Evolutionary Intelligence (2020) 13:83–91
https://doi.org/10.1007/s12065-018-0167-z

SPECIAL ISSUE

Performance evaluation of support vector machine and convolutional neural network algorithms in real-time vehicle type and color classification

Ali Şentaş[1] · İsabek Tashiev[1] · Fatmanur Küçükayvaz[1] · Seda Kul[1] · Süleyman Eken[1] · Ahmet Sayar[1] · Yaşar Becerikli[1]

Received: 30 May 2018 / Revised: 12 August 2018 / Accepted: 16 August 2018 / Published online: 28 August 2018

Abstract
In order for traffic management and information systems to provide proper traffic flow, it is necessary to obtain information about traffic with the help of various sensors. In this context, in recent years the use of video cameras in traffic observation and control has become very widespread and actively used. Numerous studies such as license plate recognition, vehicle number finding, traffic intensity determination, vehicle speed calculation, band violation and vehicle classification can be done with the help of video processing based video monitoring systems. Traffic surveillance videos are very actively used for this purpose. In this paper, we have developed a system that classifies vehicles according to their type. Firstly we create a vehicle dataset from an uncalibrated camera. Then, we test Tiny-YOLO real-time object detection and classification system and support vector machine (SVM) classifier model on our dataset and well-known public BIT-Vehicle dataset in terms of recall, precision, and intersection over union performance metrics. Experimental results show that two methods can be used to classify real-time streaming traffic video data.

Keywords Vehicle detection and classification · Video processing · Tiny-YOLO · Intelligent traffic management systems

1 Introduction

Vehicle type recognition is an important research topic during the last decades. It has a wide range of applications such as automatic vehicle identification [1], road capacity [2], traffic density measurement [3], speed detection [4–6], and traffic violation detection [7]. It is also important to identify vehicle categories for crime prevention and transportation investors [8].

Nowadays, information about vehicles is obtained from sensors and cameras that have been placed on the road. The advances in the technologies image processing and pattern recognition can be used to classify different vehicle types. However, the main reason to be unable to put real-time image processing applications into practice is scalability and performance problems for the computer. Processing of videos from traffic surveillance cameras is an example of such applications, in which video is processed for early warning or extracting information through some real-time analysis. Video data can get very large in size, thus, using traditional processing techniques to get the information in the right time (also can be real-tine) and correctly is not an easy task. It even gets worse in the case

Initial of this paper was presented at EIDWT 2018, Albania.

✉ Seda Kul
seda.kul@kocaeli.edu.tr

Ali Şentaş
alisentas96@gmail.com

İsabek Tashiev
isabek.tashiev@gmail.com

Fatmanur Küçükayvaz
fatmanur.kucukayvaz@gmail.com

Süleyman Eken
suleyman.eken@kocaeli.edu.tr

Ahmet Sayar
ahmet.sayar@kocaeli.edu.tr

Yaşar Becerikli
ybecerikli@kocaeli.edu.tr

[1] Computer Engineering Department, Kocaeli University, 41380 Izmit, Turkey

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 the right time.

There are many vehicle classification methods, which can be divided into hardware based and software based in general. Vehicle-based classification methods are implemented on different systems such as radar, infrared detectors, microwave method. Radar technology is widely used in speed measurement and vehicle detection, but studies on vehicle classification in particular are too small to be analyzed correctly. However, Fang and his team achieved a 95% success rate in the system using continuous waves (CW) without modulation [9]. The disadvantages of hardware-based systems are: the size of the detectors is large and difficult to manage, the installation is expensive and the damage to the installed path, the detectors are expensive and the information obtained is limited. Software-based recognition systems have many advantages over hardware-based systems, they can be installed easily and do not harm the environment while being monitored. The disadvantages of software tool classification methods are that they provide much better results than hardware-based systems, despite software problems such as slow processing and low real-time processing capacities [10]. Many studies have been carried out on software-based vehicle detection and classification. In general, there are two steps that these studies follow, background subtraction and vehicle segmentation from the background. General problems in vehicle classification and detection are addressed with the problems of how to extract the features that will represent the vehicle how to remove the foreground with dynamic background or how to deal with bad weather conditions. This paper focuses on solving such problems and how to perform vehicle detection in software-based systems.

