Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

A novel image-based convolutional neural network approach for traffic congestion estimation

Ying Gao [a,1], Jinlong Li [a,1], Zhigang Xu [a,*,2], Zhangqi Liu [a], Xiangmo Zhao [a], Jianhua Chen [b]

[a] School of Information Engineering, Chang'an University, Xi'an, Shaanxi 710064, China
[b] Transportation Information Center, China Academy of Transportation Sciences, Beijing 100029, China

ARTICLE INFO

Keywords:
Traffic congestion
Convolutional neural network
Vehicle detection
Deep learning
Traffic flow parameter

ABSTRACT

Traditional image-based traffic congestion estimation methods generally include two steps, which first extract the vehicles from the surveillance images, then calculate the congestion index using the vehicle counts. When working with vast amount of video frames, these approaches are time-consuming and hardly guarantee the real time detection of traffic congestion. In this study, firstly a specific and accurate definition of traffic congestion is proposed to quantify the level of traffic congestion. Then we construct an image-based traffic congestion estimation framework, in which a traffic parameter layer is integrated to the basic convolutional neural network (CNN) model. The proposed framework can directly perform traffic congestion calculation and estimation, which shortens the processing time and avoids the complicated postprocessing. A dataset of 1400 traffic images including 66,890 vehicles is collected for training the proposed CNN model. Another new dataset of 2400 traffic images including 113,516 vehicles is collected to test the proposed method on estimating traffic congestion. Experimental results show that our proposed approach has better efficiency and stability in both free flow and congested traffic conditions, as well as sunny and rainy scenes.

1. Introduction

Estimating traffic congestion is critical for monitoring road traffic conditions and maximizing the efficiency of transport operations (Bacon et al., 2011, Sadollah et al., 2019, Li et al., 2021). In particular, in intelligent transportation systems (ITS), a real-time and automatic traffic congestion estimation method is essential to alleviate urban traffic congestion (Ding et al., 2020). There are various detectors for collecting traffic flow information, such as loop detectors (Wu et al., 2016, Liu & Sun, 2014), light detection and ranging (LiDAR) sensors (Zhao, Xu, & et al., 2019), microwave detectors (Ma et al., 2015), GPS devices on the vehicle (Simoncini et al., 2018), floating car (Kong et al., 2016), etc. These detectors' ability to provide rich and precise traffic information is very limited, such as the undetectable areas of GPS devices and floating vehicles, laser and microwave detectors are easily affected by weather variations. More and more surveillance cameras in many cities have been mounted in recent years, and promising applications have been shown to monitor and provide more accurate traffic information (Li et al., 2013). The use of these available surveillance cameras to estimate traffic congestion is of great importance for alleviating traffic congestion in real-world applications (Calderoni et al., 2014).

In general, traffic information is obtained from videos or images by vehicle counting and detecting (Venkatesvara Rao et al., 2018, Hu et al., 2012). To achieve that, some traditional methods use various image processing algorithms to extract vehicles from road background, like Speeded up Robust Features (SURF) (Hsieh et al., 2014), background subtraction (Gupte et al., 2002, Kong et al., 2007), and temporal difference (Li et al., 2009). However, their performance is poor and even inaccessible in congestion and complicated traffic conditions or low frame rates (Willis et al., 2017). More recently, many deep learning methods have achieved state-of-the-art successes in some challenging tasks of computer vision (Zhao, Zheng, & Xu, 2019), such as image classification, object detection, and tracking. There are also several pieces of research focus on traffic flow parameter estimation using deep learning methods. Obviously, it is a straightforward way that using

[*] Corresponding author.

E-mail addresses: gaoying888@chd.edu.cn (Y. Gao), lijinlong1117@chd.edu.cn (J. Li), xuzhigang@chd.edu.cn (Z. Xu), 2019124090@chd.edu.cn (Z. Liu), xmzhao@chd.edu.cn (X. Zhao), chenjianhua@catsic.com (J. Chen).

