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, India
sneha riet@gmail.com[a], dileep252000@gmail.com[b], farah.tabassum1999@gmail.com[c], adivi2214@gmail.com[d]

Abstract

In the current scenario, the society is looking for various applications, technology upgradations, and the number of real-time video systems. Since last decade, some active around the world and the demand is exponentially needed with upgradation. So, there is a requirement for such application that should run under the cloud environment. So, this research paper presented in good idea and investigated a mechanism for video surveillance that work under cloud environment. Thus, this paper developed a frame difference scheme based on background subtraction for detection of moving vehicles in a foggy environment. Here, input is provided through cloud environment and output is also delivered to the cloud environment. This work is presented in five distinct sections: (i) to compute differences between two adjacent frames. (ii) To change the moving object to grayscale with a threshold and update the background. (iii) To blur and smooth out the edges and remove noise and the threshold is used to avoid unusual false positives. (iv) To apply the morphological operators to improve the quality of evidence, (v) To demonstrate the shape of a moving object. The final outcome depicts the better detection quality of the moving vehicles in foggy environment.

Keywords: Object detection, Google's Firebase, Foggy Environment.

1. Introduction

The main purpose of scrutinizing and analyzing computer vision is to simulate the behavior and types of the human eye directly with the help of computers and then to develop systems that reduce human effort. It shows a basic object recognition block diagram. With the development of information technology, people are increasingly adopting digital video [1 2, 3, 4]. Detecting and tracking moving targets is a very important application direction. Mobile target detection technology distinguishes moving objects and backgrounds and extracts moving targets from video applicable to banks, highways, homes, and other intelligent surveillance systems [5, 6, 7, 8].

In video sequence analysis, detecting moving objects is constantly in focus and difficult. Long-term moving object detection research has offered a variety of traditional methods, including image differentiation, optical flow, and background extraction. The use of image differentiation and background extraction to detect moving objects has recently received a lot of attention. A superior method of recognizing objects based upon the differences between frames and background model, is proposed. It reduces the number of spirits in removing the critical gaps and background in the differences between the frames and effectively detects moving objects in the

* Correspondence Author

presence of background mutations. However, in complicated dynamic situations, detection is not accurate, as false positives will occur in most cases. This means that a misclassified background due to environmental or lighting influences on the background frame will follow any detected object. Sometimes the slow motion of leaves, shadows, and blockages are temporarily recognized as the foreground.

This article suggests a new way to detect moving objects. Conception of mixture of background subtraction and frame differences in this method, including videos stored in Firebase Storage, and static backgrounds showcases the following new attributes: 1) In the method presented, images can be sequenced in a time series and significantly fill in the blanks of foreground objects 2) bilateral blurring removes noise because this is a synthesis method; 3) image correction and morphological processing techniques are included in the process; 4) Contour detection, which allows users to get information that is more accurate.

The main motivation behind this work is problems associated with bad weather while driving. While the entire car body is fogged up, the windshield and windows are most visible and can create dangerous driving situations. In the thick fog, the driver is unable to see beyond the limits of his own vehicle. Driving in such conditions can be very challenging and also cause accidents as it 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 markings. 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 for every road user [16, 17, 18, 19, 20, 21]. Hence, the proposed method works optimally and detects objects on the screen even in an inclement weather.

This document mainly focuses on a sequence of video frames. *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*.

This document mainly focuses on a sequence of video frames. 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.

2. 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.

R. Li *et. al.* [4] has proposed method for the moving object in the video that has been detected by the cloud sever. It also shows the computing ability of the suggested cloud platform and enables utilization of the available resources. It also avoids having to rely on mobile devices for the limited processing capacity.

S. 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. 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 the real-time natural issue such as fog, snow, haze, rain, or sun glare and these are very dangerous for drivers. For visibility problem, the driver face problems while driving. This paper develops a method to improve the visibility in adverse weather conditions.

Rahul 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 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 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 video in cloud Environment. This paper work also focuses on the real-time based problematic challenges.

S. Yadav et. al. [11] also developed a method for enhancing services in transportation system. This work focused to automate the Indian transportation system through intelligent searching and retrieving mechanism in the amazon's elastic compute service.

