EAI Endorsed Transactions
on Smart Cities
Research Article EAI.EU

# Vehicle counting application utilizing background subtraction method with large-scale camera data

Mien Phuoc Doan[1,*], Vu The Tran[2] and Sy Ngo Van[3]

[1]Tra Vinh University. 126 Nguyen Thien Thanh Street, Ward 5, Tra Vinh City, 87000, Vietnam
[2]Institute for Research and Executive Education, The University of Danang. 158A Le Loi Street, Hai Chau District, Da Nang city, 50000, Vietnam
[3]Vietnam Research Institute of Electronics, Informatics and Automation, Da Nang City, 50000, Vietnam

## Abstract

In modern society, people are increasingly using cameras at home, in shops, and on the streets. Traffic systems have also invested in building more surveillance camera systems. The data collected by cameras contains valuable information for traffic regulation and recording traffic violations. The challenge is how to effectively use this data. In this article, we will discuss the use of real-time data from surveillance cameras on some roads in Da Nang City for vehicle counting using background subtraction methods. Additionally, we also tested the detection of red-light violations to contribute to the development of a smart traffic system. So, the use of background subtraction in analysing real-time data from surveillance cameras can greatly improve traffic management.

**Keywords:** Large-scale camera, background subtraction, Da Nang City

Received on 05 April 2023, accepted on 09 March 2024, published on 17 April 2024



doi: 10.4108/eetsc.3211

## 1. Introduction

Traffic is always the most frequently mentioned issue in modern society. Safety in traffic is always a major challenge, which causes headaches for many managers.

In developed countries, public transportation such as buses and subways are the main means of transportation. The use of cars is also very common. In addition, the infrastructure is developed, and people's awareness of traffic rules is high. Unlike the traffic situation in Vietnam, where there are many types of vehicles on the road and the awareness of participants is not always good. For example, some drivers stop their vehicles on the highway to eat or even reverse on the highway. Traffic violations such as running red lights, driving in the wrong direction, and lane encroachment are common. Therefore, the unpredictability of traffic in Vietnam in general and Danang, in particular, is quite high. Currently, there is no system deployed in Danang to automatically detect behaviors such as lane encroachment and running red lights.

After a trial period, in August 2016, a high-quality camera system at key locations such as Hue intersection, Han River Bridge, Dragon Bridge, Cham Museum, Pham Van Dong Beach, Nguyen Hue Gate (Quang Trung Street), etc., officially became operational. Currently, there is no system that utilizes data from these surveillance cameras.
With the advancement of information technology, reading and retrieving massive amounts of data or using the outputs of various applications on different technology platforms is being paid attention to. However, in the field of traffic, this is still a relatively new issue. Currently, there is no architecture that can read and interpret multiple types of output data using different programming languages. For example, the results of vehicle counting using visual C++ language can be used to connect with a system used to calculate the density using Python language.

*Corresponding author. Email:phuocmien@tvu.edu.vn

In this article, we will focus on addressing the data issues from multiple surveillance cameras. The specific tasks are as follows:

- Receiving camera data (video) from 0511.vn
- Analyzing and storing the data on cloud servers
- Processing the data in real-time

The purpose of our research is to augment the existing amount of data in the study, particularly focusing on traffic images in Da Nang, Vietnam.
In the following section, we provide a brief overview of relevant research. Subsequently, we will proceed with the experimental research steps in part 3, and finally, conclude with the achieved results.

## 2. Related work

In this paper, we focus on researching relevant literature on the use of large-scale camera data and data processing methods from cameras for vehicle recognition and counting purposes.

### 2.1. Research on literature related to large-scale camera data

In the study by [1], the authors designed and implemented an extensive annotation system that provides comprehensive image labels for a large-scale driving dataset. This dataset comprises over 100 thousand videos and is annotated with various types of information, including image-level categorization, object bounding boxes, drivable areas, lane markings, and full-frame instance segmentation. In [2], it is demonstrated that cameras can be initially deployed for one application and simultaneously shared for other analytic applications, showing the potential of using existing cameras for multiple purposes. Despite recent advancements, video analytics platforms for surveillance still have limitations in systematic access to large datasets and video analytics in the presence of noisy data.