[1] Co-first authors.

[2] ORCID: 0000-0002-8479-4973.

https://doi.org/10.1016/j.eswa.2021.115037
Received 19 November 2020; Received in revised form 21 February 2021; Accepted 10 April 2021
Available online 2 May 2021

current existing knowledgeable methods to estimate traffic flow parameters when accurate vehicle detection results are obtained by deep learning methods (Ke et al., 2018, Chung & Sohn, 2017). However, these approaches are time-consuming and complicated postprocessing is inevitable when vehicle detection results obtained, which is not easy to satisfy the real-time traffic congestion estimation in the ITS.

Different from traffic congestion estimation based on vehicle detection, some studies aimed at mapping the traffic image feature to the level of congestion. For feature extraction, some traditional image processing techniques are used, such as edge detection (Pan et al., 2010, Tahmid & Hossain, 2017), texture analysis (Wei & Hong-ying, 2016), histogram of oriented gradient (HOG) (Rybski et al., 2010), etc. The congestion level can be divided into several groups that can be identified by K-means, K-nearest neighbor (KNN), or support vector machines (SVM) algorithms. Furthermore, for congestion classification (Wang et al., 2018), an unsupervised learning algorithm with traffic density information is also used. Using the convolutional neural network (CNN) to classify traffic images for congestion estimation becomes an alternative, and some typical CNN models are applied, such as GoogleNet (Szegedy et al., 2015), and AlexNet (Krizhevsky et al., 2017). However, a trained CNN model is difficult to be applied to different traffic roads (e.g., expressway and urban road) because the traffic congestion threshold in different roads is inconsistent. Moreover, the simple binary classification methods (such as congested and non-congested) cannot quantify the level of congestion.

To address traffic congestion estimation and mitigate the above-mentioned issues, first, a specific and accurate definition of traffic congestion is proposed to quantify the level of traffic congestion. In this paper, traffic congestion estimation is treated as a regression problem since the traffic congestion level is converted to a real number through the proposed definition. A novel traffic congestion estimation framework for surveillance video based on the Faster R-CNN model (Ren et al., 2015) is then proposed due to the excellent computer vision performance of Faster R-CNN. Our proposed CNN model is embedded with a traffic parameter layer, which can directly perform traffic congestion estimation. To validate the proposed method, a new dataset is collected using the urban traffic surveillance video from Xi'an, including 1400 images of 66,890 vehicles. We manually labeled each vehicle in all these images to train the proposed model. Another dataset of 2400 images of 115,316 vehicles is also collected from the same camera to evaluate the proposed method. Finally, several traditional image processing methods and a deep learning method are adopted to perform comparative experiments. The results show that our proposed method achieves the best performance. In summary, the major contributions of this paper are as follows:

- A specific and accurate definition of traffic congestion is proposed. Based on this definition, the traffic congestion levels of different traffic roads (such as expressway and urban road) can be compared in a unified evaluation system.
- An image-based traffic congestion estimation framework is proposed, in which a traffic parameter layer is integrated to the basic CNN model. The proposed framework can directly perform traffic congestion calculation and estimation, which shortens the processing time.
- A dataset for traffic congestion estimation is collected and manually labeled, including four types of traffic scenes: non-congested scene on sunny days, congested scene on sunny days, rainy non-congested scene, and rainy congested scene. Each subset contains 600 images for a total of 2,400. This dataset can be used as an assessment platform for estimating traffic congestion.

