Check for updates

Research Article

INTERNATIONAL JOURNAL OF ADVANCED ROBOTIC SYSTEMS

International Journal of Advanced Robotic Systems
July-August 2017: 1–12
© The Author(s) 2017
DOI: 10.1177/1729881417720782
journals.sagepub.com/home/arx
SAGE

Distributed and collaborative real-time vehicle detection and classification over the video streams

Seda Kul, Süleyman Eken and Ahmet Sayar

Abstract

Traffic surveillance cameras are widely used in traffic management and information systems. Processing streaming media in real time is resource and time-consuming processes and even impossible to realize in most real-world applications. To overcome the performance problems in such applications, this article introduces a middleware system based on pub/sub messaging protocol and a dispatcher to preprocess the streams in real time. Experimental results show that proposed middleware may be utilized in different areas such as infrastructure planning, traffic management, and prevention of traffic offenses.

Keywords

Real-time systems, topic-based publish/subscribe, vehicle detection and classification, intelligent traffic management systems, video processing

Date received: 18 February 2017; accepted: 12 June 2017

Topic: Vision Systems
Topic Editor: Antonio Fernandez-Caballero
Associate Editor: Tiziana D'Orazio

Introduction

Traffic surveillance cameras are widely used in traffic management and information systems. There are some studies involving image and video processing on traffic surveillance data, mostly used in traffic management systems. Some of these studies are as follows: automatic vehicle identification,[1] road capacity,[2] traffic density measurement,[3] speed detection,[4–6] traffic violation detection,[7] and specifying car categories.[8–11]

Vehicle classification is a very important process in traffic management systems. Outcomes of the vehicle classification can be used for different aims such as reporting and statistical analysis, determining the asphalt thickness, and searching for vehicles for specific classes which are built by classifying their varying sizes, colors, velocities, and so on.

Advancement in real-time image processing over the Internet draw attention in many application areas. In intelligent traffic/vehicle management systems, transferring traffic surveillance video data from source to destination for further processing is very important. Surveillance cameras have limited processing power and small storage area. Therefore, they can only capture and store data locally and perform some basic image processing. Some even do not have storage or processing capabilities. They only capture data and transfer to the processing units via Internet connections. However, in some traffic emergency systems, real-time image processing on temporal image sequences is

Department of Computer Engineering, Faculty of Engineering, Kocaeli University, Kocaeli, Turkey

Corresponding author:
Süleyman Eken, Department of Computer Engineering, Faculty of Engineering, Kocaeli University, Kocaeli 41380, Turkey.
Email: suleyman.eken@kocaeli.edu.tr

Creative Commons CC BY: This article is distributed under the terms of the Creative Commons Attribution 4.0 License (http://www.creativecommons.org/licenses/by/4.0/) which permits any use, reproduction and distribution of the work without further permission provided the original work is attributed as specified on the SAGE and Open Access pages (https://us.sagepub.com/en-us/nam/open-access-at-sage).

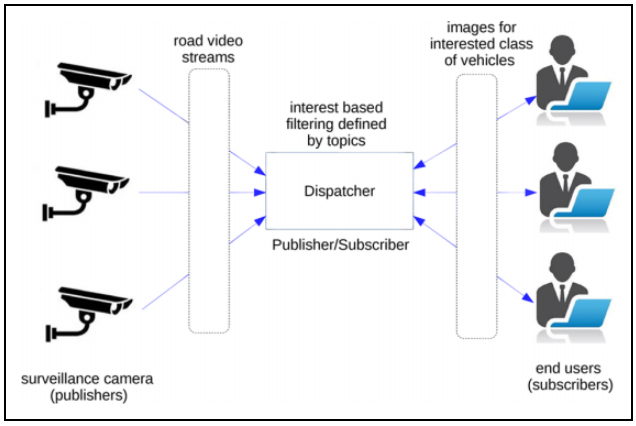

Figure 1. Overview of the architecture.

required. Since video data are a streaming media and get very large in size, the central processing techniques are not efficient in their processing. It even gets worse in the case of real-time scenarios. Moreover, in such systems, they need to be processed and analyzed in a reasonable time period. For the decision making, the right data need to be dispatched to the right places in right time.

In real world, a user may want to get an image or a piece of video of small-sized vehicles (such as sedan), another user may want to get the same information about large-sized vehicles (such as trucks), and one another user may want to get the same video streaming about a vehicle travelling in a wrong lane and wrong direction. To realize such a system, a surveillance camera, a dispatcher, and end users are needed. The surveillance camera transfers video streams to the dispatcher. The dispatcher processes the video stream and chunks it into pieces according to the registered end users and then transfers the related video streams to the registered end users.

The work presented in this article is summarized as a distributed and collaborative real-time vehicle detection and classification over the video streams provided by traffic surveillance cameras. There are two phases in the proposed framework—TPSVedet (see Figure 1). These are (1) classification of vehicles (small, medium, large) and (2) transferring classified images to the registered/subscribed users via topic-based publish/subscribe system. In the first phase, an uncalibrated camera, located on the bridge over a highway, sends captured video to a dispatcher. Then, vehicle images are obtained from videos by video–image processing and vehicle extraction processes. In the last phase, vehicle images are classified and sent to the registered users via topic-based publish/subscribe messaging protocol. Topics correspond to the actual vehicle classes such as car, van, and truck.

To extract vehicle objects from the video frames, moving blobs are detected by a background–foreground extraction algorithm. The classification process is done by extracting geometric features (width, length, height) and binary features (area ratio, aspect ratio, major and minor axis length, diameter, etc.) of detected blobs and by giving those features to different classifiers such as support vector machine (SVM), artificial neural network (ANN), and AdaBoost. The performance of the alternative classification algorithms is evaluated according to the different metrics such as accuracy, recall, and precision.

The remainder of this article is organized as follows. "Related work" section presents the related work. In "Publish/subscribe messaging" section, pub/sub models are presented. The proposed framework is given in "Proposed framework" section. The performance results and their analyses are given in "Experimental setup, results, and analysis" section. The last section concludes the article.

Related work

Vehicle detection and classification

Object classification studies can be divided into two categories: object classification in still image and object classification in videos. The work presented in this article focuses the latter one. Detection and classification of moving vehicles in traffic surveillance videos have been worked by many researchers in recent years.[12–14] However, it is a very compelling process because of increasing number of vehicle models and sizes, even within a single class. Due to noisy background subtraction, change in the size of regions, occlusion, uncontrollable environment conditions (e.g. fog, rain, lighting, and haze), and shadow, the classification task gets even more challenging. Consequently, building robust vehicle classification architecture to tackle those problems is desired in real-world applications.

