Journal of Ambient Intelligence and Humanized Computing (2022) 13:1603–1614
https://doi.org/10.1007/s12652-019-01429-5

ORIGINAL RESEARCH

# Multiple vehicle tracking and classification system with a convolutional neural network

HyungJun Kim[1]

Received: 16 October 2018 / Accepted: 26 August 2019 / Published online: 30 August 2019


## Abstract

This paper proposes a traffic monitoring system that detects, tracks, and classifies multiple vehicles on the road in real time using various digital image processing techniques and the process of machine learning based on a convolutional neural network (CNN). With this system, a video camera is installed on the road, and calibration is used to obtain the projection equation of the actual road on the image plane. Several image processing techniques, such as background modeling, background extraction, edge detection, and object tracking, are used to develop and implement a prototype system. The proposed system also uses a transfer learning process that is more efficient than starting CNN from scratch. This maximizes training efficiency and increases prediction accuracy in vehicle classification. Preliminary experimental results demonstrate that multiple vehicle tracking and classification are possible while calculating vehicle speed. The ultimate goal of this study is to develop a single digital video camera system with embedded machine learning process that can monitor and distinguish multiple vehicles simultaneously in multiple lanes.

**Keywords** Vehicle tracking system · Surveillance camera · Adaptive background extraction · Transfer learning · Convolutional neural networks

## 1 Introduction

This paper introduces a vehicle monitoring system that uses a road-mounted surveillance camera system to measure different vehicle types and speeds in real time. Automatic video analysis from traffic surveillance cameras is technology based on image processing that is rapidly developing. In recent years, a number of research projects have sought to develop an automatic traffic management system (Buch et al. 2011). The main purpose of a traffic management system is to observe the road environment and efficiently provide accurate, real-time information about the condition of vehicles as well as the road (Mu et al. 2016). Traffic monitoring using image processing technology has several distinct advantages over inductive loop-based detectors. Inductive loop-based technology can collect only traffic flow data from specific points, while image processing technology provides a wide range of standard traffic information about traffic flow at one point, as well as monitoring larger areas. Another unique feature of video monitoring is that real-time images can be sent to the monitoring control center to provide additional information. Video surveillance systems provide much more information, including the plate number, location, speed, classification and status of vehicles. Video cameras are also inexpensive compared to loop sensors, making them easy to install and maintain. In surveillance, it is important to recognize not only the speed of a vehicle but also the type. Generally, in Korea, large vehicles cannot be driven on the left-most lane of the road and motorcycles are not permitted on certain highways. Conversely, regular passenger cars are not allowed to use the left-most lane at certain times because this lane is designated for buses. Currently, vehicle speed violations are measured on most roads in terms of loop-type, camera-type, or a combination of equipment. Because loop-type speed detectors currently laid on the road cannot identify the type of vehicle, we differentiate the category of vehicle by implementing a machine learning algorithm with a camera. This is for both speed measurement and vehicle categorization.

Detection of vehicles on the road has drawn attention in many areas of research and has been widely applied to

✉ HyungJun Kim
harry@hansei.ac.kr

[1] Hansei University, Gunpo-City, Republic of Korea

many fields in daily life, such as public safety and security (Manana et al. 2017). Capobianco et al. (2018) investigated the processing of FM-CW radar signals and the capability of CNN for vehicle classification. Koga et al. (2018) proposed using hard example mining (HEM) in the training process of a CNN for vehicle detection in aerial images. However, they used radar or aerial images. Dong et al. (2015) proposed a vehicle type classification method using a semi-supervised convolutional neural network from vehicle frontal-view images. Despite the system's excellent perception, the disadvantage of the algorithm is that it analyzes only one vehicle that comes from the front. Adu-Gyamfi et al. (2017) proposed a system that separates object recognition into two main tasks: localization and classification. Kim and Lim (2017) introduced a new vehicle type classification scheme using four concepts to increase the performance on the images from a multi-view surveillance camera. Monitoring or detecting vehicles on the road accurately in real time is one of the tasks of image processing and artificial intelligence research. For this purpose, computer vision techniques are commonly used, and we have applied CNN here to increase the accuracy of detection of lane violations.

Object recognition using machine learning has developed rapidly and has been useful in many applications. In recent years, machine learning techniques have evolved substantially in computer vision, especially in object recognition. Such networks have received much attention because they have achieved very high accuracy in image recognition applications, and in some cases even outperformed human recognition. Deep Learning is a machine learning technique that uses deep neural networks, a structure that learns how to express high-level features with many layers (Deng and Yu 2014). The goal of deep learning is to perform end to end learning. This means that the image itself is an input and both features and classifications are learned directly from the image. Krizhevsky et al. (2012) showed that a large and deep convolutional neural network—known as AlexNet—can also be used as purely supervised learning to produce excellent results for ImageNet that are very difficult to recognize. Nair and Hinton (2010) also showed how to create a more powerful type of hidden unit for a Restricted Boltzmann machine by combining the weights and biases of an infinite set of binary units. Deep convolutional neural networks with a Rectified Linear Unit (ReLU) can train several times faster than their equivalents with $\tanh(x)$ units. This is important, because faster learning methods affect the performance of larger models that are trained on large datasets. Overfitting refers to the problem of analyses that may not fit additional data or predict future observations reliably by responding too closely or with too much accuracy to a particular set of data. To avoid overfitting problems, Nair and Hinton (2010) adopted a normalized approach called dropout. That is, half of the activations of each training sample at the hidden layer are set to a probabilistic zero during training. In this study, we classify vehicles into five categories: cars, sport utility vehicles (SUVs), buses, trucks, and motorcycles using object recognition algorithms based on neural networks.

In previous research, we developed technology that used a single camera to measure the speed of cars in different lanes (Kim 2013a). In the current study, we have developed a system that uses a single camera to calculate vehicle speed whilst identifying categories of vehicles. This system allows the user to identify prohibited vehicles in designated lanes while simultaneously checking for speed violations by means of one camera mounted over the road. One inexpensive camera device will ensure smooth traffic flow, because it can identify the type of vehicle and speed violations simultaneously. The main purpose of this study is to demonstrate the validity of machine learning (which can distinguish in real time the types of vehicles) and the effectiveness of image processing technologies in building an embedded system that can monitor multiple vehicles in different lanes. The ultimate goal of this study is to make embedded-type devices that can be applied to CCTV on the road.