Advancement in real-time image processing over the Internet and the technology of deep learning attract attention in many application areas such as intelligent traffic/vehicle management systems, transferring traffic surveillance video data. The work presented in this paper is summarized as a real-time vehicle detection and classification over the video streams provided by traffic surveillance cameras. In our work, we used two different methods to classify vehicles. One of them is support vector machine (SVM) and the other is the You Only Look Once (YOLO). We apply the state-of-the-art, real-time object detection system Tiny-YOLO [8], which have proven a good competitor to Fast R-CNN's and SSDs both in terms of detections and speed. In this paper, we will apply the Tiny-YOLO and SVM classifier on own dataset (TPSdataset) and well-known public dataset BIT-Vehicle. The contributions of this paper are as follows:

- Training and applying the state-of-the-art, real-time object detection system Tiny-YOLO for vehicle type detection and classification.
- Filtering real time streaming video data according to vehicle types
- End users (which are searching for a vehicle) have capability of searching for a vehicle defined in type property.
- Building TPSdataset
- Vehicle color recognition

The remainder of this article is organized as follows. Section 2 presents the related work. The main architecture for vehicle detection, classification and color recognition are given in "Vehicle detection, classification and color recognition" section. The performance results and their analyses are given in "Experimental analysis" section. The last section concludes the article.

2 Related works

Object detection and classification the most well-known and most challenging problem in computer vision. It aims to divide one image into many different categories. Therefore, vehicle detection and classification is also a challenging problem [10], mostly because there are lots of different vehicle types and dimensions. Also, the most common problems with object detection and classification based on image processing are variable environments, uncontrollable weather conditions, and the possibility of damage to the camera and trying to solve these problems in real time makes it even more difficult.

There have been many different types of approaches for object detection and three commonly used methods have been focused on. Frame differencing, one of the most popular object detection methods, is based on the calculation of the difference between two consecutive images. The application of the algorithm is simple and easy, and there is also a rapid adaptive structure to changing natural conditions. However, the working accuracy of the Frame differencing method depends on the frame rate and the speed of the object. In this the case, it is inevitable that the object to be detected cannot be found in the state of stopping or slowing down [11]. The second popular method, Optical flow method aims to calculate the optical flow distribution characteristics of the image [12]. This method works better, however, requires a large quantity of calculation. Also due to its sensitivity to noise, it is not ideal for real-time applications. The last popular method we mentioned about is background subtraction [13]. The background subtraction algorithm is based on the background modeling method. BGS tries to create an average model over the past frames and should be as sensitive as it can to detect moving objects. The created model is called the reference frame. The difference between the reference frame and the current frame is the presence of moving objects. The BGS algorithm is basically divided into the recursive and non-recursive algorithm. The main difference between them is the buffer structure. The recursive BGS model does not contain buffers so that the reference frame is highly dependent on the background, and any error can remain in the reference frame for an extended period of time. In the non-recursive method, there is a buffer of size N, and the history of the reference model does not exceed N. Given the heavy traffic and slow traffic conditions, large buffer sizes will be better for the solution.

There are many different approaches to object classification methods. Shape-based classification contains a description of the shape information of the motion regions such as point, box and blob representations for classification. In each frame, classification is done at each blob and the results are kept on the histogram [14]. Unlike many other features, color is relatively stable under viewing angle changes and is easy to achieve. Color feature is not always appropriate as a means of detecting and monitoring but lower costs of computational algorithms make the approach desirable. Texture-based technique [15] estimates the occurrence of gradient orientations in localized portions of the image on a dense grid of equally spaced cells.

In the literature, the most commonly used object recognition approaches are Haar-like cascade classifier and histogram of oriented gradients (HOG) attribute descriptors with support vector machine classifier. These two common approaches these two approaches provide good performance and real-time performance [16, 17].