In the research [3], a large volume of data was collected and labelled from multiple camera sources. Around 212 webcams were used to collect data, resulting in over 60 million frames. With a large amount of data, deep learning methods can be applied to accurately count the number of vehicles on the road.

In [4], linear transformations are applied to individual features of each pixel with uniform weights across the entire image. Therefore, the accuracy may not be high when the scene captured by the camera is large.
In most studies, the data primarily focuses on cars with clear perspectives, and there are very few instances where multiple types of vehicles on the road appear, such as cars and motorcycles.

### 2.2. Research on related literature regarding vehicle recognition and counting methods

Methods and techniques for tracking and detecting vehicles have been of great interest to researchers both domestically and internationally to create traffic congestion warning systems, monitor the number of vehicles moving in cities, and regulate traffic automatically. Many methods have been proposed to address the issue of traffic congestion, as demonstrated in [5][9]. However, there are two methods that are currently receiving the most attention: sensor-based methods [6] and image recognition techniques for analysing traffic density [7]. In addition, [5] used image processing from video and the results of object detection were studied for the purpose of estimating traffic density and flow.

In [10], methods such as point detection and edge detection were used in the process of detecting and tracking vehicles. It can be said that one of the most important research breakthroughs is object detection in images [8], which serves as a foundation for object detection from videos [11]. In works like [12], methods to distinguish between the front and rear images are used to extract moving vehicles from videos. Some studies, such as [13] and [14], have shown that using feature vectors from input images can be effective in vehicle detection.

The study [15] presents the estimation of vehicle size with near-accurate results by using a set of coordinate mapping functions. Moreover, in [16], a series of enhancement algorithms were developed for object detection using machine learning methods that can detect and classify moving objects based on type and colour.
In [17], an overview of background subtraction steps is provided, where the first step initializes the background with N frames collected from the initial background where the object is in a stationary state. Then, motion detection is performed through foreground detection, which includes classifying pixels as foreground or background by comparing the background image and the current frame. Finally, the background needs to be maintained to update the background over time. The last two steps are repeated continuously during the processing time.

Despite many successful applications of foreign studies in practical operations, it is not possible to directly apply templates to traffic in Vietnam. The traffic system in Vietnam is very complex, including factors such as geographical conditions, types of vehicles, and traffic culture. Especially, there is a wide variety of vehicles in Vietnam, including motorcycles, bicycles, electric bicycles, three-wheeled cars, four-wheeled cars, trucks, buffalo carts, and agricultural vehicles, in addition to pedestrians and livestock on the roads. As shown in **Figure 1**, in this paper, we plan to build a technological platform that can collect signals from camera data through programming interfaces such as NodeJS. The data will pass through a background service layer that is built to standardize all types of data. The input query data is very diverse, and the system allows for data retrieval from video files by directly connecting to the source device, or by retrieving data through an address, or by retrieving through a metadata query request (this is complex data with multiple fields of information).

In addition, the data will be stored in the cloud through protocols such as RabitMQ and NoSQL databases such as MongoDB, Firebase, etc. Application arrays such as vehicle recognition, vehicle counting, pedestrian detection, etc. will be able to connect to the system.

The video streams are first fetched from the cloud storage and decoded to extract individual video frames. Each frame is then processed separately to detect and recognize objects. This approach allows for processing individual frames on cloud resources, resulting in highly informative and scalable information.

The analysed data can be used in transportation fields such as traffic flow analysis, red light violation detection, etc.