This paper is presented in the following order: Sect. 2 explains the overall monitoring system. Section 3 describes the adaptive algorithm for background extraction, and compares it with other methods. It also describes how to extract regions of interest. Section 4 demonstrates how to estimate the actual speed of a vehicle on the image plane. Section 5 discusses convolutional neural networks for machine running and the transfer learning process we have used. Section 6 presents comparative experimental results. Finally, Sect. 7 presents the conclusions of this research.

## 2 The monitoring system

This section briefly describes the overall system for measuring vehicle speed and detecting types of vehicles. Figure 1 shows the overall flow chart of the proposed traffic monitoring system through image processing techniques with Convolution Neural Networks. We obtain still images of 320 pixels by 240 pixels from the video sequence being entered, to which we apply image processing techniques. We first extract the background from the image sequence to be used as the reference image for vehicle detection. Typically, surveillance cameras are installed on the road. Therefore, the foreground image changes rapidly and the background rarely changes. This property allows the extraction of objects (i.e. moving vehicles) from the background. We apply a fast and simple adaptive method for background extraction. We then carry out lane detection

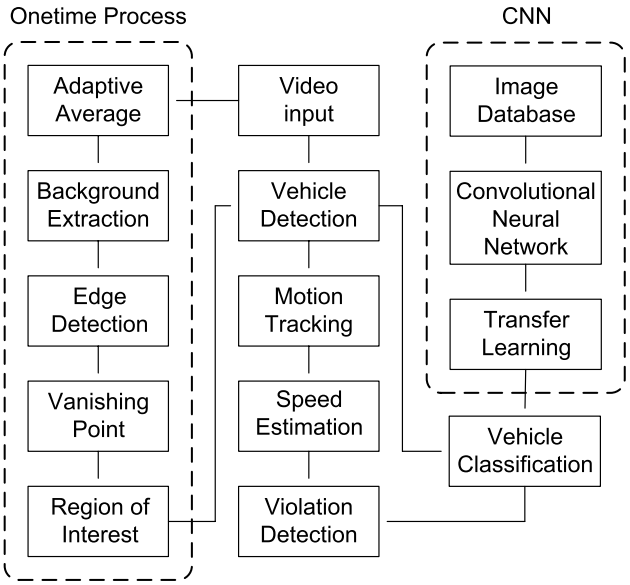


**Fig. 1** Flowchart of the proposed traffic monitoring system

and vanishing point calculation processes. From the extracted background image, lane makers and centerlines are detected using Canny edge detection and Hough transform. Vanishing points can be calculated using selected lane marking information, and one vanishing point is acquired to measure the speed of a particular passing vehicle. Instead of processing the entire input image, we also define some regions of interest to save calculation time and efficiently detect the vehicle. Then we track and identify the actual vehicle based on its temporal and spatial characteristics. Finally, we apply a perspective projection model to estimate the actual distance of the road from the point of view of the input image.

We create an image database for five vehicles (i.e. cars, SUVs, buses, trucks, motorcycles) separated by size and shape, and use them in training for machine learning. Instead of using CNN from the start, we use the best-known AlexNet and modify it through transfer learning (Krizhevsky et al. 2012). Out of the 25 layers of AlexNet that can distinguish between 1000 categories, only the last three layers (23rd layer for Fully connected, 24th layer for Softmax, and 25th layer for Classification output) are transformed for five categories. This reduces the experimental time required for machine learning and reduces the overall time needed to increase the recognition rate of the vehicle type. The overall system shows that a single camera can identify the type of vehicle and measure its speed simultaneously. Note that pre-processing using input images and the training process using an image database only need to be done once. When they are ready, the speed estimation and classification of vehicles from the input image sequence will proceed in real time.

## 3 Background and region of interest

Various image processing techniques should be applied to estimate the speed of a vehicle from video footage. Among these, background extraction is a critical pre-processing procedure for efficient development of detection of moving vehicles in real time (Piccardi 2014). It provides information with the most functional characteristics, but is highly sensitive to active scene changes due to illumination and external influences. We therefore propose a simpler process for analyzing the image sequence to extract the background. In addition, the vanishing points in the image also contain important information for camera calibration. A vanishing point is the point at which projected parallel lines intersect. Lane markers starting from the vanishing point can be modeled in a series of straight line equations. Several methods have been proposed to utilize the line segments detected in the images. The objective of this section is to design automatic background extraction and vanishing point detection systems as pre-processing for the tracking task and counting of vehicles on a road.

### 3.1 Background extraction

The performance of the background extraction algorithm depends primarily on how the background is modeled. We have considered a variety of approaches, from simple to more complex probabilistic modeling techniques (Sobral and Vacavant 2014). First, information about the background can be configured in the image via a Gaussian mixture model (Bouwmans et al. 2008). The Gaussian mixture model is one of the most widely used recursive modeling techniques when there is no prior information about the situation (Stauffer and Grimson 2000). The probability of a particular pixel with a value of $x_K$ at a specific time can be determined by the $M$ weighted Gaussian sum:

$$p(x_K) = \sum_{i=1}^{M} w_i \mathrm{N}(\cdot) \tag{1}$$

where $w_i$ is the weight of the $i$th Gaussian and $N(\cdot)$ is the normal distribution. One of the main advantages of this method is that it does not destroy existing models that may be part of the background. The Gaussian mixture model is a parametric and allows adaptive updating of model parameters without maintaining large video sequences. However, it requires much more computational complexity than other methods.

If a camera takes images of the same spot several times, one can get different images of the same place. By averaging these images, a background image can be obtained. Assuming that there are $n$ copies of the same street image

with different cars at different speeds, then the *i*th captured noisy image is $A + N_i$, where $A$ is the ideal background image and $N_i$ can be interpreted as normally distributed with mean 0. We can find the mean $A'$ of these images by using a typical averaging method:

$$A' = \frac{1}{n}\sum_{i=1}^{n}\left(A + N_i\right) = A + \frac{1}{n}\sum_{i=1}^{n} N_i \tag{2}$$

It is easily proven that the mean of all $N_i$ is close to zero. Thus, the average image is very close to the ideal background image and closer to approximation when using a larger number of input images (Cucchiara et al. 2003). However, there is the fundamental problem of storing a large number of input images in memory in order to calculate the average.