The combination of HAAR and SVM algorithm was first shown by Viola et al. [18]. It was used for human face recognition. Thus, it was mainly used for face recognition systems, but there were also studies targeting different objects such as Han et al. [19]. Feature extraction from an object can also be done with the HOG algorithm by Dalal et al. [20], one of the algorithms used in this paper. Lowe introduced the scale invariant feature transform (SIFT) algorithm for classification and can be used for feature extraction but it's unsuitable for real-time applications. To overcome the time problem in the SIFT algorithm, speeded up robust features (SURF) introduced by Bay et al. [21]. It gives similar results to the SIFT algorithm but it is a more successful algorithm in terms of time. Ojala et al. [22] introduced local binary patterns (LBP) approach. The most paper that uses LBP cascade classifiers for face recognition. LBP algorithm also used with HOG to increase detection performance by Wang [23]. Wang, extend his work to gender recognition. Principal component analysis (PCA) used for feature extraction [24] and the feature set classified with SVM [25] or AdaBoost [26]. The methods have been described are for general literature view for object detection and classification/recognition. In this paper, there are four main stages, detection, classification, tracking, and vehicle color classification. Vehicle detection methodologies, most of them [27, 28] assumes the camera, where the image is obtained, is static. Background extraction algorithm is a popular method for vehicle detection as well as object detection in general. Lu et al. [29] proposed a system uses fuzzy background subtraction to detect vehicles. One of the main disadvantages of the background subtraction algorithm is that it's not suitable for static environments. In one of our previous work [30] shape-based binary features (the width of the rectangle surrounding the vehicle width, height, and minor /major axis length, and etc.) used for classification vehicles by their size. Three classification algorithms (SVM, Adaboost, and ANN) were performed with 87.5, 81.6, and 85.4 achieved accuracies. Also another one of our previous work [31], Kul et al. introduced a middleware system based on pub/sub messaging system in real time. Vehicles were classified according to their dimensions, and the results were sent to users who subscribe to the corresponding topic in each class. In this work, we did not choose the background subtraction algorithm mostly because it cannot cope with dynamic environmental conditions.

Along with improvements in image processing and hardware, it is not surprising that deep learning techniques have shown good results in object recognition and are much better than other known methods. Deep learning algorithms are state of the art in object recognition so far. A vehicle type recognition system proposed by Dong et al. [32], a half-supervised convolutional neural network was used and trained with front view images of the vehicles. Dong et al. also created a dataset named BIT-Vehicle data set. It consists of 9850 high-resolution images with only the front views of the vehicles. They achieved 96.1% accuracy in daytime and 89.4% accuracy in the nighttime. There are also another works that used deep learning algorithm for vehicle detection system. Faster R-CNN proposed for vehicle type classification by Wang et al. [33]. They classify car and truck, their system achieved 90% accuracy with the system NVIDIA Jetson TK1 board with 192 CUDA cores. It takes around 0.3 s to detect a vehicle, this also shows that this work suitable for real-time systems. The other deep learning method which is proposed by Gao et al. [34] is slightly different from others. It combines frame difference method with CNN. The results come from frame differencing are the binary images used to detect the cars. Gao et al. achieved 88% accuracy. Lee et al. [35] proposed K local expert networks for vehicle recognition. They used, ResNet, GoogLeNet, and AlexNet CNN, they choose The MIO-TCD dataset. The accuracy of their work achieved is 97.92%. Huo et al. [36] proposed model vehicle classification captured in the real-time surveillance system. Besides the works done by taking only the front view, they aimed to use all angles (front, side, and rear). Their work achieved 83% accuracy.

Kim et al. [37] proposed a vehicle type classification system and their system achieved 97.84% accuracy. Zhou et al. [38] developed a vehicle detection and classification system using Deep Neural Networks (DNNs) approach. They have chosen to use the YOLO for vehicle detection. Their dataset consists of two classes, passenger and other. Passenger vehicle class includes sedans, SUV, and MPV, other vehicle class includes van, truck, and other types of vehicle. They have analyzed DNN approaches for detection and classification. The precision and recall values reached by their work are 93.3 and 83.3% for vehicle detection.

Vehicle information is an important key to intelligent transportation systems. The main problem for vehicle color detection is to select the most suitable region of interest (ROI) for classification its color. Wang et al. [39] proposed the advantages of multiple support vector machines (SVM) for feature selection in the classification of object color. Zheng et al. [40] proposed a low-cost system based on SVM for classifying objects.

3 Vehicle detection, classification and color recognition

In this section, we first explained the vehicle detection system and describe the detection algorithms that we used. Then, the vehicle classification method will be explained. Two different methods have been used for classification. In the following sections, the most appropriate method is determined by comparing the results of these two different methods.