To experiment with the proposed architecture, we are used input data from the system, including videos obtained from cameras placed at street corners in downtown Da Nang. After obtaining this data, the system will process it to extract individual frames from each video. Each frame will be processed by a background subtraction method to transform the original image (with color) into a binary mask containing only two types of pixels: black and white. In this case, black pixels correspond to the background and white pixels correspond to the foreground. The next step uses the obtained image along with the object recognition method to segment moving objects. Based on the height and width of the detected objects, we can determine if they meet the requirements or not. If yes, the total number of counted vehicles is increased, and finally, the total number of vehicles is displayed on the screen. The proposed model in this paper is illustrated in **Figure 2**.

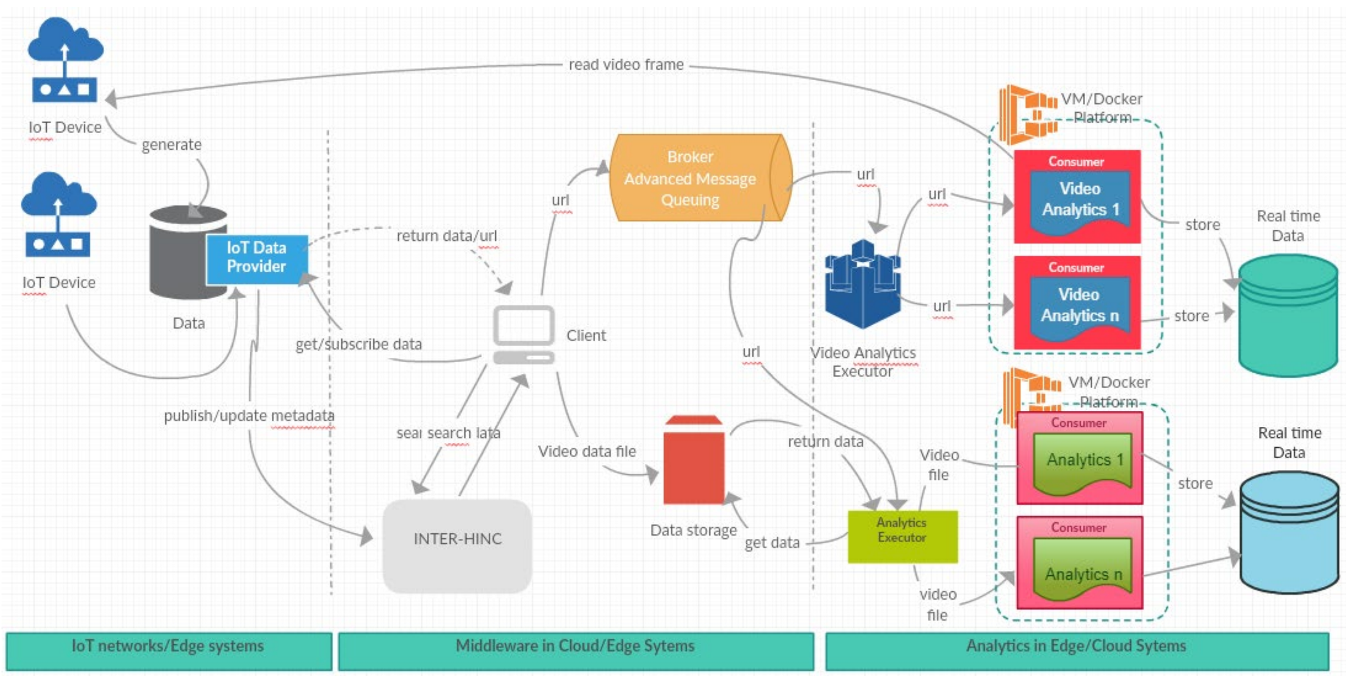


**Figure 1.** System architecture – HAIVAN-CVA System.

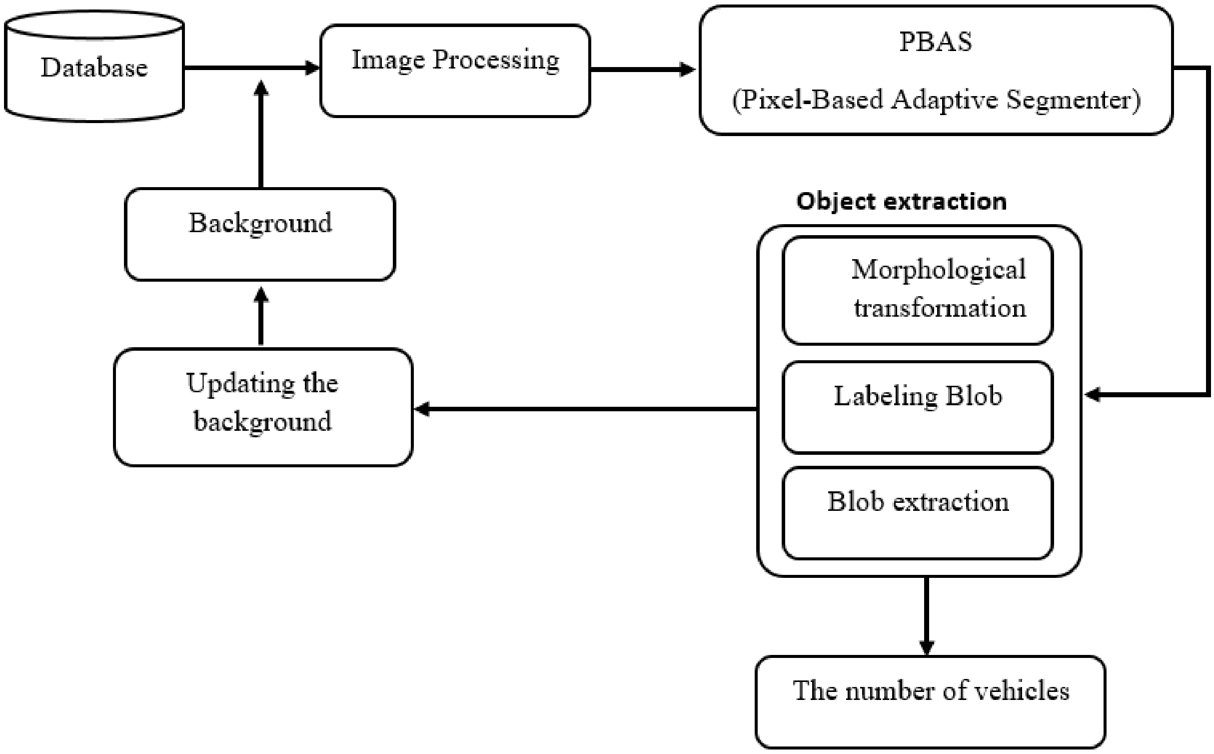


**Figure 2.** The vehicle counting system utilizes the proposed big data architecture.

## 3. Experimental Results

In this paper, to demonstrate the effectiveness of the proposed model for building a large-scale data analytics architecture, a testing system was constructed using a vehicle counting application. This application was implemented using a background subtraction solution combined with blob detection for vehicle recognition and counting. The parameters were then extended to count other types of vehicles.

### 3.1. Experiment with Large-scale Data Source

**Data Source**: In this paper, the data source for serving the selected applications was collected over 1 year from surveillance cameras, as shown in **Figure 3**, in Da Nang city. Each video has a display time of approximately 120 seconds and is formatted according to H.264 standard, with a frame rate of 18 frames per second. The data processing rate for each video is 6.156 kbps. Each video is divided into about 3000 frames and filtered down to approximately 120 frames. Each frame has a size of 160 kb.

**Figure 3.** Camera source

**Hardware Configuration:** The configuration of the client machine is as follows: Intel Core i5 M 520 @ 2.40GHz x 4 CPU, 8GB RAM, 120GB SSD storage. It runs Ubuntu 32-bit operating system. For the project, three virtual servers were set up on cloud computing with Ubuntu operating system, each having 1GB processing speed and 897GB storage capacity. In addition, five Amazon EC2 web services were used to run the vehicle counting applications, and five Amazon EC2 web services were used to run the pedestrian detection applications. The configuration of the Amazon EC2 web servers is shown in Figure 4.