3. Proposed Model for Moving Object Detection

In this section, the proposed a model is developed using background extraction techniques that used the background model and a foreground model. During background subtraction, it develops background independently at each location of each pixel relative to the previous pixel value. Here, the bilateral blur filter is

applied for improvising the model. To ensure uniformity, frames are drawn and converted to grayscale levels. Frame differencing stores different coordinates in grayscale1 and grayscale2 frames to detect irregularities in the two frames. Bilateral Blur is used to smudge the frame to cause saturation and dilation to make it easier to find contour areas. All of this acts as frame saturation for more accurate motion results. The basic working model of the suggested work is shown in *Figure-1* and Similarly, *Figure-2* depicts the it's working with Firebase Google's cloud platform.

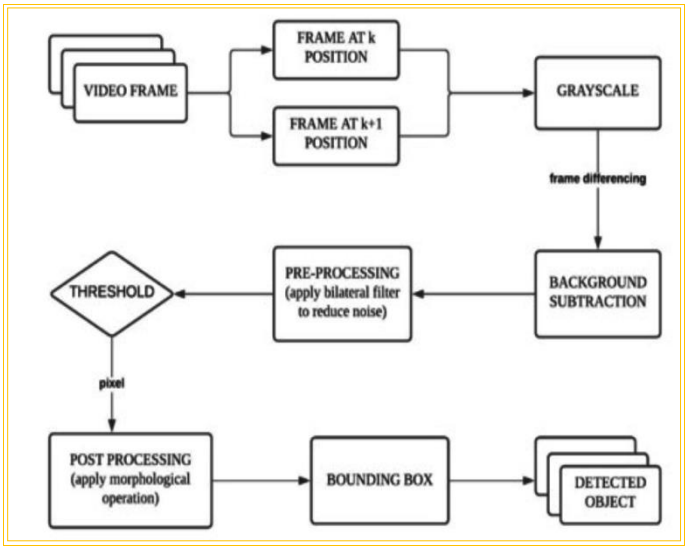


Figure 1. Basic work flow of proposed method.

4. Algorithm and Procedure

This step is performed in two main steps:

- Open the application: Here, the suggested task is started through the application.
- Cloud Data Connectivity: This task is experimented through the connectivity using Google's Firebase platform in the cloud environment. The proposed task is experimented through the following steps.

1. Firebase

The Firebase [11, 17, 18] is a platform provided by Google to accelerate application development. It offers BaaS or backend as a service, which means that Firebase takes care of the cloud infrastructure and all backend requirements. This allows users to grow and deploy faster. The Firebase has some great products; it also enables hosting, and has APIs for machine learning tasks such as predictive text, image captions, and more.

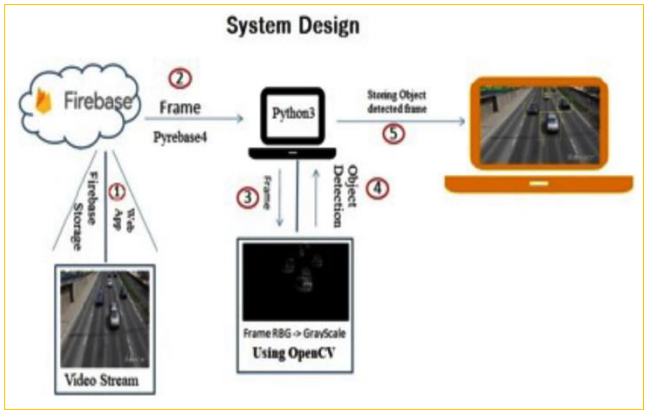


Figure 1: System design to demonstrate connection of Firebase to PyCharm

2. Firebase console

In order to work at Firebase, Google account is a must. Open the Firebase console and register and name the app with Firebase. It has the following project environment according to the requirements: (i) iOS for Apple, (ii)

Android, (iii) Network. In the Firebase console click web setup link present on the Auth tab to get the authDomain, databaseUrl, apiKey and storageBucket variables needed to link to 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 to the application to use Firebase. The interface of Python's to the Firebase REST API is Pyrebase i.e. one can use Python to manipulate your Firebase database. Pip is required to install Pyrebase and its dependencies which are a package manager to install and manage packages of Python received from third-party depository. The Pyrebase apps can use many Firebase services:

1. firebase.auth() – Authentication
2. firebase.database() – Database
3. firebase.storage() – Storage