We present an adaptive algorithm to overcome the complex calculation process of the Gaussian mixture model and eliminate memory problems in the average method. If a pixel of the current frame $I_t$ has a value greater than that of the background image $B_t$, increase the pixel value of the next background image $B_{t+1}$ slightly, and in the opposite case, decrease it slightly as shown in the following formula:

$$\begin{cases} B_{t+1}(i,j) = B_t(i,j) + \delta, & \text{if } I_t(i,j) \geq B_t(i,j) \\ B_{t+1}(i,j) = B_t(i,j) - \delta, & \text{if } I_t(i,j) < B_t(i,j) \end{cases} \tag{3}$$

where $\delta$ is the properly chosen update rate. Calculations of the proposed method can be applied to real-time processing because only two consecutive frames are required. A single background model is updated repeatedly based on each input frame, eliminating the need for memory space. In addition, update values are modified in pixels to prevent errors from continuing to be passed to the next frame. The performance of the proposed adaptive average method is similar to that of the most powerful and highly sophisticated methods. Detailed comparison results of various background extraction algorithms are presented in the following section.

### 3.2 Region of interest

After extracting the background from the input image sequence, we apply the Canny edge detection algorithm for edge information to extract lanes on the road (Canny 1986). The algorithm looks for the gradient of a bright area with large spatial differences after smoothing the image. It then inhibits non-maximum areas and uses two thresholds. If the gradient is between two threshold values, it is considered a boundary. It is excluded from the edge if the gradient value is less than the lower threshold; otherwise it is considered an edge if the value lies between two threshold values and is associated with neighboring edge points. The Hough transformation is applied to detect straight lines, such as lane markers or centerlines, from the obtained edge information (Duda and Hart 1972; Fernandes and Oliveria 2008). All straight lines can be localized using the Hough transformation, and the straight line detection process can be restricted within the appropriate region of interest in the image to increase algorithm efficiency and reduce computational burden. We apply a region of interest to each input image to proactively remove unnecessary information, such as trees, billboards, and traffic signals at the top of the image. We then concentrate on moving vehicles. Among the detected straight lines, we check the angle and focus on the lines located between 45° and −45°, because the lane markers of the road image are usually within this angle range (Kim 2013b). This region of interest (ROI) is an area between the left line of the fourth lane and the centerline, from the middle of the image to the bottom of the image. In the upper half of the image, the size of the vehicle is so small and distorted that the information is meaningless. This area is initially determined by straight lines obtained from the lane detection process. The connected component extraction and spatial classification processes are only performed on the vehicle candidate objects located in the ROI area. By only handling certain areas, real-time image processing is possible and misidentification can be reduced. We can also quickly identify reliable and effective information about moving vehicles and reduce the cost of computing built-in systems.

To find vanishing points from straight lines obtained from lane markers and centerlines, the contact points of the lines must be calculated (Rother 2002). We need to calculate the intersection of two lines among the detected lines, where the two lines represent different lane markers in the background image. Since straight lines of several lane markers can be detected from the input images, a random combination of two straight lines is selected and used to calculate the vanishing point. If two lines intersect at a certain point $(x, y)$, then this point must be on both lines, so both line equations must be satisfied at the point. By solving these two equations simultaneously, we can calculate the vanishing points of the lines. Due to small deviations in the detected straight lines, the vanishing points are not merged into one point. However, these errors are negligible and the average of these vanishing points can later be used as the final vanishing point as the basis for predicting vehicle speed. Figure 2 shows an image of a vehicle approaching the surveillance camera, the extracted background image, several selected edges of lane markers with vanishing points, and the region of interest where vehicles are tracked. Using this information, we apply object tracking and classification algorithms to the vehicle, which will be described in the following section.

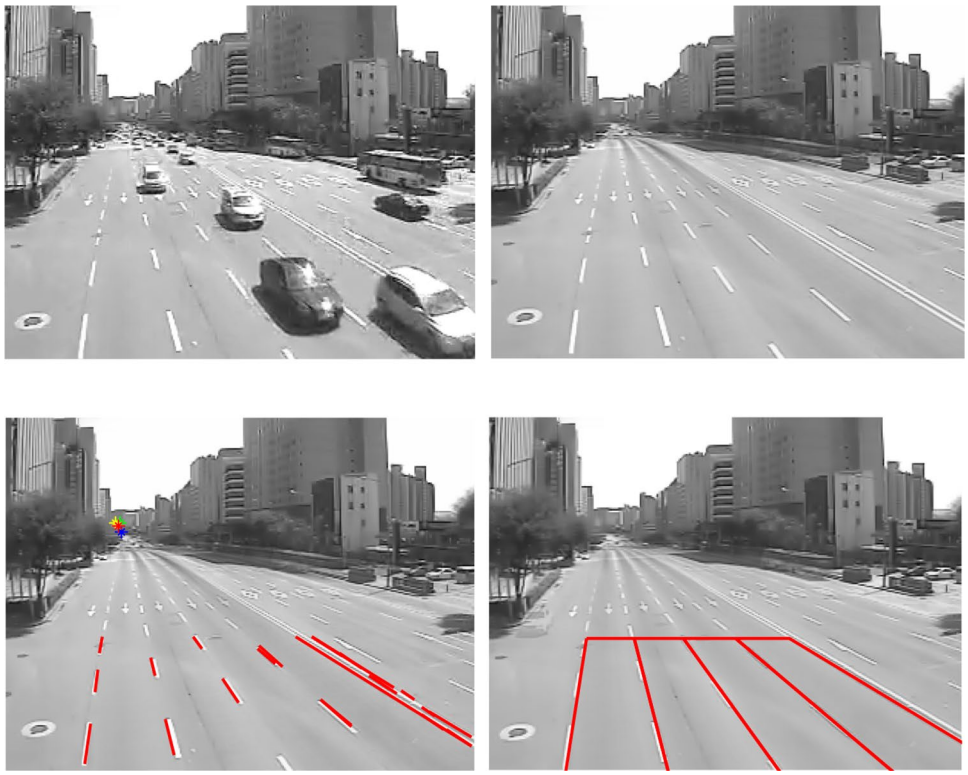

**Fig. 2** Example images of original oncoming vehicles, the extracted background, selected edges of lane markers with vanishing points, and region of interest, respectively

## 4 Vehicle speed estimation

The length of a random straight line on a road can be calculated from the measured coordinate value of two points on a straight line on the image plane (Cathey and Dailey 2004; Schoepflin and Dailey 2003). The relationship of the vehicle trajectory between that on the ground plane and that on the image plane can be explained using the vanishing points of lines. The image coordinate system gradually shrinks until a vehicle located relatively far away from the camera is placed in a higher position and the vanishing point is reached. Thus, the lane markers departing from the vanishing point may be modeled with a series of straight lines as described in Sect. 3. Figure 3 illustrates an example of a vehicle trajectory on the image plane.