Features used in classification processes to extract objects in still images can be divided into two categories. These are appearance-based features and geometry-based features. In appearance-based methods, objects (vehicles) are presented as vectors in high-dimensional space. Most used appearance-based feature extraction methods are scale invariant feature transform (SIFT) and speeded-up robust features (SURFs). On the other hand, as geometry-based features, width, length, height, area, and so on are considered.

Zhang et al.[15] developed a length-based vehicle detection and classification system with 97% accuracy rate. Zhang et al.[16] proposed an appearance-based approach for recognizing moving objects such as cars, vans, trucks, people, and bikes by utilizing multi-block local binary patterns. Morris and Trivedi[17] used blob features followed by linear discriminate analysis, fuzzy C-means clustering, and weighted k-nearest neighbor classifiers to classify vehicles. Moussa[18] combined geometry-based (width, length, and height) and appearance-based features (SIFT and SURF) to classify vehicles in multi-classes such as small, medium, and large size vehicles. Then, medium size vehicles are categorized into intra classes such as pickup, sport utility vehicle (SUV), and van. They report true positive rate of

97% for classification. Kul et al.[19] compared three classifiers to classify vehicles in multi-classes such as small, medium, and large size vehicles. Their precision and recall values for three types of vehicles are 80.7, 82.4, and 100 and 97.4, 40, and 50, respectively.

In this context, we use geometry-based features to detect and classify vehicle objects in still images. However, we get the still images as frames from the traffic surveillance video and the processing is done in real time. To do so, for the sake of performance, we introduce a middleware system based on pub/sub messaging protocol and a dispatcher to preprocess the streams in real time.

Topic-based pub/sub models in video surveillance

Besides vehicle detection and classification difficulties, the biggest problem is deploying the large-scale video surveillance system to transmit data over the network. If the system delivers whole video content instead of filtered one, end users may retrieve the unwanted video parts/streams and network bandwidth is wasted. So, the surveillance becomes inefficient. Pub/sub models can be used to improve the efficiency of video retrieval.

There are some applications utilize pub/sub models in traffic management. Zou et al.[20] realize a pub/sub-based real-time video surveillance system. The video surveillance is also realized on the interconnection of wireless sensor networks (WSNs) and wireless mesh networks (WMNs). While WSN event publishers publish text notification as pairs of <attribute, value>, WMN event publishers publish the video content. Jiannong et al.[21] suggest ubiquitous wireless video surveillance system with the ability to detect predefined events and transmit corresponding high-quality video streams to the registered users by means of WSNs. Zou et al.[22] develop delay constrained pub/sub-based large-scale wireless video surveillance system to solve the problem of an optimal subset of brokers for subscription forwarding with a heuristic approach. However, to the best of our knowledge, no previous study has been done on vehicle detection and classification based on topic-based pub/sub model.

Publish/subscribe messaging

The proposed framework consists of geographically distributed computation nodes, a networked camera, and operator nodes. In a distributed system, a number of networked nodes communicate and coordinate their actions by means of various ways such as pub/sub, synchronous request/response, asynchronous message passing, message queues, and tuple spaces.[23]

In the pub/sub system, event producers (publishers) publish events, and events are delivered to interested event consumers (subscribers). Here, subscriptions are subscribers' interests. Communication between publisher and subscriber nodes can be either synchronous or asynchronous. Also, the number of publishers and subscribers may be more than one. Moreover, event services are responsible for delivering the publication to the subscribers.

There are three types of publish/subscribe models: (i) topic based, (ii) content based (aka property based), and (iii) type based. In the topic-based subscription model, messages are published to logical channels (topics). Subscribers in this system will receive all messages which they subscribe. Namely, all subscribers to the same topic will receive the same messages. The real-world examples are news alerts, location-based services, online stock quotes, Spotify playlists, and so on. The topic-based pub/subs can be seen as a special case of the content-based subscription model. In the content-based system, messages are only received from a subscriber if the content of those messages matches rules specified by the subscriber. The subscriber is responsible for classifying the messages. In type-based pub/sub, event topics are replaced with an event type. So, it guarantees a given set of attributes and the subscriber can filter on the contents of those attributes.

In this article, topic-based publish/subscribe messaging model is used, and the Java messaging services (JMSs) open source library is utilized to realize messaging. There are three vehicle classes (car, van, or truck) and corresponding three topics with the same names.

Proposed framework

As shown in Figure 2, the proposed architecture is summarized in the following steps:

- background subtraction and detection of vehicles Region of Interest (ROI) ("Background subtraction and vehicle detection" section);
- extraction of features representing vehicles ("Feature extraction" section);
- dimensionality reduction to reduce dependencies of some features ("Dimensionality reduction" section);
- vehicle classification ("Vehicle classification" section); and
- designing a distributed system for filtering and transferring classified images to the registered users ("Topic-based pub/sub architecture" section).

Background subtraction and vehicle detection

Background subtraction, also known as foreground detection, is an important method in image processing mostly used in applications regarding object detection and extraction. Background subtraction process tries to build a background model which is a reference image used for comparison purposes. Background modeling is a difficult task for foreground detection due to dynamic nature of environments. Therefore, many methods have been proposed in order to find correct background model.[24]

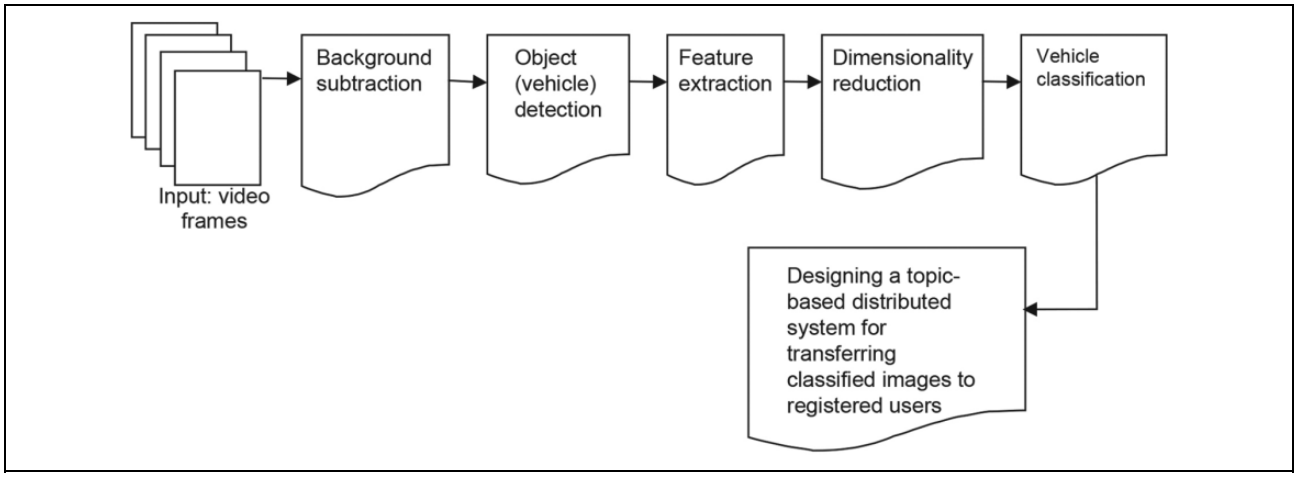

Figure 2. General architecture for TPSVedet.