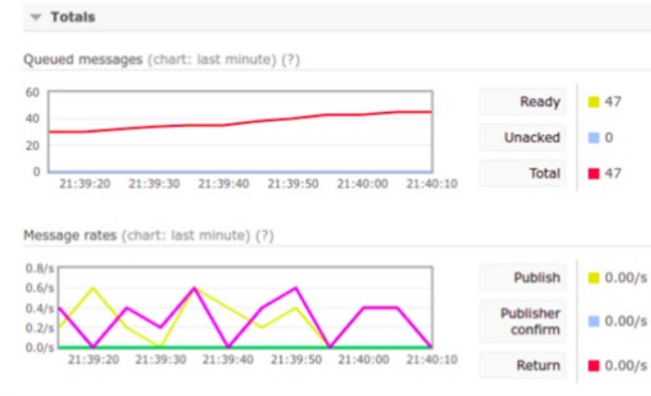


**Figure 4.** Query request information.

**Query data from surveillance cameras:** In this project, NodeJS is used for querying surveillance cameras. The variables and metadata queries are declared as shown in **Table 1**.

The results after querying with the "url" as "http://haivandn.com.vn:3000" are shown in **Figure 5**.


```
[{"_id":"5a40dcc3f342be122606ae65","id":"2co2.vp9.tv@chn@DNG33"
,"name":"Camera gần Trường Mầm Non Tiên Sa","description":"view
1 - DNG33 (1010 IPC7)","address":"106 Quang Trung, Phường Thạch
Thang, Quận Hải Châu, TP. Đà Nẵng","phonenumber":"+84 236
3822348","type":"video","datapoint":"http://2co2.vp9.tv/chn/DNG
33","datapoint-controller":"dng-camera-
provider","fps":"20","conn":"[object Object],[object Object],
[object Object]"},
```


**Figure 5.** Camera addresses from the provider

For example, to view the camera at point 2co2...NG33, the query address would be: url/ camera/2co2...NG33/list/now

- Download videos to a local machine from cloud service using the address: url/camera/:cameraName/. The request will be processed and allowed for downloading. Extracted videos will be temporarily stored and automatically deleted after 3 days from the requested date. The variables to access camera points through the background data processing layer at the address http://35.185.26. 121:9000/global-management-service-1.0 are shown in **Figure 6.**


```
{
  "iotUnitID":   "string",
  "name":   "string",
  "datatype":   "string",
  "measurementUnit":   "string",
  "dataApi":   "string",
  "dataApiSettings": {},
  "connectingTo":{
       "endpoint":   "string"
},
"uuid":   "string"
```


**Figure 6.** Define API

Table 1. Explanation of data retrieval information

| Field | Type | Description |
|---|---|---|
| _Id | string | Primary key |
| Id | string | Frame ID |
| Name | string | Camera name |
| Description | string | Description |
| Address | string | Camera installation address |
| Phone number | string | Hotline number |
| Type | string | Video type |
| Datapoint | string | Video file export address |
| Datapoint-controller | string | Camera service provider |

**Storing Data on Cloud Servers:** As mentioned in the integration structure section, the processed data is stored online in real-time using the Flickr cloud computing server. The real-time storage results on the Flickr cloud server are shown in **Figure 7**.

