{W_p}(||p-q||)\, G\sigma_r(|-I_q|)I_q BF = \frac{1}{W_p}(||p-q||)\, G\sigma_r(|-I_q|)I_q} \qquad (2)$$

cv.bilateralFilter() is very effective at removing noise while keeping the edges sharp. However, it works slower than other filters. The cv.bilateralFilter() function accepts the following parameters.

- The first parameter is the source image.
- The second parameter is the diameter of each quarter of the pixels used during filtering. If it is not positive, sigmaSpace will calculate it.

- The third parameter is sigmaColor, which is sigma, filters in the color space. A higher parameter value means that the color further next to the pixel are blended, resulting in a larger area with semi-flat colors.
- The fourth parameter is sigmaSpace, which is filter sigma in coordinate space. Higher parameter values mean farther away pixels are affected as long as they are close enough. If d > 0, the size of the environment is determined independently of the sigmaSpace. Otherwise, d is proportional to sigmaSpace.

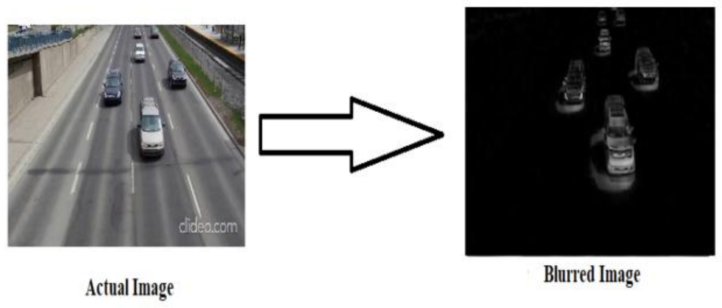


Figure 6. Result of the execution of bilateral blurring dependent on picture successions

G. Thresholding

The threshold (T) is a more adopted segmentation technique used to separate an object from a particular frame. The threshold [9, 15, 16] is determined by comparing the value of each pixel in the image (pixel intensity) to a specific value of the threshold. This separates the input image's pixels into two groups:

1. Pixels with a value of intensity less than the threshold.
2. Pixels with a higher intensity more than the threshold value.

Now, these two groups receive different scores depending on the type of segmentation. The pixel value is set to 0, if it is less than the threshold [15, 23, 24]; else it is set to the utmost value. The cv.threshold() function is used to set the threshold value.

- First parameter is an input frame on which Gaussian Blur operation is applied.
- Second parameter is classifying pixel values by threshold used.
- Third parameter is the utmost value, which is assigned to a value of pixel above the threshold.
- Fourth parameter is various types of thresholds indicated, offered through OpenCV.

The basic traction described above is performed with type cv.THRESH_BINARY. The classification function can be computed by using the given equation (3), this function classifies the pixel during runtime and uses threshold (T) for classification for computing the maximum value of the pixel.

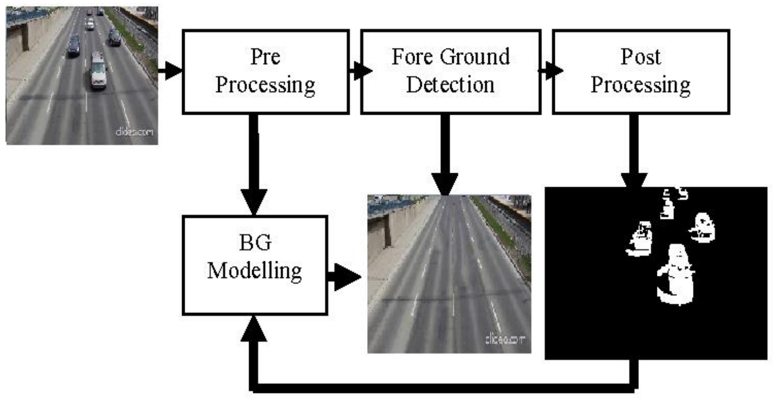


Figure 7. Working model of the proposed method with the outcome

H. Dilation:

The dilation process expands 1 pixels to the boundaries of an object in a frame. It is performed on binary images. The main effect of dilating a binary frame is to enlarge the boundary of the foreground area [22, 23, 24]. So, the pixel area in the foreground expands if the holes in this area get smaller. The dilate() function accepts the following parameters.