Table 1. Evaluation results.[19]

Algorithm	Object detection	Noise	Real-time operations
AdaptativeBG Learning	With shadow	Moderate	Very fast
DPWren GABGS	With shadow	Low	Fast
MixtureOfGaussian	With shadow	Very low	Fast
V1BGS DP Eigenbackground BGS	With shadow	High	Slow

Illumination changes, camera movements, and dynamic nature of environment are the most frequently faced problems in background subtraction process. Kul et al.[19] tested and reviewed BGSLibrary algorithms.[25] The results of the review are investigated for real-time systems as tabulated in Table 1. The table shows that adaptive background learning (AdaptativeBG Learning) algorithm is more appropriate for real-time systems.

AdaptativeBG Learning creates a background model by averaging frames over time. This approach uses parameters for blending the current image into the background model.

In Figure 3, α is the learning rate and the value between 1 and 0 determines the speed of learning of the algorithm. As two extreme cases, 0 means background model is not updated and 1 means background model is updated with the previous frame.

Feature extraction

In vehicle classification approaches, features can be extracted in two ways: geometry and appearance based. The geometry-based feature extraction method uses measurements such as length, width, area, and so on of the object which is detected in the frame.[26]

On the other hand, in appearance-based methods, objects (vehicles) are presented as vectors in high-dimensional space. Most used appearance-based feature extraction methods are SIFT and SURFs.

In this work, for the classification of vehicle images, geometry-based feature extraction method is used. The features used for the classification are listed below.

- The area of contour. A contour is a curve tie of all continuous points along the boundary. In other words, a contour is an outline of a figure or body or the line or lines representing such an outline.
- The width and height of the rectangle surrounding the image contour. The rectangle can be considered as minimum bounding box.
- The ratio of the contour is to the rectangle area surrounding the contour.
- The lengths of the major and minor axes of the ellipses surrounding the contour.
- The number of white pixels in the rectangle surrounding the contour.

Aforementioned feature extraction methods are tested and evaluated for their performances in the real-time image processing systems (see Table 2). The results show that SIFT and SURF algorithms are not suitable for real-time systems with their bare forms. In our experiments, we have used an Intel Core i5-3210M 2.5 GHz processor and Windows 10 Pro x64.

The features used in the tests are tabulated in Table 3.

Dimensionality reduction

Dimension reduction methods are the processes of reducing the number of the feature set and commonly used in machine learning. The dimensionality reduction methods can be divided into two steps: feature selection and feature extraction. The feature selection methods are used for finding a subset of features which has the greatest impact toward targeted classification. Unlike feature selection, the feature extraction methods transform the data into the different dimensional spaces. For instance, principal component analysis (PCA) is the most commonly used method for

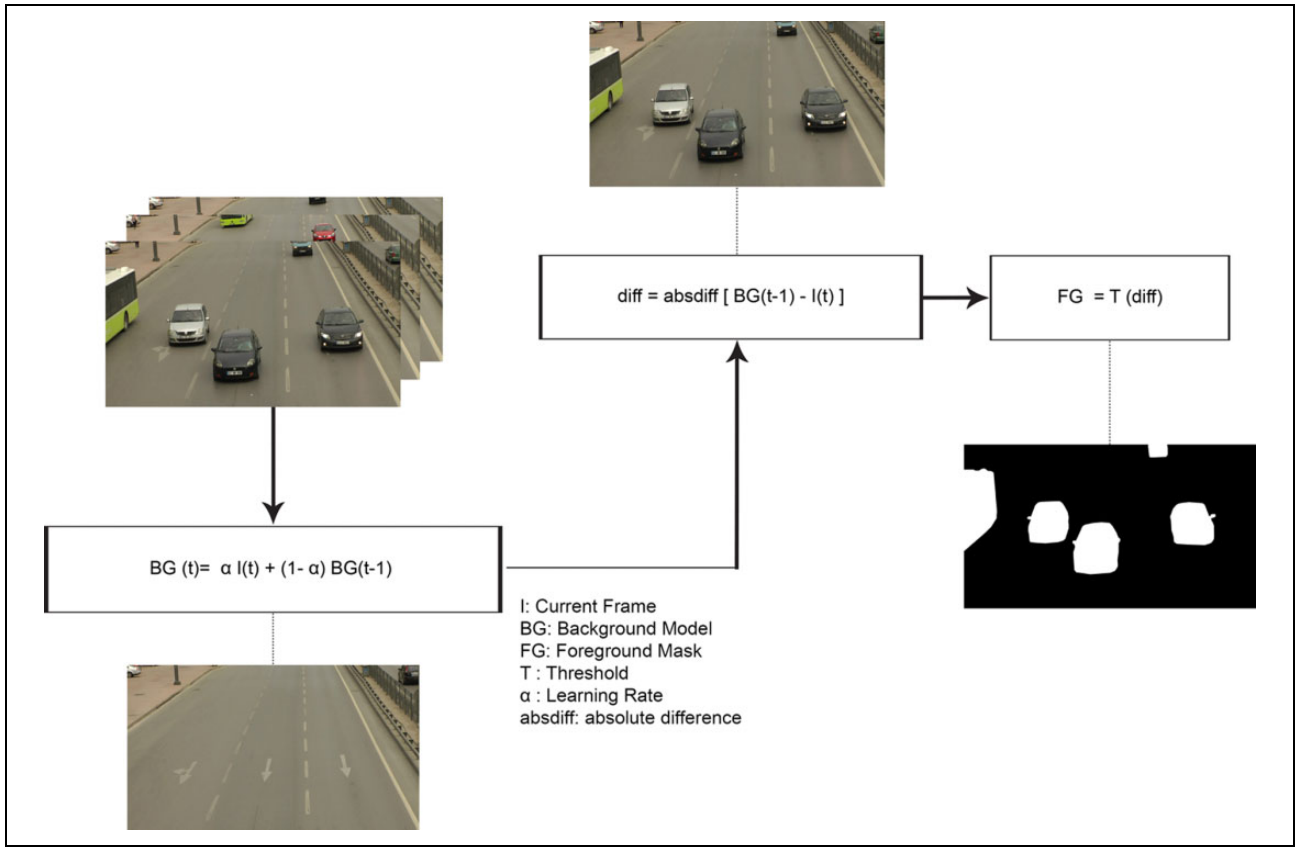

Figure 3. Adaptive background learning.