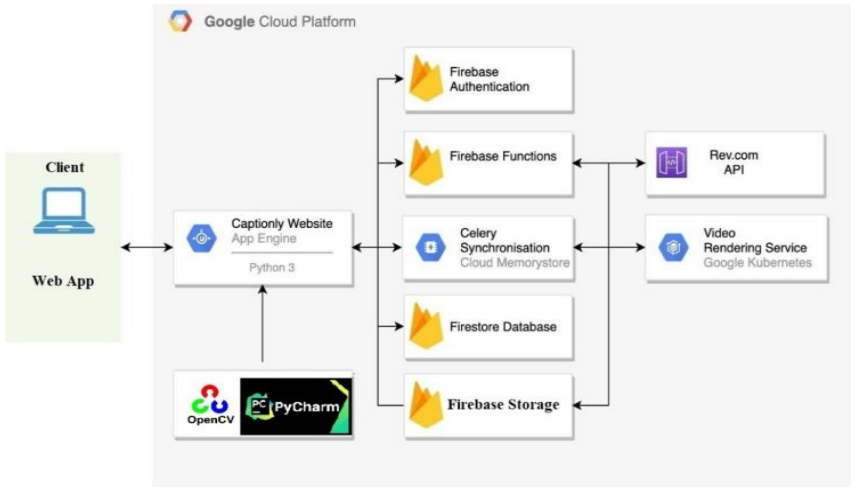


Figure 2. Architectural Diagram of Cloud (Firebase)
https://blog.smartandnimble.com/content/images/2019/11/captionly_architecture.jpg

4. Firebase Storage - Read video through frames

Cloud Storage provides additional protection for Google when uploading and downloading files for the Firebase app, irrespective of network quality. SDKs are used to accumulate video, images, audios, or other content generated by the user, Google Cloud Storage is used to retrieve the same data on the server.

With the help of the child method, paths can be built to data with the Storage service and the download method takes the path to the saved database file.

A. OpenCV

OpenCV is an open-source library for computer vision, machine learning, and image processing and plays a vital role in real-time work in today's systems. It can be used to refine videos and images to detect faces, objects or even the handwriting of a person. To perform object recognition, the following modules must be imported.

1. import cv2 - python binding library for solving computer vision problems.
2. import imutils - a set of functions to facilitate basic image processing operations and viewing Matplotlib images.

3. import time - useful for knowing and analyzing each object found in the video based on its time.

B. Grayscale

The grayscale image is converted from other color spaces such as RGB, CMYK, HSV, etc. It varies between completely black and completely white. The image contains grayscale information and no color information. A typical grayscale image contains 8-bit / pixel (bpp) data. In other words, this image type contains 256 different color tones from 0 to 255, with 0 black and 255 white.

This reduces the size and complexity of the model and is important because other algorithms are adapted to work only with grayscale images.

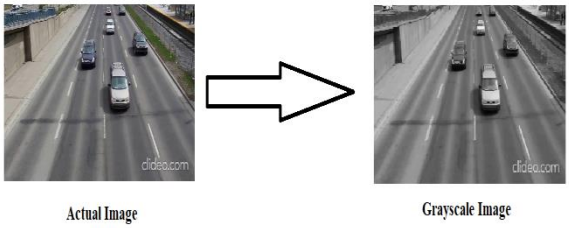


Figure 4(A): Result of the execution of grayscale dependent on picture successions [2]

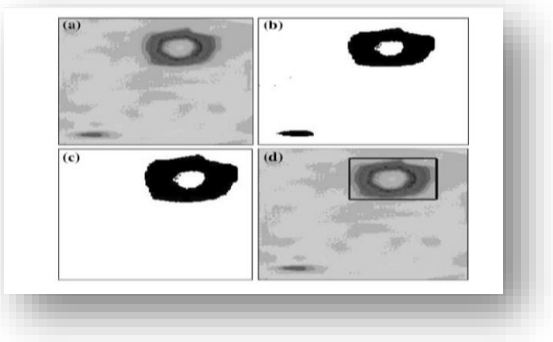


Figure 4(B) Detection result. (a) Grayscale image; (b) Grayscale threshold image; (c) Grayscale threshold image after clutter removal; (d) Grayscale image with object outlined

C. Background Subtraction

The background subtraction method allows a foreground to be extracted for advanced processing. It is a generally used approach for detecting moving objects in videos from fixed/static camera. It is the method of sorting out foreground objects from the background in a cycle of video frames.