- The first parameter is a source frame with several channels (all processed independently of each other)
- The second parameter is the kernel element, the origin of this is determined by the anchor (standard (-1, -1) *i.e.* the middle of the structuring element.

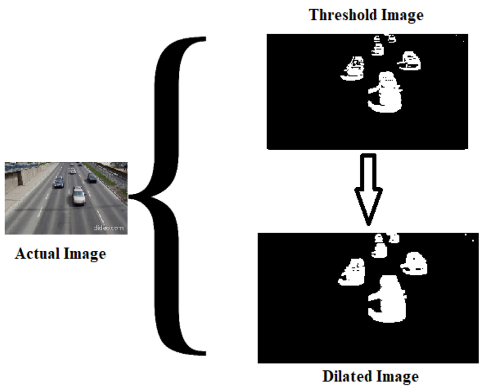


Figure 8. Result of the execution of dilation dependent on picture successions

- The third parameter iteration shows how often the process is repeated. It is done using the "bordertype" and "borderValue" arguments.

$$cfn_{(x,y)} = \begin{matrix} \boldsymbol{maxVal} & \boldsymbol{if\ src_{(x,y)} > T} \\ \boldsymbol{0} & \boldsymbol{otherwise} \end{matrix} \qquad (3)$$

I. Bounding Box

A bounding box [25, 26, 27] is a rectangle *or* square that serves as a starting point for finding a vehicle in frame and creating a box for that vehicle in frame. To draw the rectangle in the frame. It makes easier for machine learning algorithms to find what they are looking for, identify the path, and saves valuable computational resources. A contour is a closed curve that joins all continuous points of a certain color *or* intensity. It represents the shape of the object found in the frame.

To correctly identify the bounding box, it need to convert the frame to a monochromatic color format (such as grayscale) and then apply a binary threshold. The object's edges are completely white with the same color intensity. It detects the boundaries of vehicle with white pixels (255) and the background pixel value (0).

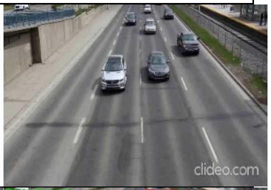
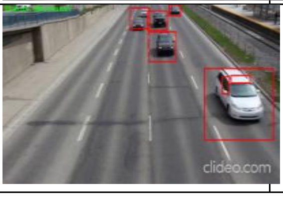
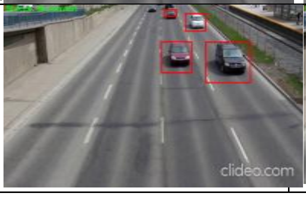
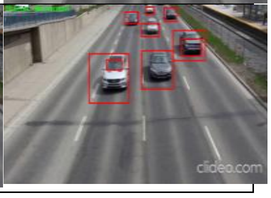
The cv.findContours() function that stores coordinates (x, y) at the boundary of a particular form, and takes three arguments:

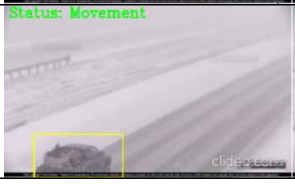
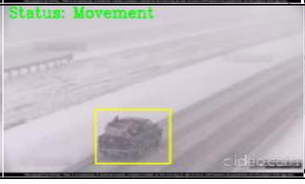
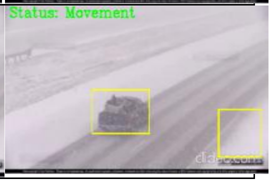
1. Original frame,
2. Contour extraction mode.
3. Contour approach method.