Table 2. Running times for extraction methods using geometry and appearance features.

Features	Average execution time (ms)
Appearance-based	
SIFT	2.15378e+007
SURF	3362.86
Geometry-based	
Contour	0.08
Width and height	0.60
Ratio of the contour	0.082
Major and minor axis lengths	0.078
Number of white pixels in the rectangle	0.061

SIFT: scale invariant feature transform; SURF: speeded-up robust feature.

Table 3. The features and their range values.

Numbers	Attributes	Value range
1	Contour area	614–3852.5
2	Width of rectangle	26–55
3	Height of rectangle	30–99
4	Area of rectangle	810–5445
5	Major axis of ellipse	35.2–948.4
6	Minor axis of ellipse	25.9–487.2
7	Number of white pixel	660–3977

this purpose but many dimensionality reduction methods also exist.[27,28] PCA does not choose or eliminate some characteristics of the features. It builds new characteristics using all the features as well. In this article, we use feature selection methods for dimension reduction. The reasons for using feature selection are to simplify and accelerate the learning model, enabling shorter classification times and avoiding overfitting.[29]

In the proposed system, dimensionality reduction technique is used to create the best feature subset from the features listed in Table 3. This feature subset is expected to maximize the correct classification rate and has the lowest probability of error. Since the number of features are reduced, overfitting is avoided and outcomes are obtained much faster.

There are some feature selection methods proposed in the literature. They can be divided into the three classes depending on how they build the model: (i) filter methods, (ii) wrapper methods, and (iii) embedded methods. The filter methods are mainly used as preprocess methods and they choose features regardless of the effects of the selected features on the performance of the learning algorithm. The wrapper methods try to find a good combination of attributes using the learning algorithms. The embedded methods try to combine the advantages of wrapper and embedded methods.[30] Hall and Holmes[31] show that

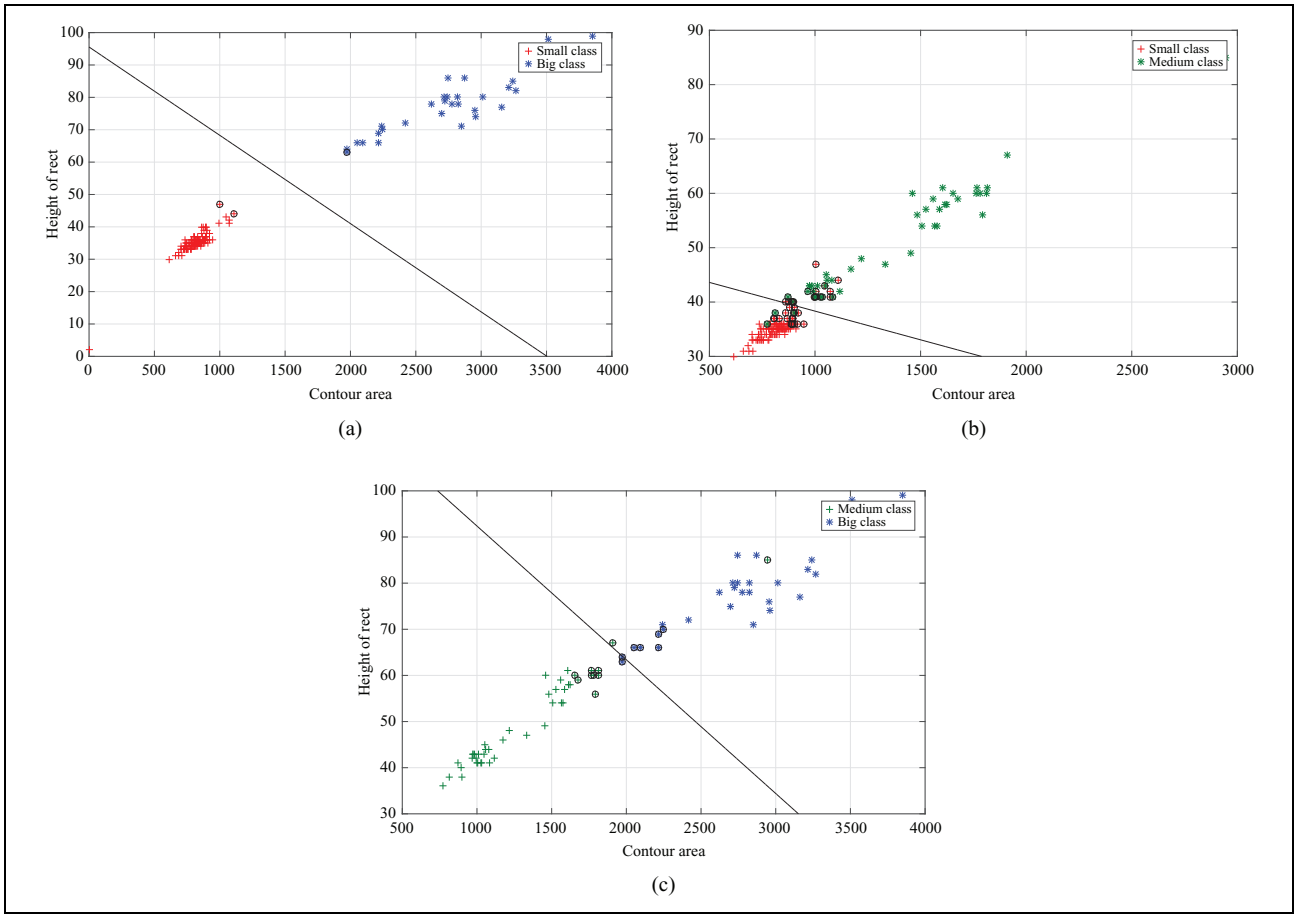

Figure 4. SVM-one versus one approach. (a) Small-large class. (b) Small-medium class. (c) medium-large class. SVM: support vector machine.

wrapper is the best feature selection method. We also use wrapper method for dimensionality reduction for vehicle classification. One of the wrapper methods is the forward search algorithm (FSA) which adds the best result giving feature to each data set in each iteration.[32] The pseudocode for feature selection is given below.

Algorithm 1 Feature Selection

Input: x-features $\in X$
Output: Desired feature set F

1. **Step 1. Initialization**
2. **Initialize feature set**
3. $F = \{\}$
4. **Step 2. Find *f* with lowest error to *F***
5. **while** // not selected desired number of features
6. **for** $i = 1$ **to** feature_count
   Estimate model's error on feature set $(F \cup i)$
   Add x_i with lowest error to F
7. **end**
8. **end**

The subset for the feature set given in Table 3 is calculated by FSA algorithm, and the result set is $F = \{1, 3, 4, 5, 7\}$. Two features are eliminated after applying FSA.