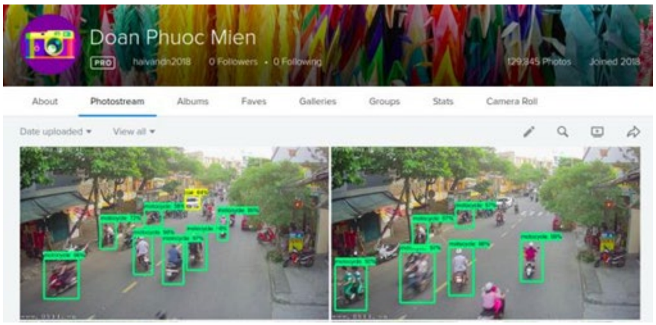


**Figure 7.** Storage of results on the Flickr cloud computing server.

In the architecture of this paper, in addition to direct querying of videos, the system also allows data access through addresses or metadata, and they are stored in real-time using the Firebase database. The results of storing data addresses in the Firebase database are shown in **Figure 8**.

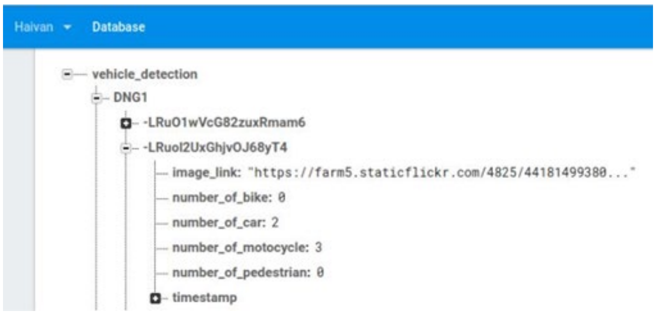


**Figure 8.** Firebase Data Storage History.

Furthermore, the download data usage status is shown in the chart in **Figure 9.**

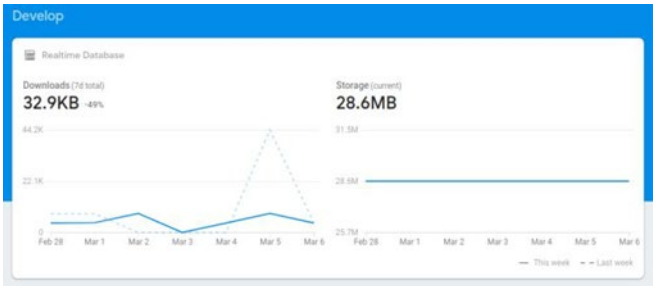


**Figure 9.** Historical Chart of Data Loading on Firebase Storage.

## 3.2. Experimental Vehicle Detection and Counting on the Proposed Architecture Platform

**Car detection Results:** Applying the steps according to the proposed model, we conducted experiments using a video as input. The program successfully recognized the cars in each frame. The results are shown in **Figure 10**.

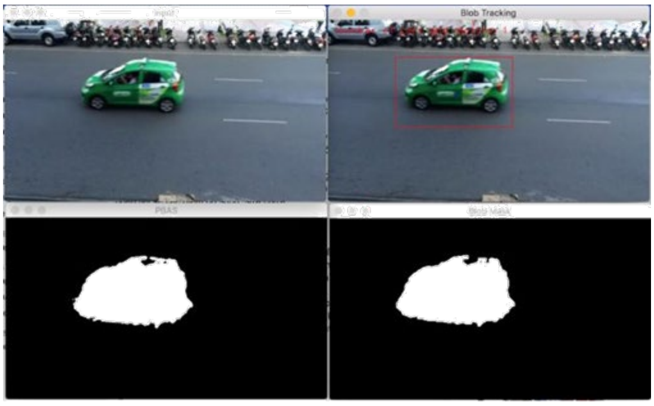

**Figure 10.** Car detection Results

**Motorcycle detection results:** To perform recognition of other objects, specifically motorbikes, in this paper, we have adjusted some parameters of the desired frame size during the object filtering step (using the blob detection method). With these adjustments, we can easily limit the objects to be recognized. **Figure 11** illustrates the specific values of the smallest (minArea) and largest (maxArea) parameters for an object to meet the search requirements. Accordingly, with minArea = 400, the demo program performs recognition of objects corresponding to the size of motorbikes, bicycles, and cars. On the other hand, with minArea = 500, the program only recognizes cars. The motorbike recognition results are shown in **Figure 12.**


**Figure 11.** Object detection frame parameters

**Figure 12.** Motorcycle detection results

## 4. Conclusion

In this paper, we have successfully addressed the challenges of receiving data from a large camera source, real-time data retrieval and storage, and detecting red light violations. The results of detecting red light violations are shown in Figure 13. In the future, we will continue to research and utilize the available camera sources in Da Nang to further improve the quality of traffic services.