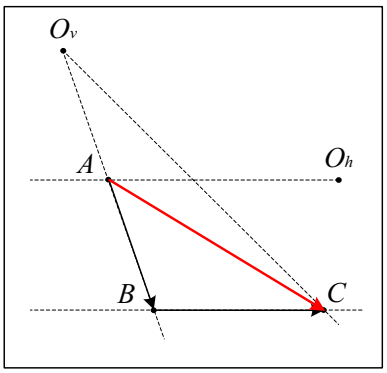


**Fig. 3** Example of a vehicle trajectory on the image plane

The origin (0, 0) of the imaging system is assumed to be in the upper left corner of the image. Since we use only one surveillance camera, we can assume that the horizontal vanishing point, $O_h$, is located at the infinity point, and that the vertical vanishing point, $O_v$, is the vanishing point achieved during pre-processing. Suppose that the specific locations $A'$, $B'$, and $C'$ on the ground plane correspond to $A$, $B$, and $C$ on the image plane, respectively. The vehicle then moves from point $A$ to point $C$ via point $B$ on the image plane as the vehicle moves from point $A'$ to point $C'$ on the ground plane. Point $A$ is the intersection of lines $O_hA$ and $O_vA$ for calculating the travel distance; the length of line $AC$ is the vector sum of lines $AB$ and $BC$. We can calculate the real lengths of $A'C'$ from the coordinate values of $A$, $B$, and $C$. We can calibrate the measurement function using the length of the actual road segment, such as a lane marker that can be identified in the image. Therefore, vehicle speed can be measured by calibrating the actual distance between the vanishing point and two known points on the road. We apply the perspective projection model to estimate the actual distance of the road using input images.

The rigid components of a moving vehicle are classified as a possible target vehicle for tracking by evaluating motion vector information (McFarlane and Schofield 1995). We assume that the vehicle moves along the road at a constant speed. Suppose the vehicle in the image located at position

$(x, y)$ is denoted as $I_t(x, y)$ at time $t$ and $I_{t+1}(x, y)$ at time $t+1$, respectively. Object based tracking is used to find the motion vector $(d_x, d_y)$, which minimizes the following residual function within the search area:

$$\varepsilon(d_x, d_y) = \arg\min_{d_x, d_y} \sum_x \sum_y \left| I_t(x, y) - I_{t+1}(x + d_x, y + d_y) \right| \tag{4}$$

The position of the newly matched vehicle is updated with minimal residual functionality from the existing vehicle. Using the relationship between the aforementioned image plane and the ground plane, the vehicle speed can be predicted by recording the time and position of the ROI of each vehicle. Once all the components are procured from the input images, we can estimate the speed of the tracking vehicle. Figure 4 shows some results of tracking multiple vehicles in the region of interest under various circumstances. It can be seen that the monitoring system simultaneously tracks different types of vehicles.

## 5 The selected convolutional neural network

Many studies have been conducted to analyze the types of stationary or moving vehicles, or to measure the speed of a vehicle, but systems that analyze both the speed and type of a moving vehicle have not been proposed. We create an image database for each vehicle class to distinguish them by type and then use it to train for machine learning. Instead of using CNN from the beginning, we utilize the most well-known AlexNet, and modify it through a transfer learning process. The proposed system uses a transfer learning process that is more efficient than starting with CNN. This maximizes training efficiency and increases prediction accuracy of vehicle classification. The entire system shows that a single video camera is sufficient to extract the background image and simultaneously calculate and detect the vanishing point applied to the traffic monitoring system for preprocessing. At the same time, the system uses CNN to train on how to classify the types of vehicles.

Convolutional Neural Network is a powerful and commonly used type of neural network that is designed for applications whose inputs have a two-dimensional structure like an image. Like all neural networks, CNN has multiple layers and information is passed between them. Although the architecture of a network, such as the type, size, and order of layers, depends on the person who originally designed the network, there are many variables in the layers themselves known as weights. These weights determine how a layer behaves when data is passed through it. The values of these weights are earned by training the network on known data, and the nature of the network is determined by that data. A CNN passes an image through the network layers and outputs the final class. The network can have multiple layers, each layer learning how to detect different features. Most

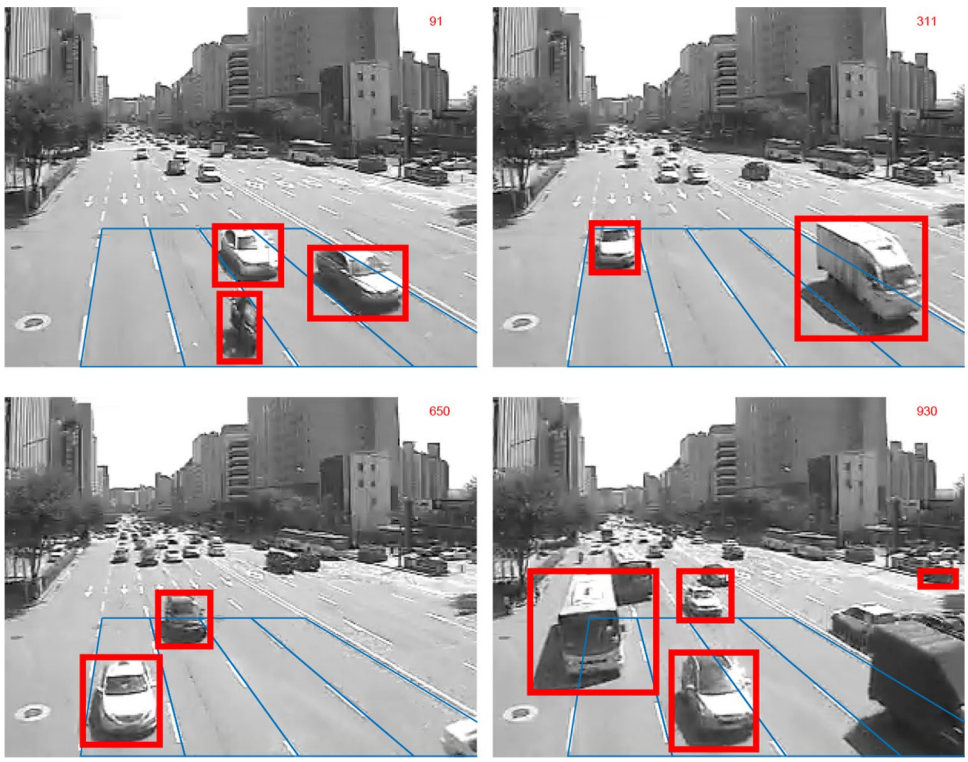


**Fig. 4** Examples of multiple vehicle tracking results with overlapped ROI under various circumstances

of the layers consist of convolution, pooling, and rectified linear unit layers, which allow us to import the original input image and extract various features for classification. Various kinds of filters are applied to each training image at different resolutions, and the output of each convolved image is used as the input for the next layer. These filters can be started with very simple features, such as boundaries and corners, and as layers progress, they can increase in complexity for features that uniquely define objects.