3.1 Vehicle detection

Vehicle detection was performed before classifying the vehicles and SVM classifier was used for this purpose. SVM consist of a set of learning methods used for classification and regression. Unlike many classifications, SVM aims to find the best separation line for distinguishing data belonging to different classes. A feature identifier is an information that represents and simplifies the image by extracting useful information and discarding extraneous. There are several feature acquisition algorithms, in this paper features used in SVM classification were obtained by the Histogram of Oriented Gradients algorithm. The main purpose of the HOG method is to define the image as a group of local histograms. These groups are the histograms in which the magnitudes of the gradients are collected in the orientations of the gradients in a local region of imagen. HOG is a descriptor and uses a gradient vector orientation histogram and SVM is classier with good generalization uses the features extracted by the HOG algorithm. HOG feature descriptor is a representation of an object that simplifies images by extracting useful information and throwing unnecessary information. The main aim of the HOG algorithm is trying to define the object in the image as a group of local histograms. The features of the vehicle in the images have been obtained with the HOG algorithm. The HOG property extraction is a rectangle. The frames divided into 8×8 cells and HOG detector uses a sliding window which is moved throughout the image. At each position of the detector window, a HOG descriptors compute with the image and try to find if there is an edge through that blob, and how visible is this edge. HOG descriptors are obtained from positive (vehicle) and negative (non-vehicle) images. After extracting HOG features, linear SVM is trained with those datasets. In the following section SVM is also used for vehicle type classification.

3.2 Vehicle classification

In this paper two different approaches used for vehicle type classification. HOG + SVM and YOLO. The second usage of SVM is to classifying vehicles. The steps taken during the detection process were also carried out here. For each class of vehicles, the HOG features have been removed and the SVM model has been trained. The second usage of SVM is to classifying vehicles. The steps taken during the detection process were also carried out here. For each class of vehicles, the HOG features have been removed and the SVM model has been trained. The images in the data set we use for the HOG property extraction are square. The image is divided into 8×8 cells and The HOG detector uses a sliding detection window which is moved around the image.

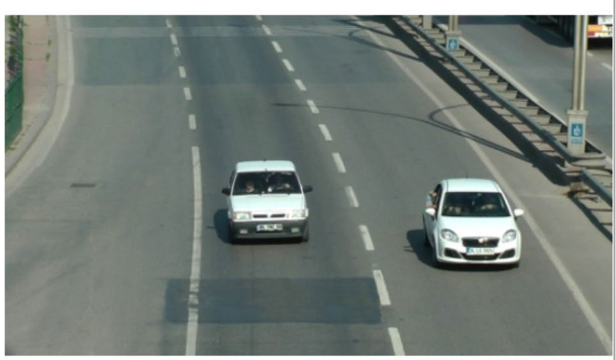
Convolutional neural networks (CNN) are the key players of object classification and detection tasks in nowadays. However, one of the main reasons why convolutional neural networks were not commonly used in real-world applications was that they require powerful computational resources and with the significant improvements of GPU boosting technologies. In the last few years, there were developed a lot of variations of convolutional neural networks like R-CNN and its modifications Fast R-CNN and Faster R-CNN. Each of them improved the previous one on especially important criteria like speed and accuracy of the classification. One of the advantages of convolutional networks is that their can-do object classification and detection simultaneously. YOLO concurrently performs feature extraction, bounding box prediction, nonmaximal suppression, and contextual reasoning operations. Thanks to feature extraction step, one of the advantages of YOLO, there is no need to search to a different algorithm for feature extraction. YOLO can also be applied to the entire image. But in order to save time in our work, firstly the vehicle is detected and then the image of the car is classified by YOLO. The main criteria of object classification and detection are fast, accurate and able to recognize a variety of objects.

CNN consists of these main layers like convolutional layer, pooling layer, and fully connected layer. Depends on architecture CNNs use these layers in different variations. In our experiments, we used Tiny-YOLO, a state-of-the-art and real-time object classification and detection architecture. It has a simpler model architecture and it needs small GPU resources appropriately. It uses a model which named Darknet-19 and consists 9 convolutional layers, 6 max-pooling layers, one average pooling layer and the last one is the softmax layer.

3.3 Vehicle color recognition

Vehicle color identification plays an important role in information retrieval. The challenge of this method is to find the most suitable region of interest (ROI) part. SVM algorithm was used for color determination. The front view of the vehicle selected for ROI. We collected colors in seven different files: white, black, blue, red, gray, yellow, green. First, the image is converted from BGR format to Lab format. Contrast limited adaptive histogram equalization (CLAHE) filter is applied to improve the visual control over the image. After filtering, the image is converted back to BGR format. Then, the image is converted to HSV format. The calculated 3D histogram is resized as a one-dimensional vector. The vector has zero mean and unit variance. A histogram is a graph showing the number of each color value in a numerical image. From this chart, you can have information about the state of the brightness or tones. After these steps, the SVM model was trained with the data we obtained. Figure 1, shown examples of vehicle color recognition also detailed information is given about the vehicles in the xml file shown in Fig. 2.