2. Related work

2.1. Imaging process-based vehicle detection

Generally, it is a straightforward way to estimate traffic congestion when vehicle detection results are already available. Accurate vehicle detection results from videos and images are thus the cornerstones for the high-precise estimation of traffic congestion. The traditional image processing algorithms for vehicle detection fall into two categories. The first type is to extract moving objects (foreground) from traffic scenes (background) by analyzing the pixel-level changes. Some studies extract moving objects using temporal difference (Li et al., 2009) and background subtraction algorithm (Gupte et al., 2002, Kong et al., 2007). These techniques can be effectively applied to some simple traffic scenes with sunny conditions after the decision threshold has been manually tuned. Several algorithms are proposed to improve the performance for complex scenes, such as K-Nearest (KNN) (Keller et al., 1985), Mixture of Gaussians (Bouwmans et al., 2008), Adaptive Background Gaussian Mixture Model Background Subtraction (DPGrimsonGMMBGS) (Stauffer & Grimson, 1999), MultiLayer Background Subtraction (MultiLayerBGS) (Yao & Odobez, 2007), Pixel-Based Adaptive Segmenter (PBAS) (Hofmann et al., 2012) and Geometric Multigrid (Papandreou & Maragos, 2006). The second category is to extract the objects by their multi-scale feature extraction. In surveillance images, some uncomplicated and stationary objects can be detected using object texture, shape, color, and power spectrum (Oliva et al., 1999). For complicated object detection, there are also some methods available such as Haarlike features (Han et al., 2009), Scale Invariant Feature Transformation (SIFT) (Mu et al., 2016), Speeded up Robust Features (SURF) (Zhao et al., 2017), Local Binary Patterns Histograms (LBPH) (Ahonen et al., 2006) and Histogram of Oriented Gradient (HOG) (Rybski et al., 2010). However, traditional image processing algorithms such as ImageNet introduced in (Deng et al., 2009), can hardly show the available performance of the large public dataset. Currently, deep learning methods have achieved outstanding performances for object detection in many large public datasets. Although deep learning techniques require a large amount of labeled data to train and adjust model parameters, several layers can efficiently extract a lot of feature information to establish a complex mapping between low-level features and high-level semantics. Deep learning methods therefore have a greater ability to learn feature representation, eliminating the manual selection of the feature. Some representative deep learning models perform well in many large public datasets, such as Region-based Fully Convolutional Networks (R-FCN) (Dai et al., 2016), Single Shot MultiBox Detector (SSD) (Liu et al., 2016), You Only Look Once (YOLO) (Redmon et al., 2016), and Faster R-CNN (Ren et al., 2015). They also provide robust and advanced vehicle detection performances in different traffic scenes. Unlike all these above-mentioned, to obtain a fast and efficient performance, we focus on directly estimating traffic congestion using CNN model, rather than on the results of vehicle detection.

2.2. Traffic congestion estimation

To estimate traffic congestion, some studies use a variety of traffic flow parameters obtained from images or videos, such as space headway, density, speed, and queue length. Combined with some computer vision algorithms such as K-means clustering algorithm (Lozano et al., 2009), Kanade–Lucas–Tomasi (KLT) algorithm (Cao et al., 2012) and Optical Flow algorithm (Chen & Wu, 2015; Ke et al., 2018), they have achieved better performance for traffic flow parameter estimation using CNN methods (Ke et al., 2018, Chung & Sohn, 2017, Bautista et al., 2016). In addition, some studies concentrate on holistic approaches to create a mapping form traffic image feature to congestion level. An unsupervised learning algorithm (Yuan et al., 2016) is proposed using local density information to classify traffic congested scenes. A locality constraint distance metric learning algorithm (Wang et al., 2018) is