AlexNet, a pre-trained network, is an example of a CNN (Krizhevsky et al. 2012). It consists of five convolution layers and three fully connected layers, and the last fully connected layer uses Softmax as an active function for classification into 1000 predetermined categories. Instead of creating a CNN from scratch, it is much more effective to take AlexNet and modify it to fit our problems and train it with images of vehicles. The process of modifying layers and retraining new data using pre-trained networks is called transfer learning, which is a very effective way to solve many deep learning problems. Figure 5 illustrates a simplified diagram of transfer learning using AlexNet. Generally speaking, AlexNet consists of convolution, pooling, ReLU, fully connected layers, and a Softmax layer. For classification using deep running, a color image of 227 pixels by 227 pixels is placed into the network. Convolution activates certain features from the image by passing the input image through a series of filters. Pooling simplifies the output by performing nonlinear down sampling to reduce the number of parameters the network needs to learn. ReLU provides faster and more effective training by mapping negative values to zero and maintaining positive values (Nair and Hinton 2010). Fully connected layers convert 2D spatial features of the network to 1D vectors representing the image-level features for classification. If we look at layers of pre-trained networks like AlexNet, the 23rd layer is a fully connected layer with 1000 neurons. This maps features extracted from the previous layer to 1000 output classes. The following Softmax layers convert the raw values of 1000 classes into normalized scores, each of which can interpret the probability that an image belongs to a certain class as a prediction of the network. Softmax provides probabilities for each category of data set. The last layer then takes this probability and returns the most likely class for network output.

When carrying out transfer learning, it is usually only necessary to change the last few layers to accommodate a particular problem. The network we are trying to launch in this way has the same pre-trained network and feature extraction behavior, but has not yet been trained to apply these features to vehicle classes. There are more details about the different types of CNN layers, but generally speaking, most networks transform input images into a set of features that the last few layers can use to perform classification. For this classification problem, the output layer returns the strength of the network prediction for each possible class: cars, SUVs, buses, trucks, or motorcycles.

## 6 Experiments and discussion

We show the experimental results here, first showing the results of background extraction and secondly the results of training using the transfer learning process, and finally the speed estimation and vehicle classification results of proposed algorithms applied to actual image sequence.

### 6.1 Background extraction results

Background extraction is an important pre-processing procedure for efficient detection of moving vehicles in real time. Because the performance of the background extraction algorithm depends primarily on the background modeling method, the experimental results are presented here as the peak signal-to-noise ratio (PSNR) values. Figure 6 shows the PSNR values for average, Gaussians mixture models, and the proposed adaptive method. The average image was calculated by acquiring the first 200 video frames and used as a reference image to compare with the proposed method. Experiments show that a PSNR value of about 30 dB is sufficient to extract a good background image. The PSNR value of the average method increased exponentially with the addition of image data to calculate the mean. For the Gaussian mixture models, the background was not adapted fast enough, so the PSNR values gradually increased to 30 dB. The proposed adaptive method converges very quickly to 30 dB. It is not only simple, but also requires no parameter settings and uses only the previous and current frames, so