J. Detection of Moving Vehicle: The algorithm detects moving vehicle from the video frames.

Experimental Setup and Result Analysis

To test the efficiency of the proposed method, the analysis deals with three video datasets in this paperwork. The input videos are randomly captured. The results are shown in *Figure 9*, where (a) the time is normal; (b) foggy weather; (c) binary output. It selects three frames from each video as shown in the results. In Figure-: (a) and (b), in each *Figure*, the first row depicts video frames and the second row focuses on the tracking of moving vehicles using the proposed method. Figure-9 (c) computes the detected results of moving vehicles on a highway. In this figure, the first row depicts the original frame, the second row consists of the ground truth frame and the third row clearly shows the extracted information of the moving vehicle in binary form. The binary results demonstrate better qualitative observations and show that the proposed method is suitable for moving vehicle detection under Google's Firebase cloud environment.

	Frame-1	Frame-2	Frame-3
Original Frame			
Detected Frame			

	Frame-1	Frame-2	Frame-3
Original Frame			
Detected Frame			

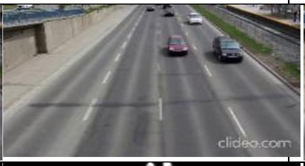
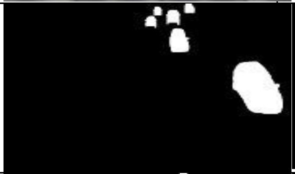
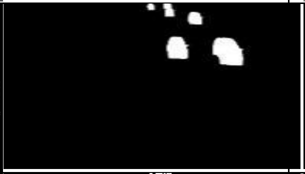
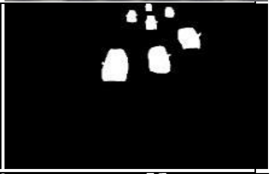
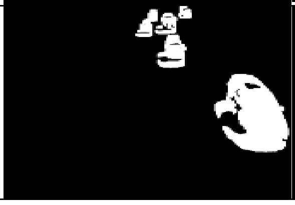
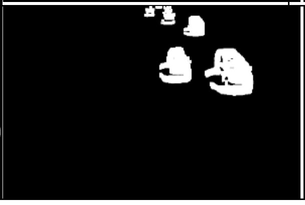
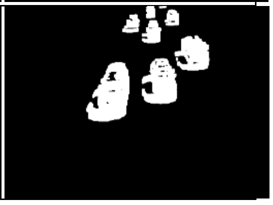
	Frame-1	Frame-2	Frame-3
Original Image			
Ground Result			
Detected Result			

Figure 9. Detection result. (a) Normal weather (b) Snow/foggy Weather; (c) Binary output

This work also solves the problem related to adverse weather conditions and detect moving vehicle on road against challenging issues [20, 21] in a real-time environment.

Conclusion

In the real-time environment, during foggy weather conditions, it is very difficult to drive safely. In adverse weather conditions, the accidental *or* pile-up cases happened on the highway such as in foggy *or* snow environments due to poor vision *or* blurred vision. Therefore, this paper presents a background subtraction-based method for moving vehicle detection in a foggy environment. All this work has been carried out in Google's Firebase platform that provides a new direction for video surveillance applications in the area of computer vision. It also demonstrates better detection results and attracts the attention of researchers in this direction too. In this work, the Google's Firebox storage (Pyrebase API) delivers the solution for the storage of video for the authentication and storage services. However, this work has experimented using OpenCV and demonstrated better results in less execution time.

In future, this work may be used for vehicles on autopilot mode facing an obstacle.

References

[1] S. Rosaline, J. J. J. Sheela, Aswini M, A. T. Anasuya, M. Velmururgan, "Fog Density Detection Based Automotive Vehicle", European Journal of Molecular & Clinical Medicine, vol. 7, issue 3, pp. 1633-1639, 2020.