Fig. 1 Example from vehicle color




```
<?xml version="1.0" ?>
<annotation>
    <filename>image-0000158.jpg</filename>
    <size>
        <width>1920</width>
        <height>1080</height>
    </size>
    <object>
        <name>auto</name>
        <bndbox>
            <xmin>681</xmin>
            <ymin>453</ymin>
            <xmax>979</xmax>
            <ymax>728</ymax>
        </bndbox>
    </object>
    <object>
        <name>auto</name>
        <bndbox>
            <xmin>1308</xmin>
            <ymin>512</ymin>
            <xmax>1645</xmax>
            <ymax>810</ymax>
        </bndbox>
    </object>
</annotation>
```


Fig. 2 An example from TPS-dataset and its XML file

4 Experimental analysis

4.1 Building TPSdataset

In addition to the publicly available dataset, we also create our own dataset. TPSdataset consists three video files shot in Kocaeli, Turkey. 4944 frames obtained from videos. The task of labeling the dataset is an important process for this "LabelImg" graphical image annotation tool used [41]. Using this tool, the vehicles in the picture are enclosed in the rectangular area. Annotations are saved as XML files in PASCAL VOC format, the format used by ImageNet. The XML file contains the coordinates, width, height, and class label of the vehicles in that image (see Fig. 1 for one frame and its XML file).

There are five classes in our dataset: bus, minivan, minibus, truck, auto. The auto class is a broad class containing sedans, SUVs and hatchback vehicles. The truck contains trucks, long vehicles and smaller trucks which are called minivans in the BIT-Vehicle dataset. Minibusses are like minivans but bigger and have more space between the roof and front window. Also in our dataset, all samples are taken with daylight. Table 1 shows the distribution of vehicle types in TPSdataset.

4.2 Experimental setup and performance metrics

In our experiments, we have used an Intel Core i5-4210U 1.7 GHz processor and Arch Linux OS. We test the system both on TPSdataset and another data set named BIT-Vehicle. Classification result is evaluated using recall and precision parameters as the performance measures.

4.3 Tests on BIT-Vehicle dataset

BIT-Vehicle dataset [42] is a well-known dataset created by Dong et al. It consists 9850 vehicle images, whose sizes are of 1600 × 1200 and 1920 × 1080 and captured from two cameras at the different time and places. There are six classes in this dataset: bus, microbus, minivan, SUV, sedan, truck. Table 2 shows that this dataset does not perform well with the HOG algorithm. The main reason for that because HOG requires samples to be aligned similarly, however cars alignment variation very differently from one sample to another. SVM used with a parameter of C set to 0.3 and the image size for HOG descriptor is 128 × 128 pixels, block size is 16 × 16, block stride is 8 × 8 and cell size is 8 × 8 pixels. The dataset is divided into two, 75% for training and 15% for testing. Table 2 shows the performance results of HOG + SVM classifier on BIT-Vehicle dataset. It's noticeable that precision and recall of minivan is lower than other classes. The reason for this irregularity is that our trained model confuses minivans with trucks. 22 out of all 118 minivan test sample are misclassified as trucks. When you see all the vehicles from the same square window minivan and trucks does look alike. Also, minivan samples are less than others in BIT-Vehicle dataset.

You Only Look Once (YOLO) used for classification and detection objects using bounding boxes. Intersection over Union (IOU) metric is an evaluation metric used to measure the accuracy of our model on the test dataset. In order to apply IOU for evaluation of our model we need: the ground-truth hand labeled bounding boxes which specify where in our object is and the predicted bounding boxes from our model. Table 3 shows the performance results of Tiny-YOLO performance on BIT-Vehicle dataset.

4.4 Tests on TPSdataset

We use the same parameters for training SVM classifier to get the best results. Table 4 shows the performance results of the SVM classifier on TPSdataset.