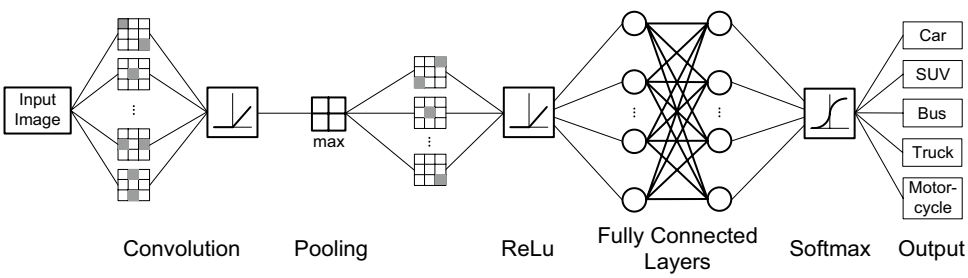


**Fig. 5** Simplified diagram of transfer learning using AlexNet

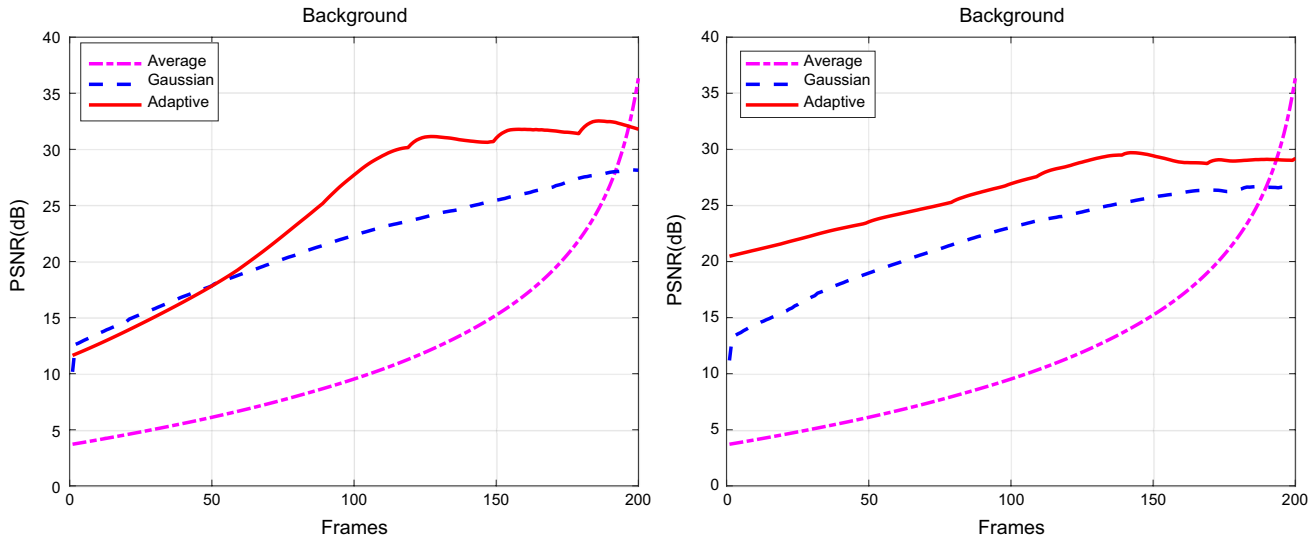


**Fig. 6** PSNR values for average, Gaussians mixture models, and the proposed method

no memory space is required. On the other hand, if there is a fast moving object in the image at a value close to 30 dB, a slight fluctuation occurs because only two consecutive frames are used for the background extraction process. The reason for the up and down after the 100 frame portion of the graph is to account for sudden changes in the brightness of the input image. Even if only two frames are used, there may be some leaps and bounds depending on the change in the overall image. Nevertheless, these changes do not affect the extraction of moving objects. Experimentation with a real urban image sequence shows that the proposed method produces better performance compared to other background extraction techniques as shown in Fig. 6. The proposed method is powerful for monotonous movement and is very efficient for real-time processing.

## 6.2 Transfer learning process results

In this section, we explain how to fine-tune a pre-trained deep neural network for a new recognition task. Because pre-trained layers at the end of the network are designed to classify 1000 objects, we have to classify the objects to perform a new five classification task. The first step in transfer learning is to replace the last three layers of the pre-trained network with a set of layers that can categorize five classes. First, import and use the layers of the network that define network architecture and contain learned weights. Out of the 25 layers of AlexNet that can distinguish between 1000 categories, only the last three layers are transformed for five categories. The data is five different categories of automobiles: cars, SUVs, buses, trucks, and motorcycles; each category has 200 images. Note that there should be an equal number of images in each class. During the training process, the network weight is adjusted so that the network associates with the desired output from a given input. It is generally recommended that some training data be set aside for testing. This reserved test set is not used to train the network but is used only for performance evaluation. Therefore, we divide the data into learning and testing sets. We use 80% of images per category to train, and specify 20% as a validation set to test our network after it has been trained. We want to see how accurate our network is on data it has never seen before, which is the new input video sequence.

We change the network slightly for fine-tuning, and how much the network changes during training is controlled by learning rates. It does not modify the learning rate of the original layers except the last three, because the rates of these layers are already quite low and do not need to be lowered anymore. Instead, we increase the learning rate of the new layer we have added, so that it changes faster than the rest of the network. In this way, the previous layers do not change significantly and we learn the weight of the new layer quickly. We want the last layer to train faster than the others. This is because training can quickly improve the last layer and keep other layers relatively unchanged. Figure 7 shows two examples in which the accuracies of the mini-batch accuracy are very low in the beginning and quickly converge to 100% as iteration increases. At each iteration, a subset of the training images (called mini-batch) is used to update the weights. In each iteration, a different mini-batch is used, and one epoch refers to the use of the entire training set. The final prediction accuracy is different from each training session since we have trained 200 images randomly divided into 8:2 ratios, but the lower one is approximately 85% and the higher one is about 98%.

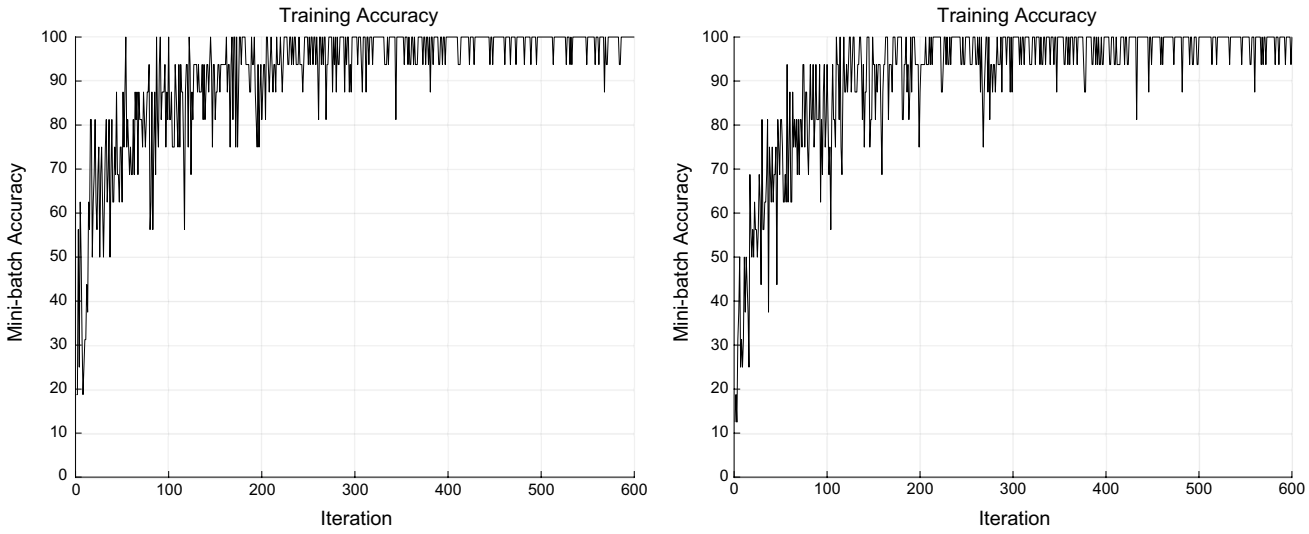


**Fig. 7** Examples of mini-batch training accuracy according to iteration

### 6.3 Speed estimation and vehicle classification results

In this section, the experimental data on measuring the speed of the vehicle and the analysis of the vehicle type are introduced using input image sequences. Figure 8 shows five example images of speed estimation and classification of vehicles. The images on the leftmost column show the tracking areas of vehicles approaching the surveillance camera. The upper right corner of each image represents the sequence number of each frame. The speed of the vehicle closest to the camera is shown with the lane number on the upper left corner of the next frame, as shown in the second column images, where L1 represents the first lane closest to the centerline, L2 represents the second lane, and so on. The speed of a vehicle is estimated by time and distance as the vehicle enters the region of interest and the result is displayed when the vehicle leaves the zone. At the same time the vehicle can be captured to determine the type of class as shown in the third column of images. The captured vehicle image is made from a color image of the appropriate size of 227 pixels by 227 pixels by 3 for classification and then entered into the transferred AlexNet for the classification process. It displays the type of vehicle with its prediction score of classification. Some are highly likely and some are low in probability (depending on the training process) but generally show five classifications correctly. Because current CNN needs a fixed size (e.g. $224 \times 224 \times 3$) input image, this can increase misclassification during the process of increasing input images to this size. He et al. proposed a new network structure, the SPP-net, which can generate fixed-length representations regardless of image size or scale (He et al. 2015). With the SPP-net, there is no need to increase the size of input images, which may increase the recognition rate. Therefore, in real time, it is possible to know which type of vehicle has moved to which lane at what speed. This allows detection of speed limit violations or certain lane driving violations.