The new objects detected in the image scene are based upon the method of differentiating between frames. The background model is used as reference model and difference on the new frame used to search movement and new object. When the object is detected, its state is either static or moving. The method here works with frame pixel differences to search static or moving objects based upon the calculated location. Assumption: Nothing but a moving subject in consecutive pairs of images.

In this section, cv2.absdiff() function is applied to determine the absolute difference between the pixels of two image arrays. This way we can only extract pixels from moving objects. To use cv2.absdiff, first convert the image to grayscale (grayscale is the grayscale area that changes from black to white). The frame differencing function can be computed by using the given equation (1) and detects object between two frames.

$$\boldsymbol{Fd}(\boldsymbol{k},\boldsymbol{k}+\mathbf{1})=|\boldsymbol{Fk}+\mathbf{1}-\boldsymbol{Fk}| \qquad (1)$$

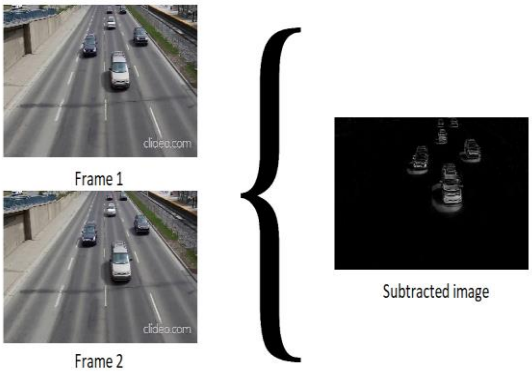


Figure 5. Result of frame differentiation dependent on picture successions

D. Blurring – Bilateral Filter

Image blur refers to making an image less vivid or less vivid. Various low-pass filter cores are used to accomplish this. The bilateral filter is a non-linear image smoothing filter, preserves edges and reduces noise. It replaces each pixel's intensity with a weighted average of nearby pixels' intensity values. Similar as 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 utilised by the Gaussian filter. The second factor considers the intensity difference between the neighbours and the estimated pixels.

$$\boldsymbol{BF}\,[\mathbf{I}]_{\boldsymbol{p}} = \frac{\mathbf{1}}{\mathbf{W_p}} \sum_{\boldsymbol{q}\varepsilon \boldsymbol{S}} \mathbf{G}\boldsymbol{\sigma_r}\left(\left|\left|\boldsymbol{p}-\boldsymbol{q}\right|\right|\right) \mathbf{G}\boldsymbol{\sigma_r}\left(\left|\mathbf{I_p}-\mathbf{I_q}\right|\right)\mathbf{I_q} \quad (2)$$

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 colors 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.

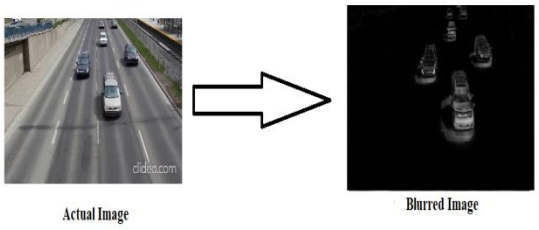


Figure 6. Result of the execution of bilateral blurring dependent on picture successions

E. Thresholding

The threshold (T) is a very popular segmentation technique used to separate an object from its background. The threshold [9, 15, 16]is determined by comparing the value of each pixel in the image (pixel intensity) to a specific value of 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 than the threshold value.

Now, these two groups receive different scores depending on the type of segmentation.

Threshold:

The pixel value is set to 0 if it is less than the threshold [9, 15]; 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 also uses threshold (T) for classification for computing maximum value of pixel.

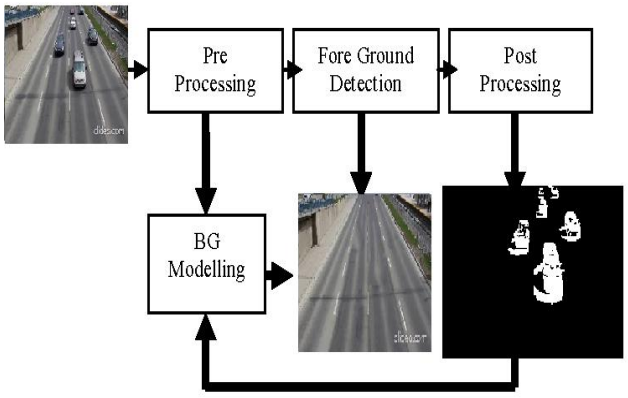


Figure 7. Working model of proposed method with outcome

F. Dilation:

Dilation is performed on binary images. The main effect of dilating a binary image is to continuously enlarge the boundary of the foreground pixel area. Thus, the pixel area in the foreground expands as the holes in this area get smaller.