[2] H. Kasban, O. Zahran, M. El-Kordy, et al.,"False Alarm Rate Reduction in the Interpretation of Acoustic to Seismic Landmine Data using Mathematical Morphology and the Wavelet Transform", Sensor Imaging, vol. 11, pp. 113–130,2010.

[3] R. Li, X. Xie, P. Wang,C. Jin, "Cloud-based moving object detection for mobile devices," 2017 20th Conference on Innovations in Clouds, Internet and Networks, pp. 100-102, 2017.

[4] R. J. Guo, N. Liu, S. Li, F. Liu, B. Chen, J. Cheng, X. Duan, C. L Ma, "Pixel-Wise Classification Method for High Resolution Remote Sensing Imagery Using Deep Neural Networks",ISPRS, Int. Journal of Geo-Information, vol. 7,issue 3, no. 110, pp. 1-23, 2018.

[5] S. Tuli, N. Basumatary and R. Buyya, "EdgeLens: Deep Learning based Object Detection in Integrated IoT, Fog and Cloud Computing Environments", 4th Int. Conf. on Information Systems and Computer Networks, pp. 496-502, 2019.

[6] R. C. Miclea, V. I. Ungureanu, F. Sandru, I. Silea, "Visibility Enhancement and Fog Detection: Solutions Presented in Recent Scientific Papers with Potential for Application to Mobile Systems", Sensors, vol. 21, issue. 3370, 2021.

[7] R. Singh, S. Singh, N. Kaur, "A Review: Techniques of Vehicle Detection in Fog", Indian Journal of Science and Technology, Vol 9, issue 45, pp 1-4, 2016.

[8] S. Yadav, D. K. Yadav, A. K. Budati, M. Kumar, A. Suri "Automating the Indian transportation system through intelligent searching and retrieving with Amazon Elastic Compute", Special Issue: Artificial Intelligence based Network Security and Computing Technologies in Wireless Networks, IET Networks, pp. 1-14, Aug-2020.

[9] D. K. Yadav, K. Singh, "A Combined Approach of Kullback-Leibler Divergence Method and Background Subtraction for Moving Object Detection in Thermal Video", Infrared Physics and Technology, Elsevier, vol. 76, pp. 21-31, Feb-2016.

[10] K. Lochab, D. K. Yadav, M. Singh, A. Sharma, "Internet of Things in Cloud Environment: Services and Challenges", SERSC-International Journal of Database Theory and Application, vol. 10. No. 5, pp. 23-32, May, 2017.

[11] Google's Firebase Notes: https://firebase.google.com/docs/ml-kit/object-detection

[12] A. Yilmaz, O. Javed, M. Shah, "Object tracking: A survey", Acm computing surveys (CSUR), vol. 38, issue 4, pp. 1-45, 2006.

[13] Piccardi, Massimo. "Background subtraction techniques: a review", IEEE, International Conference on Systems, Man and Cybernetics (IEEE Cat. No. 04CH37583). Vol. 4. Pp. 3099-3104, 2004.

[14] J. M. McHugh, J. Konrad, V. Saligrama, P. M. Jodoin, "Foreground-adaptive background subtraction." IEEE Signal Processing Letters, 16(5), pp 390-393, 2009.

[15] Lee, Sang Uk, Seok Yoon Chung, and Rae Hong Park. "A comparative performance study of several global thresholding techniques for segmentation." Computer Vision, Graphics, and Image Processing, vol. 52, issue 2, pp 171-190, 1990

[16] D. K. Yadav, "Detection of Moving Human in Vision based Smart Surveillance under Cluttered Background: An Application for IoT", Chapter-12 in From Visual Surveillance to Internet of Things: Technology and Applications, Taylor & Francis, March, pp. 161-173, 16-Oct, 2019.

[17] C. Würthner, "Object Detection and Tracking with Firebase ML Kit and camera - ML Product Search (Part 3)", Notes, 2019.