Table 1 Distribution of vehicle types

Type	#
Auto	719
Bus	147
Minibus	72
Minivan	187
Truck	266

Table 2 HOG + SVM performance on BIT-Vehicle dataset

Vehicle class	Sample count	Precision	Recall
Bus	555 (415/138)	98.5507	98.5507
Microbus	878 (659/219)	91.7431	91.3242
Minivan	474 (354/118)	87.7358	78.8136
SUV	1381 (1032/343)	89.7898	87.172
Sedan	5796 (4310/1436)	97.5762	98.1198
Truck	821 (615/205)	90	96.5854
Average	9905 (7385/2459)	92.5659	91.7609

Table 3 Tiny-YOLO performance on BIT-Vehicle dataset

Vehicle class	Sample count	Precision	Recall	IOU
Bus	558 (446/112)	100	100	90.03
Microbus	883 (706/177)	96.67	98.31	86.42
Minivan	476 (381/95)	98.95	98.95	85.51
SUV	1392 (1114/278)	97.19	99.64	89.59
Sedan	5921 (4737/1184)	97.20	99.75	90.13
Truck	823i (658/165)	87.41	100	100
Average	10,053 (8042/2011)	97.90	99.60	89.29

Table 4 Vehicle classification with HOG and SVM results on TPS-dataset

Vehicle class	Sample count	Precision	Recall
Bus	147 (111/36)	100	100
Minivan	187 (141/46)	91.8367	97.8261
Minibus	72 (54/18)	94.4444	94.4444
Truck	266 (200/66)	100	98.4848
Auto	719 (540/179)	99.435	98.324
Average	1391 (1046/345)	97.1432	97.8159

4 out of all 46 test samples for minivan are misclassified, 3 classified as auto and 1 as truck hence precision is low for the minivan but this problem would be fixed by expanding the dataset which we aim to do so in the future. We test model with our test videos and it performed better at detecting and classifying cars than the model trained with BIT-Vehicle dataset. It's small compared to BIT-Vehicle but it's more suited to HOG approach because vehicles are aligned to each other properly and we aim to expand it and add night images.

As the results shown in Table 5, Tiny-YOLO demonstrates poor performance on TPSdataset compared to the model which was trained on BIT-Vehicle Dataset. The result of this behavior is the lack of data.

Another accuracy metric for evaluating models is the precision–recall curve (PR Curve). To find out the trade-off between precision and recall we calculate precision and recall values for threshold from 0.0 to 1.0 with step 0.1 and find the suitable threshold.

4.5 Tests on vehicle color recognition

The dataset published by Chen et al. [43] was used for the color classification of vehicles. They collected their data sets on urban roads, it contains 15,601 frames with the frontal view of vehicles and with the resolution of 1920 × 1080. Due to the noise caused by haze, illumination variation and overexposure the data set is challenging. Table 6 shows the performance results of the SVM classifier on TPSdataset.

Table 5 Tiny-YOLO performance on TPSdataset

Vehicle class	Sample count	Precision	Recall	IOU
Bus	568 (483/85)	40.00	75.29	58.96
Minivan	722 (639/83)	48.98	86.75	67.59
Minibus	297 (237/60)	36.27	61.67	49.63
Truck	540 (451/89)	44.74	76.40	61.52
Auto	3069 (2588/481)	52.53	92.93	76.05
Average	5196 (4398/798)	62.83	86.22	69.74

Table 6 Vehicle classification with HOG and SVM results on TPS-dataset

Color	Train	Validation	Accuracy (%)
Black	2408	1034	55.89
Blue	759	327	85.32
Gray	2134	912	36.4
Green	338	144	62.28
Red	1359	582	96.73
White	3319	1423	86.43
Yellow	407	174	89.65

Due to the illumination variation black, green and gray do not perform well. In the future, we will try to overcome this issue by expanding the dataset.

5 Conclusion

Image processing and deep learning based video surveillance system in traffic management systems enable many studies such as license plate recognition, finding the number of vehicles, traffic density detection, vehicle speed calculation, detection of lane violations and vehicle classification. In this study, we firstly create own video surveillance dataset. Then, SVM classifier and Tiny-YOLO are tested on TPS-dataset and well-known public dataset BIT-Vehicle in terms of recall and precision metrics. Also, we give IOU metric for Tiny-YOLO. Experimental results show that Tiny-YOLO outperforms SVM on BIT-Vehicle dataset. But, SVM is well than Tiny-YOLO on TPSdataset, because the acquired dataset is unbalanced. This generally causes overfitting, which may affect the final results.