**Figure 13.** Red light violation detection

## References


[1] Yu, F., Xian, W., Chen, Y., Liu, F., Liao, M., Madhavan, V., & Darrell, T. (2018). Bdd100k: A diverse driving video database with scalable annotation tooling. arXiv preprint arXiv:1805.04687, 2(5), 6.

[2] S. Jain, V. Nguyen, M. Gruteser, and P. Bahl, "Panoptes: servicing multiple applications simultaneously using steerable cameras." in IPSN, 2017, pp. 119–130.

[3] S. Zhang, G. Wu, J. P. Costeira, and J. M. Moura, "Understanding traffic density from large-scale web camera data," arXiv preprint arXiv:1703.05868, 2017.

[4] V. Lempitsky and A. Zisserman, "Learning to count objects in images," in Advances in neural information processing systems, 2010, pp. 1324–1332.

[5] M. Bayly, M. Regan, and S. Hosking, "Intelligent transport systems and motorcycle safety" Prevention, vol. 28, pp. 325–332, 2006.

[6] J. Sun and J. Sun, "A dynamic Bayesian network model for real-time crash prediction using traffic speed conditions data," Transportation Research Part C: Emerging Technologies, vol. 54, pp. 176–186, 2015.

[7] Jain, N. K., Saini, R. K., & Mittal, P. (2019). A review on traffic monitoring system techniques. Soft computing: Theories and applications: Proceedings of SoCTA 2017, 569-577.

[8] R. Tina and S. G. Sharmila, "Density based traffic signal system," International Journal and Magazine of engineering Technology Management and Research, vol. 2, no. 9, pp. 149–151, 2015.

[9] N. G. Narole and P. R. Bajaj, "A neurogenetic system design for monitoring driver's fatigue: A design approach," in 2008 First International Conference on Emerging Trends in Engineering and Technology. IEEE, 2008, pp. 711–714.

[10] M. Hofmann, P. Tiefenbacher, and G. Rigoll, "Background segmentation with feedback: The pixel-based adaptive segmenter," in 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops. IEEE, 2012, pp. 38–43.

[11] P. G. Michalopoulos, "Vehicle detection video through image processing: the autoscope system," IEEE Transactions on vehicular technology, vol. 40, no. 1, pp. 21–29, 1991.

[12] Q. Cai, A. Mitiche, and J. K. Aggarwal, "Tracking human motion in an indoor environment," in Proceedings., International Conference on Image Processing, vol. 1. IEEE, 1995, pp. 215–218.

[13] D. G. Lowe, "Distinctive image features from scale-invariant key-points," International journal of computer vision, vol. 60, no. 2, pp. 91–110, 2004.

[14] D. A. Forsyth and J. Ponce, "A modern approach," Computer vision: a modern approach, vol. 17, pp. 21–48, 2003.

[15] A. H. Lai, G. S. Fung, and N. H. Yung, "Vehicle type classification from visual-based dimension estimation," in ITSC 2001. 2001 IEEE Intelligent Transportation Systems. Proceedings (Cat. No. 01TH8585). IEEE, 2001, pp. 201–206.

[16] M. Piccardi, "Background subtraction techniques: a review," in 2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No. 04CH37583), vol. 4. IEEE, 2004, pp. 3099–3104.

[17] T. Bouwmans, "Traditional and recent approaches in background modeling for foreground detection: An overview," Computer science review, vol. 11, pp. 31–66, 2014.