Finally, we show some additional examples of correct and incorrect classifications. Figure 9 shows additional examples of vehicles that are correctly classified into five categories. Figure 10 also shows examples of misclassified vehicles. The first example was the misperception of an SUV as a car, and the second one was the misinterpretation of a truck as an SUV. The rest of the examples also show incorrect recognition of a truck as a car, a motorcycle as a car, and a car as a bus. This is because there are only 200 image samples for each category we use, of which only 80% are used for the learning process and the remaining 20% for validation. If the number of images being studied is increased sufficiently, such misperception results will be dramatically reduced.

## 7 Conclusions

In this paper, we proposed a traffic monitoring system that would determine the speed of multiple vehicles simultaneously, as well as identify the types of vehicles. To this end, a more efficient, high-precision vehicle detection and tracking approach was proposed, and a machine learning process was developed to classify vehicle types. The experimental results show that a vehicle can be tracked and its speed can be estimated at the same time, as well as being classified in terms of vehicle type. The vision-based multi-vehicle monitoring system proposed in this study is very accurate, and provides effective technology for future automatic traffic surveillance

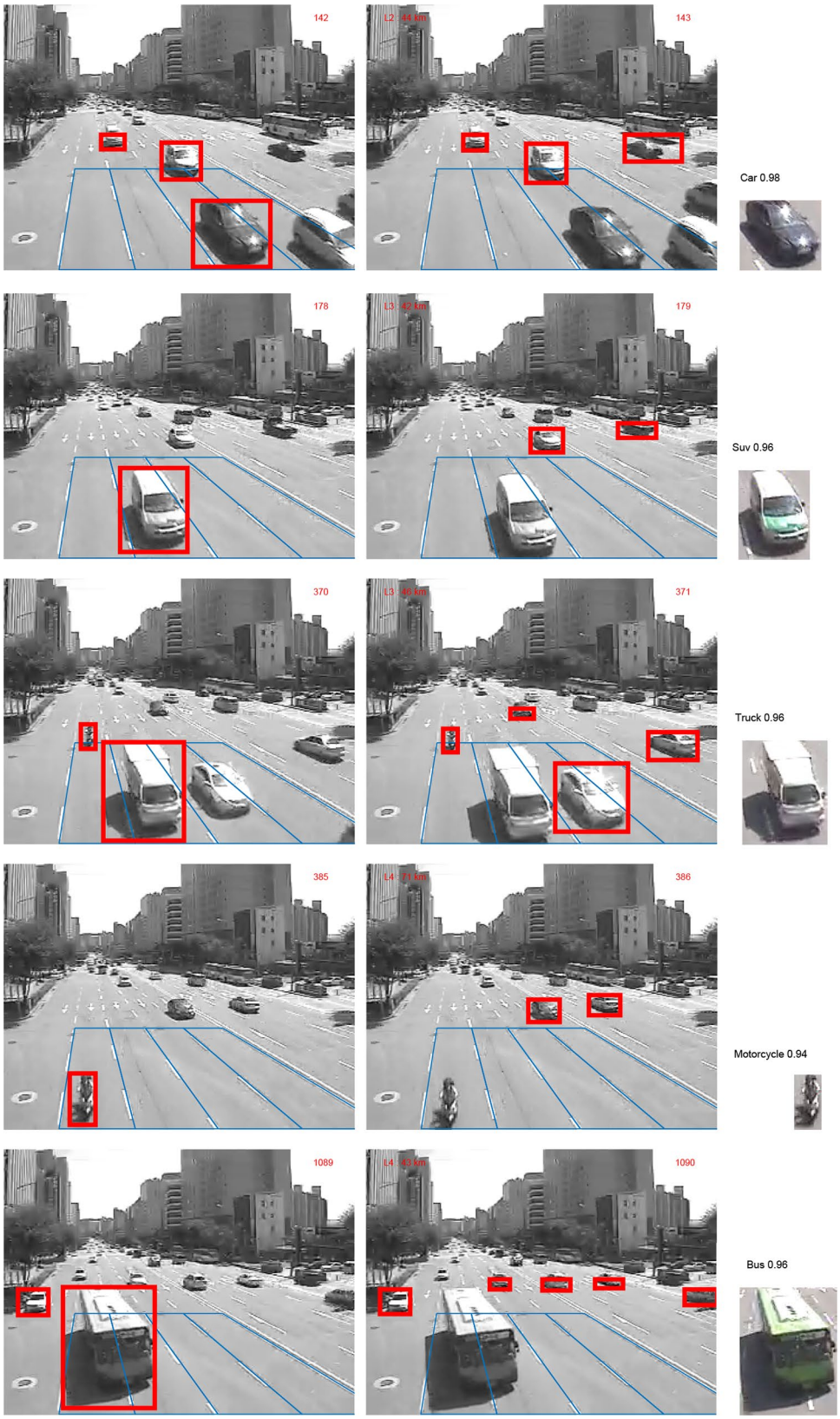


**Fig. 8** Example images of speed estimation and classification of various vehicles

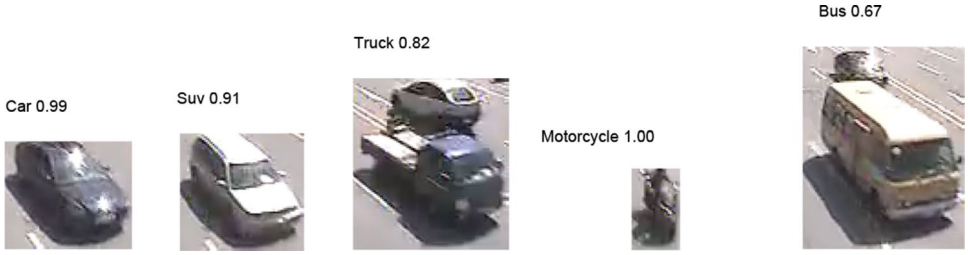


**Fig. 9** Examples of vehicles that are correctly classified into five categories

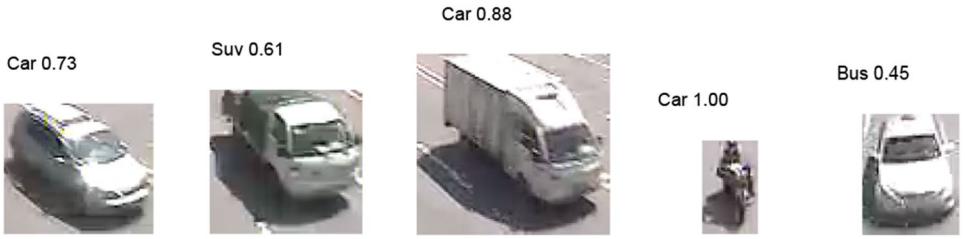


**Fig. 10** Examples of misclassified vehicles

systems. A single digital video camera with built-in video processing systems can be a very cost-effective and beneficial system for detecting, tracking and monitoring multiple vehicles simultaneously in multiple lanes. Although complex techniques often provide excellent and improved performance, our experiments show that the proposed simple techniques can yield better results with much lower computational complexity and lower memory requirements. The experiment also shows that the proposed method is applicable to real-time visual surveillance applications and can be developed as an embedded monitoring system.

The problem of expressing knowledge within a computer or using a computer to deduce knowledge is a problem in the field of knowledge engineering. In order to develop applications of artificial intelligence based on computer science, it is very important to extract necessary information through image processing. In the field of computer vision, deep learning techniques are becoming very important and highly developed tools to understand images, and they are superior to other techniques in terms of recognition results. In this study, we used only 200 images for each class of vehicles to train the neural network. Two networks can have the same architecture, but when trained using different datasets, they behave differently. There is no doubt that training networks using more images of vehicles in these algorithms will yield better recognition results. Because this requires a very large image database, we have only conducted preliminary experiments with potential for this paper, and after more experiments we plan to produce results based on the convolution neural network. The final goal of our study is to develop a single digital video camera system with embedded machine learning process that can monitor and distinguish multiple vehicles simultaneously in multiple lanes. In real time, we will be able to see which type of vehicle has moved to which lane at what speed, which will allow us to detect speed limit violations or certain lane driving violations.