In the future, instead of classical approaches of centralized computation, a distributed scalable network of collaborating computation nodes is going to be developed to process streaming real-time video data coming from traffic surveillance cameras. In this way, hierarchical topic-based publish-subscribe messaging middleware is going to be realized. Also, real-time stream processing infrastructures such as Apache Kafka, Flink, and Storm will be considered.

Acknowledgements This work has been supported by the TUBITAK under Grant 116E202.

References

1. Ying-Nong C, Chin-Chuan H, Gang-Feng H, Kuo-Chin F (2015) Facial/license plate detection using a two-level cascade classifier and a single convolutional feature map. Int J Adv Robot Syst 12(12):1–16

2. Bas E, Tekalp AM, Salman S (2007) Automatic vehicle counting from video for traffic flow analysis. In: Proceedings of IEEE intelligent vehicles symposium. IEEE Pres, Istanbul, pp. 392–397
3. Al-Sobkya AA, Mousa RM (2016) Traffic density determination and its applications using smartphone. Alex Eng J 55(1):513–523
4. Pelegri J, Alberola J, Llario V (2002) Vehicle detection and car speed monitoring system using GMR magnetic sensors. In: Proceedings of IEEE annual conference in the industrial electronics society, IEEE, Sevilla, pp. 1693–1695
5. Ryusuke K, Shigeyuki K, Tomoyuki N, Takashi A, Makoto A, Hisao O (2006) A punctilious detection method for measuring vehicles' speeds. In: Proceedings of the international symposium on intelligent signal processing and communication system, IEEE, Tottori, pp. 967–970
6. Harry HC, Benjamin DS, Joe P, Bin L, Bo C, Zhaoqing W (2005) Development and field test of a laser-based nonintrusive detection system for identification of vehicles on the highway. IEEE Trans Intell Trans Syst 6(2):147–155
7. Marikhu R, Moonrinta J, Ekpanyapong M, Dailey MN, Siddhichai S (2013) Police eyes: real world automated detection of traffic violations. In: Proceedings of ECTI-CON, IEEE, Krabi, pp. 1–6
8. Şentaş A et al (2018) Performance evaluation of support vector machine and convolutional neural network algorithms in real-time vehicle type classification. In: Barolli L, Xhafa F, Javaid N, Spaho E, Kolici V (eds) Advances in internet, data & web technologies. EIDWT 2018, lecture notes on data engineering and communications technologies, vol 17. Springer, Cham
9. Redmon J, Farhadi A (2017) YOLO9000: better, faster, stronger. In: Proceedings of conference on computer vision and pattern recognition, IEEE, Honolulu
10. Kul S, Eken S, Sayar A (2017) A concise review on vehicle detection and classification. In: Proceedings of the third international workshop on data analytics and emerging services, IEEE, Antalya, pp. 1–4
11. Rakibe RS, Patil BD (2013) Background subtraction algorithm based human motion detection. Int J Sci Res Publ 3:2250–3153
12. Liu T, Chi H, Hong C, Zhao M (2014) Moving object detection in dynamic background. In: Proceedings of the 33rd Chinese control conference, Nanjing, pp 4914–4919
13. Athanesious JJ, Suresh P (2012) Systematic survey on object tracking methods in video. Int J Adv Res Comput Eng Technol (IJARCET) 1:242–247
14. Patel HA, Thakore DG (2013) Moving object tracking using Kalman filter. Int J Comput Sci Mobile Comput 2:326–332
15. Lee J-Y, Yu W (2011) Visual tracking by partition-based histogram backprojection and maximum support criteria. In: 20 IEEE international conference on robotics and biomimetics (ROBIO), 7–11 Dec. 2011, pp. 2860, 2865
16. Gaszczaka A, Breckon T, Hana J (2011) Real-time people and vehicle detection from UAV imagery. In: SPIE conference intelligent robots and computer vision
17. Viola P, Jones M (2004) Robust real-time face detection. Int J Comput Vision 57:137–154
18. Viola P, Jones M (2001) Rapid object detection using a boosted cascade of simple features. In: Proceedings of computer vision and pattern recognition, IEEE, Kauai, pp. 511–518
19. Han S, Han Y, Hahn H (2009) Vehicle detection method using Haar-like feature on real time system. World Acad Sci Eng Technol 59(35):455–459
20. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: Proceedings of IEEE computer society conference on computer vision and pattern recognition, IEEE, San Diego, pp. 886–893