Vehicle classification

Object classification is the categorization of the object based on predefined classes or types. Here, we have three different predefined classes of vehicles, and we want to classify each detected vehicle according to the three classes. These classes are small size vehicles, medium size vehicles, and large size vehicles.

In this study, three different classification methods were used: ANNs, SVMs, and AdaBoost. Neural network model used in this article has five inputs for five features and three outputs. The outputs are class names for small size, medium size, and large class vehicles.

SVM is an effective machine learning method that operates according to the principle of minimizing structural risk in the classification of linear and nonlinear data. In multiclass problems, SVM offers two different approaches. These are one-versus-one and one-versus-all. They provide the solution by formalizing the problem as two classification problems (Figure 4(a) to (c)).

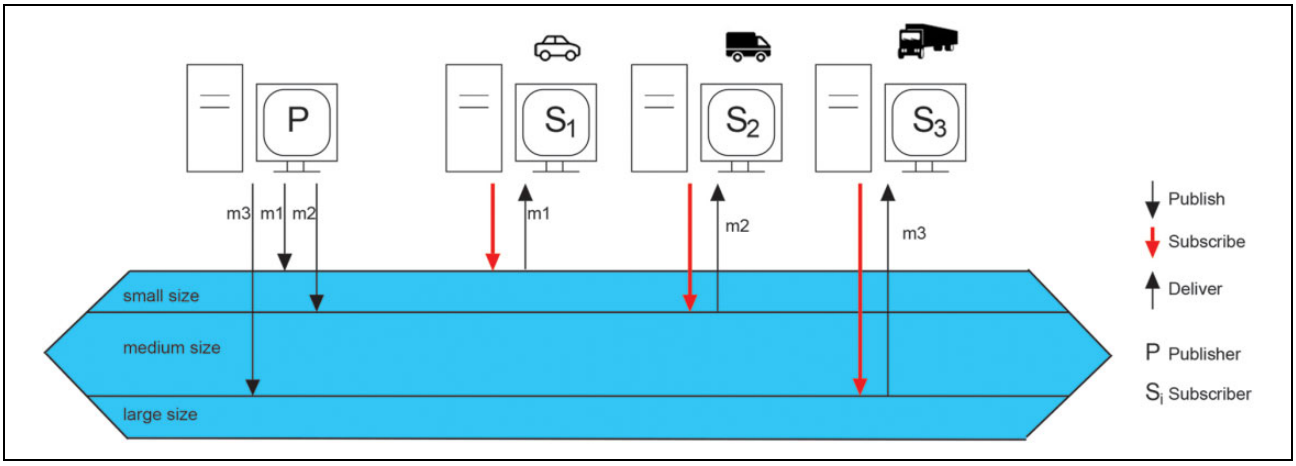

Figure 5. Proposed pub/sub model for vehicle detection and classification.

AdaBoost is another algorithm which is used in the classification problem. AdaBoost is based on the idea of creating a strong classifier by means of first identifying and then combining weak classifiers. The decision boundaries of weak classifiers for each attribute are calculated by taking the weighted average of positive and negative samples. Then a strong classifier is generated using the weakest classifiers with the lowest error rate. The attributes (features) related to weak classifiers that are not included in the strong classifier are eliminated.

All vehicles in the data set are divided into three categories (small, medium, and large) according to their sizes by means of supervisor. For each vehicle category, randomly selected samples are used for training and other samples are used as test sample. When an end user (subscriber) wants images of small-type vehicles, related images are sent to registered users via topic-based publish/subscribe system.

The performance results and analysis of these three classification methods on our data set are given in "Performance results and analysis of classification methods" section.

Topic-based pub/sub architecture

Publish/subscribe architecture is a messaging model where senders are called publishers and receivers are called subscribers. In this architecture, messages are not sent to specific subscribers but instead, messages are published on a specific topic without knowing to whom or when it was sent. In the proposed framework, topic-based pub/sub messaging is realized using JMS specifications. JMS API is an implementation to handle the pub/sub system. It allows applications to create, send, and receive messages reliably and asynchronously way.[33] Algorithm 2 gives the pseudocode for topic-based pub/sub. The publisher sends classified vehicle images with the corresponding topic and all the active subscribers to this topic receive the classified vehicle images.

Algorithm 2 Publish Subscriber Architecture

```
Publisher
1.  connect_server()
2.  If (topic == NULL)
3.      then create topic for each class
4.  end if
5.  while(frames!= NULL)
6.      vehicle = classification_vehicles(frames)
7.      topicPublisher.send(vehicle)
8.  end

Subscriber
1.  connect_server()
2.  while(listen topic)
3.      createDurableSubscriber()
4.      message = subscriber.receive()
5.      if(message != NULL)
6.          get_message()
7.      end if
8.  end
```

The proposed topic-based pub/sub model is given in Figure 5. In this system, users (S_1–S_3) can subscribe events of interests (here, events are sizes of vehicles) and publisher (p) delivers the real-time captured video content of event to the users.

To include an event filter expression, the event service applies a filtering to the content of each published event such as a subscription to all events whose name is truck and velocity is bigger than 90. Similar to SQL, a subscription language enables fine-grained filtering, event correlation, and detection of composite events in content-based pub/sub messaging.

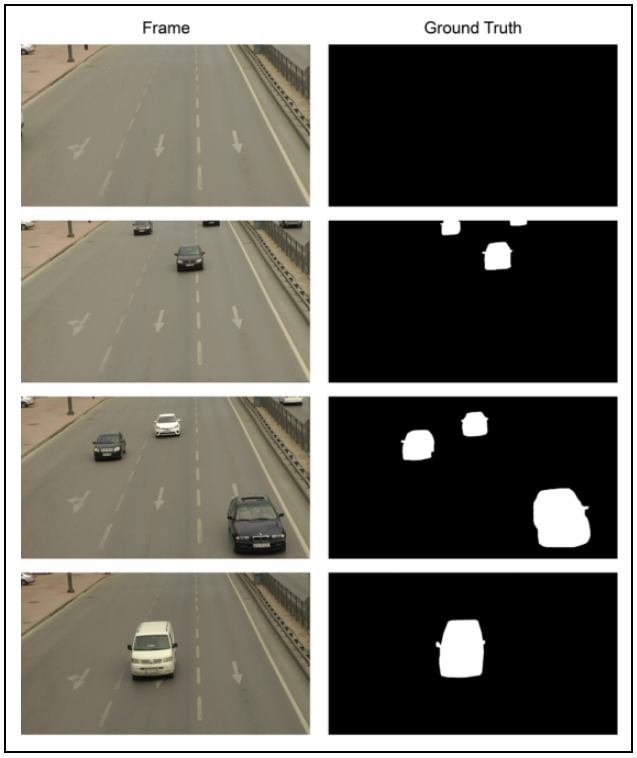

Figure 6. A set of frames and their ground truths from the data set.