## References

Adu-Gyamfi Y, Asare S, Sharma A, Titus T (2017) Automated vehicle recognition with deep convolutional neural networks. Transport Res Record J Transport Res Board 2645:113–122

Bouwmans T, Baf FEl, Vachon B (2008) Background modeling using mixture of Gaussians for foreground detection—a survey. Recent Patents Comput Sci 1(3):219–237

Buch N, Velastin SA, Orwell J (2011) A review of computer vision techniques for the analysis of urban traffic. IEEE Trans Intell Transport Syst 12(3):920–939

Canny JF (1986) A computational approach to edge detection. IEEE Trans Pattern Anal Mach Intell 8(6):679–698

Capobianco S, Facheris L, Cuccoli F, Marinai S (2018) Vehicle classification based on convolutional networks applied to FM-CW radar signals. Adv Intell Syst Comput 728:115–128

Cathey FW, Dailey DJ (2004) One-parameter camera calibration for traffic management cameras. In: IEEE intelligent transportation systems conference, Washington, D.C., pp.865-869

Cucchiara R, Piccardi M, Prati A (2003) Detecting moving objects, ghosts, and shadows in video streams. IEEE Trans Pattern Anal Mach Intell 25:1337–1342

Deng L, Yu D (2014) Deep learning: methods and applications. Now Publishers Inc., Boston

Dong Z, Wu Y, Pei M, Jia Y (2015) Vehicle type classification using a semisupervised convolutional neural network. IEEE Trans Intell Transport Syst 16(4):2247–2256

Duda RO, Hart PE (1972) Use of the Hough transform to detect lines and curves in pictures. Commun ACM 15:11–15

Fernandes AF, Oliveria MM (2008) Real-time line detection through an improved Hough transform voting scheme. Pattern Recogn 41:299–314

He K, Zhang X, Ren S, Sun J (2015) Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Trans Pattern Anal Mach Intell 37(9):1904–1916

Kim H (2013a) Image processing based multiple vehicle monitoring system. Inf Int Interdiscip J 16(2B):1491–1496

Kim H (2013b) Detecting moving objects using background modeling and local binary patterns. Inf Int Interdiscip J 16(3B):2305–2310
Kim P, Lim K (2017) Vehicle type classification using bagging and convolutional neural network on multi view surveillance image. In: 2017 IEEE conference on computer vision and pattern recognition workshops, Honolulu, pp 41–46
Koga Y, Miyazaki H, Shibasaki R (2018) A CNN-based method of vehicle detection from aerial images using hard example mining. Remote Sens 10:1–21
Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet classification with deep convolutional neural networks. Adv Neural Inf Process Syst 25(2):1106–1114
Manana M, Tu C, Owolawi PA (2017) A survey on vehicle detection based on convolution neural networks. In: 3rd IEEE international conference on computer and communications, Chengdu, pp 1751–1755
McFarlane NBJ, Schofield CP (1995) Segmentation and tracking of piglets in images. Mach Vis Appl 8:187–193
Mu K, Hui F, Zhao X (2016) Multiple vehicle detection and tracking in highway traffic surveillance video based on SIFT feature matching. J Inf Process Syst 12(2):183–195
Nair V, Hinton GE (2010) Rectified linear units improve restricted Boltzmann machines. In: Proceedings of the 27th international conference on machine learning, pp 807–814
Piccardi M (2014) Background subtraction techniques: a review. In: IEEE international conference on systems, man and cybernetics, pp 3099–3104
Rother R (2002) A new approach to vanishing point detection in architectural environments. Image Vis Comput 20(9–10):647–655
Schoepflin TN, Dailey DJ (2003) Dynamic camera calibration of roadside traffic management cameras for vehicle speed estimation. IEEE Trans Intell Transport Syst 4(2):90–98
Sobral A, Vacavant A (2014) A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos. Comput Vis Image Underst 122:4–21
Stauffer C, Grimson WEL (2000) Learning patterns of activity using real-time tracking. IEEE Trans Pattern Anal Mach Intell 22(8):747–757