The dilate() function accepts the following parameters.

- The first parameter is a source image with a number of channels (all processed independently of each other)
- The second parameter is the kernel element, the origin of which is determined by the anchor (standard (-1, -1), that is, in the middle of the structuring element.

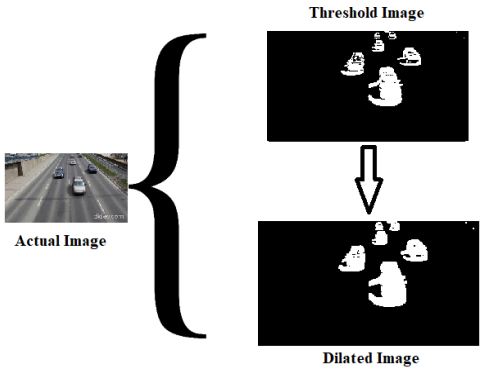


Figure 8. Result of the execution of dilation dependent on picture successions

The third parameter iteration shows how often the process should be repeated. It is sometimes useful to orient the image to edge pixel readout or when the image is irregularly shaped. This can be done using the "bordertype" and "borderValue" arguments.

$$\mathbf{cfn}(\mathbf{x},\mathbf{y}) = \begin{cases} \mathbf{maxVal} & \mathbf{if\ src}(\mathbf{x},\mathbf{y}) > \mathbf{T} \\ \mathbf{0} & \mathbf{otherwise} \end{cases} \quad (3)$$

G. Bounding Box

A bounding box is an imaginary rectangle that serves as a starting point for finding an object and creating a collision box for that object. To draw these rectangles on the image, sketch the objects of interest in each image, and determine their X and Y coordinates, this work, uses contour recognition. This makes it easier for machine learning algorithms to find what they are looking for, identify collision paths, 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 image.

In order to correctly identify the bounding box, we need to convert the image to a monochromatic color format (such as grayscale) and then apply a binary threshold. Applying binary thresholds makes the object completely black and white. The object's edges are completely white with the same color intensity. In fact, this is necessary for the contour recognition algorithm to function properly. It detects the boundaries of objects with white pixels (and of course with the same intensity, since each white pixel has a value of 255). Black pixels with a value of 0 are used as a background and ignored.

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

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

And shows modified images, contours and hierarchy.

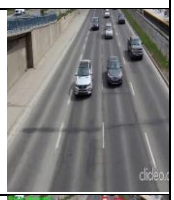
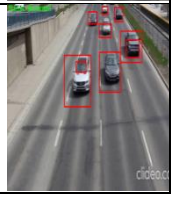
A. Finally, the algorithm will detect the objects in the frames.

B. And in the end the processed data will detect object and send the feedback to the user so that the user can take the appropriate action immediately.