Experimental setup, results, and analysis

Building data set

The data set (TPSdataset) used in the experiment was taken at a height of 6 m in the sunny daytime. Three thousand seventy-one frames were obtained from the shootings. For classification, exact references are needed. Ground truth refers to a set of measurements which is much more accurate than the system that we create. For getting ground truth images, we used hand-labeled bounding boxes method. Ground truth images are known to be correct with some of the external information. Ground truth images have been achieved with Photoshop program. Figure 6 shows some frames and their ground truths in our data set. Table 4 shows the distribution of vehicle types.

Experimental setup

In our experiments, we have used an Intel Core i5-3210M 2.5 GHz processor and Windows 10 Pro x64 for the publisher and an Intel Core i5-3317U 1.7GHz and Windows 10 Pro x64 for the subscribers.

Performance results and analysis of classification methods

Extracted feature set is evaluated using accuracy, recall, and precision parameters as the performance measures. The measures are described as below:

Table 4. Distribution of vehicle types.

Size	Type	Numbers
Small size	Car/sedan	124
Medium size	Minibus	15
Medium size	Light commercial vehicle	21
Medium size	Van	9
Large size	Bus	12
Large size	Truck	19

$$\text{Precision} = \text{TP}/(\text{TP} + \text{FP}) \quad (1)$$

$$\text{Recall} = \text{TP}/(\text{TN} + \text{FN}) \quad (2)$$

$$\text{Accuracy} = (\text{TP} + \text{TN})/(\text{TP} + \text{TN} + \text{FP} + \text{FN}) \quad (3)$$

In equations (1) to (3), TP is true positive, TN is true negative, FP is false positive, and FN is false negative. An event is said to be TP if a vehicle is correctly classified and TN when a vehicle is correctly nonclassified. In the related development, an event is said to be FN if a vehicle is incorrectly nonclassified and an FP when a vehicle is incorrectly classified. The precision measure indicates the probability that a (randomly selected) classified vehicle is correct, while recall measures the ratio of correctly predicted positive events. However, the accuracy measure indicates the degree of conformity of the correctly classified vehicles to the ground truth. Table 6 shows performance results for the three classifiers. The bold values indicate the best results. It also summarizes processing times of each classifier for both data sets. In Table 5, TPSdataset 1 refers to the data set with seven features, and "TPSdataset 2" refers to the data set with five features. Table 6 shows confusion matrices of the three classifiers, calculated on both TPSdataset1 and TPSdataset2. Table 7 also shows the examples of correctly classified and misclassified vehicles by different classifiers.

We also test our framework on another data set named BIT-Vehicle. The BIT-Vehicle data set contains 9850 vehicle images, whose sizes are of 1600 × 1200 and 1920 × 1080 and captured from two cameras at the different time and places. All vehicles in the data set are divided into six categories: bus, microbus, minivan, sedan, SUV, and truck. The numbers of vehicles per vehicle type are 558, 883, 476, 5922, 1392, and 822, respectively.[34] In this study, we used 300 samples from the BIT-Vehicle data set. Samples are divided into three categories: large, medium, and small vehicles. The large vehicle class contains two types of vehicles, 36 buses and 64 trucks; the medium vehicle class contains two types of vehicles, 96 microbus and 4 minivans; and small vehicle class contains 100 sedans. So, distribution of classes is made uniform to avoid overfitting, which may affect the final results. Table 8 shows performance results for the three classifiers. The bold values indicate the best results. It also summarizes processing

Table 5. Vehicle classification results on TPSdataset with different sets of features and classifiers.

Classifier	Processing time (ms)	Precision		Recall		Accuracy
ANN TPSdataset 1 (200 × 7)	1.833117	Small size	96.8%	Small size	85.7%	87%
		Medium size	54.3%	Medium size	83.3%	
		Large size	96.7%	Large size	96.7%	
ANN TPSdataset 2 (200 × 5)	1.588715	Small size	**97.6%**	Small size	**96%**	**95%**
		Medium size	84.8%	Medium size	**92.9%**	
		Large size	**100%**	Large size	93.8%	
AdaBoost TPSdataset 1 (200 × 7)	0.069416	Small size	95.2%	Small size	94.4%	92%
		Medium size	82.6%	Medium size	82.6%	
		Large size	93.3%	Large size	96.5%	
AdaBoost TPSdataset 2 (20 × 5)	0.058551	Small size	94.3%	Small size	**96%**	92%
		Medium size	84.9%	Medium size	81.2%	
		Large size	93.3%	Large size	93.3%	
SVM TPSdataset 1 (200 × 7)	0.016955	Small size	95.97%	Small size	92.97%	92%
		Medium size	80.4%	Medium size	84.1%	
		Large size	93.3%	Large size	**100%**	
SVM TPSdataset 2 (200 × 5)	**0.003630**	Small size	95.2%	Small size	95.9%	93%
		Medium size	**86.9%**	Medium size	83.3%	
		Large size	93.3%	Large size	96.5%	

ANN: artificial neural network; SVM: support vector machine.

Table 6. Confusion matrices of three classifiers on TPSdataset1 and TPSdataset2.

Classifier	TPSdataset 1 (200 × 7)	TPSdataset 2 (200 × 5)
ANN	Predicted Value / Actual Value Small: 120 (60.0%), 20 (10.0%), 0 (0.0%) Medium: 4 (2.0%), 25 (12.5%), 1 (0.5%) Large: 0 (0.0%), 1 (0.5%), 29 (14.5%)	Predicted Value / Actual Value Small: 121 (60.5%), 5 (2.5%), 0 (0.0%) Medium: 3 (1.5%), 39 (19.5%), 0 (0.0%) Large: 0 (0.0%), 2 (1.0%), 30 (15%)
SVM	Predicted Value / Actual Value Small: 119 (59.5%), 9 (4.5%), 0 (0.0%) Medium: 5 (2.5%), 37 (18.5%), 2 (1.0%) Large: 0 (0.0%), 0 (0.0%), 28 (14%)	Predicted Value / Actual Value Small: 118 (59 %), 5 (2.5%), 0 (0.0%) Medium: 6 (3%), 40 (20%), 2 (1.0%) Large: 0 (0.0%), 1 (0.0%), 28 (14.0%)
AdaBoost	Predicted Value / Actual Value Small: 118 (59%), 7 (3.5%), 0 (0.0%) Medium: 6 (3%), 38 (19%), 2 (1%) Large: 0 (%), 1 (0.5 %), 28 (14%)	Predicted Value / Actual Value Small: 117 (59 %), 5 (2.5%), 0 (0.0%) Medium: 7% (3.5%), 39 (19%), 2 (1%) Large: 0 (0.0%), 2 (1%), 28 (14%)

ANN: artificial neural network; SVM: support vector machine.

times of each classifier for both data sets (with seven features and five features). Table 9 shows the confusion matrices of the three classifiers on the BIT-Vehicle data set.