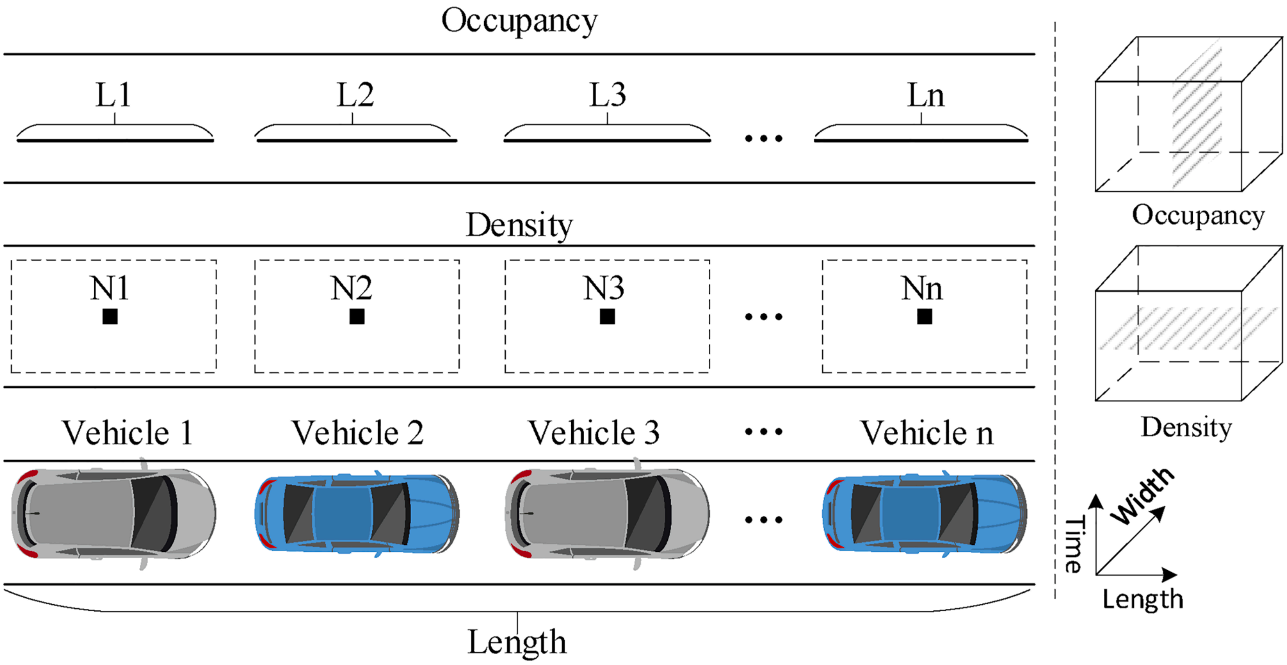


Fig. 1. Illustration of traffic congestion by traffic density and occupancy. In the right part, the density can just measure the traffic congestion at a time point, while the occupancy can only estimate the traffic congestion at a space point (Wang et al., 2018).

proposed to detect traffic congestion, which uses the low-level texture feature and kernel regression to characterize the congestion level. GoogLeNet (Willis et al., 2017) is used to compute a congestion level in surveillance camera images of London's city streets. The algorithms of Gaussian group-based histogram (GBH) (Song et al., 2011), symbolic representation classification (Dallalzadeh et al., 2013), and the extraction of texture features between congested image and unobstructed image are also used to directly estimate real-time traffic congestion (Wei & Hong-ying, 2016). Considering the different road capabilities, road characteristics, and different weather conditions, it is hard to obtain accurate results of traffic congestion estimation when directly applying the existing model to other roads. Therefore, a specific and uniform definition of traffic congestion is necessary, and thus this paper focuses on the achievement of accurate and efficient traffic congestion estimation under different traffic conditions.

3. Methodology

In this section, a unified and specific definition of traffic congestion is presented at first. The specifications of the proposed traffic congestion estimation approach based on CNN are then demonstrated based on the proposed definition.

3.1. Definition of traffic congestion

Some image-based methods currently estimate traffic congestion using one or more traffic parameters detected, such as density, occupancy, velocity, and headway. The Level Of Service (LOS) for traffic evaluation is the most widely used criterion based on vehicle speed in Highway Capacity Manual (HCM (Roess & Prassas, 2014)), which is divided into six degrees from A to F to estimate the road service capability. However, the same speed of different characters of different roads represents different traffic states such as freeway and urban road. Taking the United States as an example. Speed of less than 64 km/h is considered congestion in Washington (Lomax, 1997), while less than 56 km/h is considered severe congestion in California (Borden, 1993).