C. Close the application.

5. Experimental Setup and Result Analysis

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

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

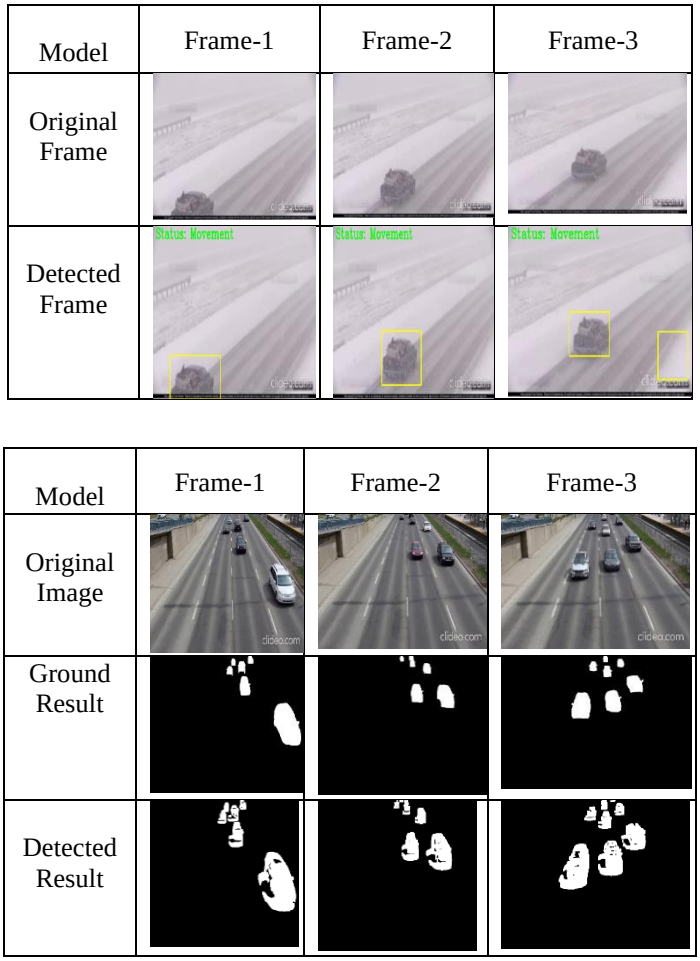


Figure 9. Detection result. (a) normal weather (b) Snow/foggy Weather; (c) Binary output

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

6. Conclusion

In the real-time environment, during foggy weather condition, it's very difficult to drive safely. In adverse weather condition such as fog, the accidental cases happened on the highway in foggy or snow environment due to poor vision or blurred vision. So, this paper presents a background subtraction based method for moving vehicle detection in the foggy environment. All this work has been carried out in Google's Firebase platform that provides a new direction for video surveillance application in the area of computer vision. It also demonstrates better detection results and attracts attention of researchers in this direction too. However, this work is experimented over OpenCV, it gives less execution time.

In the future, this program can be used for cars on autopilot and efficient in passing obstacles.

References

[1] Rasras, R. J., El, E. I. M., & Skopin, D. E. Developing a new color model for image analysis and processing. Computer Science and Information Systems, 4(1), pp 43-55, 2007.

[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. Sens Imaging 11, 113–130 (2010). https://doi.org/10.1007/s11220-010-0056-8.

[3] R. Li, X. Xie, P. Wang and C. Jin, "Cloud-based moving object detection for mobile devices," 2017 20th Conference on Innovations in Clouds, Internet and Networks (ICIN), 2017, pp. 100-102, doi: 10.1109/ICIN.2017.7899396.

[4] Guo, R. J. Liu, N. Li, S. Liu, F. Chen, B. Cheng, J. Duan, X. Li, C. Ma, "Pixel-Wise Classification Method for High Resolution Remote Sensing Imagery Using Deep Neural Networks" . ISPRS International Journal of Geo-Information, 7(3), 110. 2018. https://doi.org/10.3390/ijgi7030110.

[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. doi: 10.1109/ISCON47742.2019.9036216.

[6] Miclea, R.-C.; Ungureanu, V.-I.; Sandru, F.-D.; Silea, I. Visibility Enhancement and Fog Detection: Solutions Presented in Recent Scientific Papers with Potential for Application to Mobile Systems. Sensors, 21, 3370, 2021. https://doi.org/10.3390/s21103370

[7] R. Singh, S. Singh and N. Kaur, "A Review: Techniques of Vehicle Detection in Fog", Indian Journal of Science and Technology, Vol 9, issue 45, pp 1-4, 2016. DOI: 10.17485/ijst/2016/v9i45/106793.

[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] Dileep Kumar Yadav, Karan 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, 4-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] Yilmaz, A., Javed, O., & Shah, M. Object tracking: A survey. Acm computing surveys (CSUR), 38(4), 13-es, 2006.

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

[14] McHugh, J. M., Konrad, J., Saligrama, V., & Jodoin, P. M. "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 52.2: 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 7th International Joint Conference on Artifitial Intelligence (IJCAI), 24-28, Pp. 674-679, August, 1981.

[20] D. Feng et al., "Deep Multi-Modal Object Detection and Semantic Segmentation for Autonomous Driving: Datasets, Methods, and Challenges," in IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 3, pp. 1341-1360, March 2021, doi: 10.1109/TITS.2020.2972974.

[21] A. Němcová et al., "Multimodal Features for Detection of Driver Stress and Fatigue: Review," in IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 6, pp. 3214-3233, June 2021, doi: 10.1109/TITS.2020.2977762.