[18] H. Dwivedi, "Building a real-time object detection app on Android using Firebase ML Kit", Notes, 2019.

[19] B.D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in Proceedings of the 7thInternational Joint Conference on Artificial Intelligence, pp. 674-679, Aug. 1981.

[20] D. Feng et al., "Deep Multi-Modal Object Detection and Semantic Segmentation for Autonomous Driving: Datasets, Methods, and Challenges", IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 3, pp. 1341-1360, March 2021.

[21] A. Němcová *et al.*, "Multimodal Features for Detection of Driver Stress and Fatigue: Review", IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 6, pp. 3214-3233, June 2021.

[22] D. K. Yadav, K. Singh, "Adaptive Background Modeling Technique for Moving Object Detection in Video under Dynamic Environment", International Journal of Spatio-Temporal Data Science, Inderscience, vol. 1, no. 1, pp. 4-21, Jan, 2019.

[23] L. Sharma, D. K. Yadav, A. Singh, "Fisher's Linear Discriminant Ratio based Threshold for Moving Human Detection in Thermal Video", Infrared Physics and Tech., Elsevier, vol. 78, pp. 118-128, 2016.

[24] M. Yazdi, T. Bouwmans, "New Trends on Moving Object Detection in Video Images Captured by a moving Camera: A Survey", Computer Science Review, Elsevier, pp. 1-66, March 2018.

[25] S. Maqbool, M. Khan, J. Tahir, A. Jalil, A. Ali, J. Ahmad, "Vehicle Detection, Tracking and Counting", IEEE, 3rd International Conference on Signal and Image Processing, pp. 126-132, 2018.

[26] Y. Youssef, M. Elshenawy, "Automatic Vehicle Counting and Tracking in Aerial Video Feeds using Cascade Region-based Convolutional Neural Networks and Feature Pyramid Networks", Transportation Research Record, pp. 1-14, March 2021.

[27] https://towardsdatascience.com/bounding-box-prediction-from-scratch-using-pytorch-a8525da51ddc

[28] R. C Gonzales, R. E. Woods, " Digital image processing", 4th Ed., Pearson Education, pp. 1 – 1026, 2018.

[29] J. Zhang, X. Jia, J. Hu, K. Tan, "Moving Vehicle Detection for Remote Sensing Video Surveillance with Nonstationary Satellite Platform", IEEE, Transactions on Pattern Analysis & Machine Intelligence, vol. , no. 01, pp. 1-1, 5555,

[30] X. Chaojun, Y. Qing, L. Jianxiong, L. Liang, "Research on Vehicle Detection Based on YOLOv3", 2nd International Conference on Information Technology and Computer Application, Guangzhou, China, pp. 433-436, 2020.

[31] S. Paris, P. Kornprobst, J. Tumblin, F. Durand, "A Gentle Introduction to Bilateral Filtering and its Applications", Notes, MIT USA, pp. 2-45, 2017. (https://people.csail.mit.edu/sparis/bf_course/course_notes.pdf).

[32] S. Paris, F. Durand, "A Fast Approximation of the Bilateral Filter Using a Signal Processing Approach", In: Leonardis A., Bischof H., Pinz A. (eds) Computer Vision – ECCV, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, vol 3954, pp. 568-580, 2006.

[33] T. Yang, X. Wang,, B. Yao, J. Li, Y. Zhang, Z. He, W. Duan, "Small Moving Vehicle Detection in a Satellite Video of an Urban Area", Sensors (Basel), vol. 16, issue 9, 1528, pp. 1-15, Sept. 2016.

[34] W. Ao, Y. Fu, X. Hou and F. Xu, "Needles in a Haystack: Tracking City-Scale Moving Vehicles From Continuously Moving Satellite", IEEE Transactions on Image Processing, vol. 29, pp. 1944-1957, 2020.