Concluding remarks

The most important reasons not to be able to put real-time video processing applications into practice are scalability and performance problems. Processing of videos from traffic surveillance cameras is an example of such applications; in which video is processed for early warning or information is extracted for some real-time analysis. Since video data is a streaming media and gets very large in size, the central processing techniques are not efficient in their processing. It even gets worse in the case of real-time scenarios. The

Table 7. Examples of correctly classified and misclassified vehicles by different classifiers.

	Correct classification		Misclassification
ANN			
	Small size	Large size	Classified as small size
SVM			
	Medium size	Small size	Classified as medium size
AdaBoost			
	Small size	Medium size	Classified as large size

ANN: artificial neural network; SVM: support vector machine.

Table 8. Vehicle classification results on BIT-Vehicle data set with different sets of features and classifiers.

Classifier	Processing time (ms)	Precision		Recall		Accuracy
ANN BIT-Vehicle data set 1 (300 × 7)	1.978870	Small size	**99.0%**	Small size	86.8%	94.3%
		Medium size	85.0%	Medium size	97.3%	
		Large size	99.0%	Large size	100%	
ANN BIT-Vehicle data set 2 (300 × 5)	1.338385	Small size	99.0%	Small size	93.4%	97.0%
		Medium size	94.0%	Medium size	**97.9%**	
		Large size	98.0%	Large size	**100.0%**	
AdaBoost BIT-Vehicle data set 1 (300 × 7)	0.016991	Small size	99.0%	Small size	93.4%	97.0%
		Medium size	93.0%	Medium size	97.9%	
		Large size	**99.0%**	Large size	100%	
AdaBoost BIT-Vehicle data set 2 (300 × 5)	**0.013396**	Small size	99.0%	Small size	92.0%	96.3%
		Medium size	92.0%	Medium size	96.8%	
		Large size	98.0%	Large size	100%	
SVM BIT-Vehicle data set 1 (300 × 7)	0.017596	Small size	99.0%	Small size	92.5%	96.3%
		Medium size	92.0%	Medium size	96.8%	
		Large size	98.0%	Large size	100%	
SVM BIT-Vehicle data set 2 (300 × 5)	0.012535	Small size	99.0%	Small size	**95.2%**	**97.3%**
		Medium size	**95.0%**	Medium size	96.9%	
		Large size	98.0%	Large size	97.3%	

ANN: artificial neural network; SVM: support vector machine.

work presented in this article proposes a distributed real-time framework for vehicle detection and classification over the surveillance video streams. The effectiveness of the proposed system is tested on the real-world traffic surveillance camera, which is installed on a highway in Kocaeli city in Turkey. We see that the architecture is promising. It can be used in different areas such as infrastructure planning, struggling with traffic management, and prevention of traffic offenses.

Generally speaking, we provide the following contributions in this article:

Table 9. Confusion matrices of three classifiers on BIT-Vehicle data set.

Classifier	The BIT-Vehicle data set 1 (300 × 7)	The BIT-Vehicle data set 2 (300 × 5)
ANN	Predicted Value: Small / Medium / Large; Actual Value: Small (99, 33.0% / 15, 5.0% / 0, 0.0%); Medium (1, 0.3% / 85, 28.3% / 1, 0.3%); Large (0, 0.0% / 0, 0.0% / 99, 33.0%)	Predicted Value: Small / Medium / Large; Actual Value: Small (99, 33.0% / 6, 2.0% / 1, 0.3%); Medium (1, 0.3% / 94, 31.3% / 1, 0.3%); Large (0, 0.0% / 0, 0.0% / 98, 32.7%)
SVM	Predicted Value: Small / Medium / Large; Actual Value: Small (99, 33.0% / 8, 2.6% / 0, 0.0%); Medium (1, 0.3% / 92, 30.6% / 2, 0.6%); Large (0, 0.0% / 0, 0.0% / 98, 32.7%)	Predicted Value: Small / Medium / Large; Actual Value: Small (99, 33.0% / 5, 1.6% / 0, 0.0%); Medium (1, 0.3% / 95, 31.6% / 2, 0.6%); Large (0, 0.0% / 0, 0.0% / 98, 32.7%)
AdaBoost	Predicted Value: Small / Medium / Large; Actual Value: Small (99, 33.0% / 7, 2.3% / 0, 0.0%); Medium (1, 0.3% / 93, 31.0% / 1, 0.3%); Large (0, 0.0% / 0, 0.0% / 99, 33.0%)	Predicted Value: Small / Medium / Large; Actual Value: Small (99, 33.0% / 8, 2.6% / 0, 0.0%); Medium (1%, 0.3% / 92, 30.6% / 2, 0.6%); Large (0, 0.0% / 0, 0.0% / 98, 32.7%)

ANN: artificial neural network; SVM: support vector machine.

- We propose a general framework for real-time topic-based pub/sub systems.
- We propose a distributed real-time framework for vehicle detection and classification over the surveillance video streams.
- We extensively evaluate the efficiency and efficacy of well-known classification methods in terms of precision, recall, and accuracy.

End users, or operators, might be interested in visions of some objects captured by surveillance camera. In the proposed architecture, visions are provided as still text images. In the future, we will be filtering real-time video streaming and creating sub-streams for the subscribers. The further processing might be done by the operator itself. Operators even refilter and republish the filtered streams in a cascading manner. We will call this architecture as hierarchical topic-based pub/sub messaging middleware for real-time traffic surveillance.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work has been supported by Kocaeli University Scientific Research and Development Support Program (BAP) in Turkey under project 2016/004.