21. Bay H, Tuytelaars T, Van Gool L (2006) Surf: speeded up robust features. Proc Comput Vis ECCV 110:404–417
22. Ojala T, Pietikäinen M, Harwood D (1994) Performance evaluation of texture measures with classification based on Kullback discrimination of distributions. In: Proceedings of the 12th IAPR international conference on pattern recognition, IEEE, Jerusalem, pp. 58–62
23. Wang X, Han TX, Yan S (2009) An HOG-LBP human detector with partial occlusion handling. In: Proceedings of IEEE 12th international conference on computer vision, IEEE, Kyoto, pp. 32–39
24. Wu J, Zhang X (2001) A PCA classifier and its application in vehicle detection. In: Proceedings of international joint conference on neural networks, IEEE, Washington, pp. 600–604
25. Burges CJC (1998) A tutorial on support vector machines for pattern recognition. Data Min Knowl Discovery 2(2):121–167
26. Freund Y, Schapire R, Abe N (1999) A short introduction to boosting. J Jpn Soc Artif Intell 14:771–780
27. Beymer D, McLauchlan P, Coifman B, Malik J (1997) A real-time computer vision system for measuring traffic parameters. In: Proceedings of IEEE computer society conference on computer vision and pattern recognition, IEEE, San Juan, pp. 495–501
28. Gupte S, Masoud O, Martin RF, Papanikolopoulos NP (2002) Detection and classification of vehicles. IEEE Trans Intell Transp Syst 3(1):37–47
29. Lu X, Izumi T, Takahashi T, Wang L (2014) Moving vehicle detection based on fuzzy background subtraction. In: Proceedings of IEEE international conference on fuzzy systems (FUZZ-IEEE), IEEE, Beijing, pp. 529–532
30. Kul S, Eken S, Sayar A (2016) Measuring the efficiencies of vehicle classification algorithms on traffic surveillance video. In: Proceedings of international conference on artificial intelligence and data processing, IEEE, Malatya, pp. 1–6
31. Kul S, Eken S, Sayar A (2017) Distributed and collaborative real-time vehicle detection and classification over the video streams. Int J Adv Robot Syst 14:1–12
32. Dong Z, Wu Y, Pei M, Jia Y (2015) Vehicle type classification using a semisupervised convolutional neural network. IEEE Trans Intell Transp Syst 16(4):2247–2256
33. Wang X, Zhang W, Wu X, Xiao L, Qian Y, Fang Z (2017) Real-time vehicle type classification with deep convolutional neural networks. J Real Time Image Proc 2:1–10
34. Gao Y, Lee H (2015) Vehicle make recognition based on convolutional neural network. In: Proceedings of ICISS, IEEE, Seoul, pp. 1–4
35. Lee JT, Chung Y (2017) Deep learning-based vehicle classification using an ensemble of local expert and global networks. In: IEEE conference on computer vision and pattern recognition workshops (CVPRW), IEEE, Honolulu, pp. 920–925
36. Huo Z, Xia Y, Zhang B (2016) vehicle type classification and attribute prediction using multi-task RCNN. In: Proceedings of 9th international congress on image and signal processing, biomedical engineering and informatics (CISP-BMEI), Datong, pp. 564–569
37. Kim PK, Lim KT (2017) Vehicle type classification using bagging and convolutional neural network on multi view surveillance image. In: Proceedings of IEEE conference on computer vision and pattern recognition workshops (CVPRW), Honolulu, pp. 914–919
38. Zhou Y, Nejati H, Do T-T, Cheung N-M, Cheah L (2016) Image-based vehicle analysis using deep neural network: a systematic study. arXiv preprint
39. Wang J, Li X, Fan H, Li F (2011) Classification of lip color based on multiple SVM-RFE. In: Proceedings IEEE international conference on BIBMW, Nov. 2011, pp. 769–772
40. Zheng L, Li X, Yan X, Li F, Zheng X, Li W (2010) Lip color classification based on support vector machine and histogram. In: Proceedings 3rd International on CISP, Oct. 2010, pp. 1883–1886

41. (2015) Tzutalin. LabelImg. https://github.com/tzutalin/labelImg
42. Zhen D, Yuwei W, Mingtao P et al (2015) Vehicle type classification using a semisupervised convolutional neural network. IEEE Trans Intell Transp Syst 16(4):2247–2256
43. Chen P, Bai X, Liu W (2014) Vehicle color recognition on urban road by feature context. IEEE Trans Intell Transp Syst 15(5):2340–2346