References

1. Ying-Nong C, Chin-Chuan H, Gang-Feng H, et al. Facial/License plate detection using a two-level cascade classifier and a single convolutional feature map. *Int J Adv Robotic Syst* 2015; 12(12): 1–16.
2. Bas E, Tekalp AM, and Salman S. Automatic vehicle counting from video for traffic flow analysis. In: *Proceedings of IEEE intelligent vehicles symposium*, Istanbul, Turkey, 13–15 June 2007, pp. 392–397. IEEE.
3. Al-Sobkya AA and Mousa RM. Traffic density determination and its applications using smartphone. *Alex Eng J* 2016; 55(1): 513–523.
4. Pelegri J, Alberola J, and Llario V. Vehicle detection and car speed monitoring system using GMR magnetic sensors. In: *Proceedings of IEEE annual conference in the industrial electronics society*, Sevilla, Spain, 5–8 November 2002, pp. 1693–1695. IEEE.
5. Ryusuke K, Shigeyuki K, Tomoyuki N, et al. A punctilious detection method for measuring vehicles' speeds. In: *Proceedings of the international symposium on intelligent signal processing and communication system*, Tottori, Japan, 12–15 December 2006, pp. 967–970. IEEE.
6. Harry HC, Benjamin DS, Joe P, et al. Development and field test of a laser-based nonintrusive detection system for identification of vehicles on the highway. *IEEE Trans Intell Trans Syst* 2005; 6(2): 147–155.
7. Marikhu R, Moonrinta J, Ekpanyapong M, et al. Police eyes: real world automated detection of traffic violations. In: *Proceedings of ECTI-CON*, Krabi, Thailand, 15–17 May 2013.

8. Peng Y, Jin JS, Luo S, et al. Vehicle type classification using PCA with self-clustering. In: *Proceedings of IEEE international conference on multimedia expo workshops*, Melburne, Australia, 9–13 July 2012, pp. 384–389. IEEE.
9. Messelodi S, Modena CM, and Zanin M. A computer vision system for the detection and classification of vehicles at urban road intersections. *Pattern Anal Appl*, August 2005; 8(1-2): 17–31.
10. Buch N, Orwell J and Velastin SA. Detection and classification of vehicles for urban traffic scenes. In: *Proceedings of 5th international conference on visual information engineering*, Xian China, China, 29 July–1 August 2008, pp. 182–187. IET.
11. Morris B and Trivedi M. Robust classification and tracking of vehicles in traffic video streams. In: *Proceedings of IEEE Intelligent Transportation Systems Conference*, Toronto, Ont., Canada, 17–20 September 2006, pp.1078–1083. IEEE.
12. Dallalzadeh E, Guru DS, Manjunath S, et al. Classification of moving vehicles in traffic videos. In: Meghanathan N, Chaki N and Nagamalai D (eds) *Advances in computer science and information technology. Computer science and engineering*. Vol 85. Berlin, Heidelberg: Springer, 2012, pp. 211–221.
13. İnce E. Measuring traffic flow and classifying vehicle types: a surveillance video based approach. *Turk J Elec Eng Comp Sci* 2011; 19(4): 607–620.
14. Chen Z and Ellis T. Multi-shape descriptor vehicle classification for urban traffic. In: *International conference on digital image computing: techniques and applications*, Noosa, QLD, Australia, 6–8 December 2011, pp. 456–461.
15. Zhang G, Avery RP and Wang Y. A video-based vehicle detection and classification system for real-time traffic data collection using uncalibrated video cameras. *J Transp Res Board* 2007; 1993: 138–147.
16. Zhang L, Li SZ, Yuan X, et al. Real-time object classification in video surveillance based on appearance learning. In: *Proceedings of IEEE conference on computer vision and pattern recognition*, MN, USA, 17–22 June 2007, pp. 1–8. IEEE.
17. Morris B and Trivedi MM. Learning, modelling and classification of vehicle track patterns from live video. *IEEE Trans Intell Transp Syst* 2008; 9(3): 425–437.
18. Moussa GS. Vehicle type classification with geometric and appearance attributes. *Int J Civil Struct Construction Archit Eng* 2014; 8(3): 273–278.
19. Kul S, Eken S and Sayar A. Measuring efficiencies of vehicle classification algorithms on traffic surveillance video. In: *Proceedings of international artificial intelligence and data processing symposium*, Malatya, Turkey, 17–18 September 2016.
20. Zou Y, Cao J and Wu H. TraffiCast: real-time pub/sub based video surveillance system over interconnected WMNs and WSNs. In: *Proceedings of 6th IEEE international conference on distributed computing in sensor systems workshops*, Santa Barbara, USA, 21–23 June 2010.
21. Jiannong C, Xuefeng L, Steven L, et al. A ubiquitous wireless video surveillance system based on pub/sub. In: *Proceedings of the 13th international conference on ubiquitous computing*, Beijing, China, 17–21 September 2011, pp. 611–612. ACM.
22. Zou Y, Cao J and Zhang J. Pub-eye: the delay constrained Pub/Sub for large scale wireless video surveillance. In: *Proceedings of wireless communications and networking conference*, Cancun, Mexico, 28–31 March 2011.
23. Lauri Pesonen IW, David ME, and Bacon J. A capability-based access control architecture for multi-domain publish/subscribe systems. University of Cambridge, 2008, Technical Report. Available from: http://www.cl.cam.ac.uk/techreports/ (2008, accessed 10 January 2017).
24. Bouwmans T. Background subtraction for visual surveillance: a fuzzy approach. In: Pal S, Petrosino A and Maddalena L, Taylor and Francis Group. (eds) *Handbook on soft computing for video surveillance*. Chapter 5. Taylor and Fransic Group. 2012.
25. Sobral A. BGSLibrary: an opencv c++ background subtraction library. In: *Proceedings of IX Workshop de Viso Computacional*, Rio de Janerio, Brazil, 2013; CRC Press. Software available from: http://code.google.com/p/bgslibrary.
26. Avery RP, Wang Y, and Rutherford GS. Length-based vehicle classification using images from uncalibrated video cameras. In: *Proceedings of the 7th international IEEE conference on intelligent transportation system*, Washington, USA, 3–6 October 2004, pp.737–742. IEEE.
27. Samet H. *Foundations of multidimensional and metric data structures*. 1st ed. Morgan Kaufmann. 2006.
28. Ding C, He X, Zha H, et al. Adaptive dimension reduction for clustering high dimensional data. In: *Proceedings of the international conference on data mining*, Washington DC, USA, 9–12 December 2002. IEEE.
29. Han J and Kamber M. *Data mining concepts and techniques*. 3rd ed. Morgan Kauffmann Publishers Inc., 2011.
30. Jović A, Brkić K, and Bogunović N. A review of feature selection methods with applications. In: *MIPRO*, Opatija, Cratra, 25–29 May 2015, IEEE.
31. Hall M and Holmes G. Benchmarking attribute selection techniques for discrete class mining. *IEEE Trans. Knowl. Data Eng*. 2003; 15(16): 1437–1447.
32. Berkhin P. *Survey of clustering data mining techniques*. San Jose: Accrue Software Inc., 2002.
33. Richards M, Monson-Haefel R and Chappell DA. *Java message service*. 2nd ed. O'Reilly Media. 2009.
34. Zhen D, Yuwei W, Mingtao P, et al. Vehicle type classification using a semisupervised convolutional neural network. *IEEE Trans. Intell. Transp. Syst*. 16(4): 2247–2256.