Traffic density can intuitively reflect the degree of spatial congestion, and occupancy could represent temporal congestion of the road (Wang et al., 2018). It is thus more sensible that both traffic density and occupancy are taken into consideration to our proposed definition. The occupancy and density can measure congestion in space and time, respectively, as shown in Fig. 1. When traffic is incredibly congested, the value of the occupancy is close to 1. Still, it can not reach 1 as the Eq. (2) shows, due to vehicles would always keep space between each other (when the speed of all vehicles comes 0 km/h or vehicles move slowly on the road, the occupancy might be the same). It is not easy to estimate traffic congestion when solely considering occupancy to the severe congested condition. The value of density reaches its maximum as Eq. (1) indicates when all vehicles' speed comes 0 km/h on the road. However, it is difficult to quantify the traffic congestion of different roads when all the density values of different roads reach their maximum.

$$K = \frac{N}{Length} \tag{1}$$

$$R = \frac{\sum_{i=1}^{N} L_i}{Length} \times 100 \tag{2}$$

where *Length* is the actual length of a road, N is the total number of vehicles, L_i is the i-th length of the vehicle.

Our proposed definition of traffic congestion considers the occupancy and density simultaneously, which can be calculated via Eq. (3). Based on this definition, the congestion level can be normalized from 0 to 1, and the bigger, the more congested. 0 indicates that there is no vehicle on the road, while 1 represents extremely congested situations at which all vehicles' speed is 0 km/h.

$$Congestion = \frac{\sum_{i=1}^{N} f_i(L_i, i)}{Length} \tag{3}$$

where $Congestion \in [0, 1]$ is the calculated traffic congestion level, and the *Length* is the actual length of road sections in the images or videos with the same meaning as shown in Eq. (1) and Eq. (2). $f_i(L_i, i)$ is defined as:

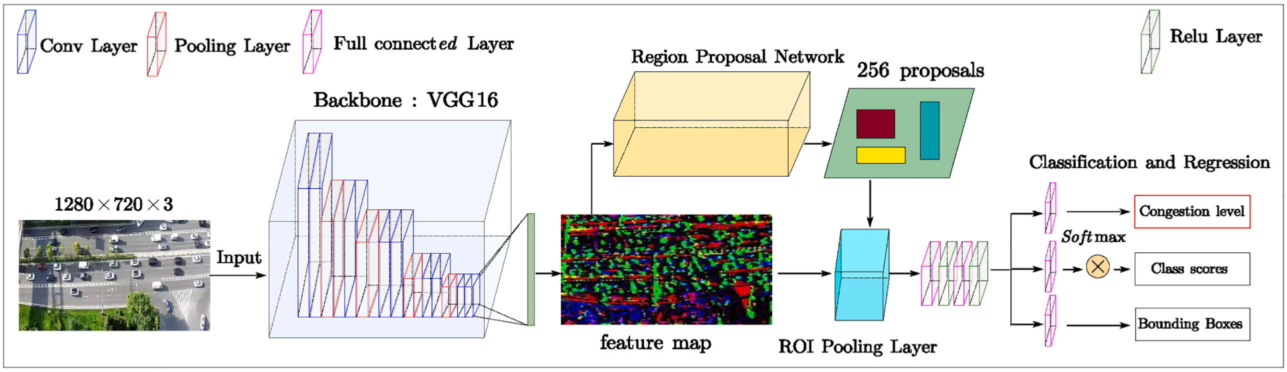


Fig. 2. The pipeline of the proposed traffic congestion estimation method based on Faster R-CNN (Ren et al., 2015). In the network, images are input, congestion level, class scores, and bounding boxes are outputs.

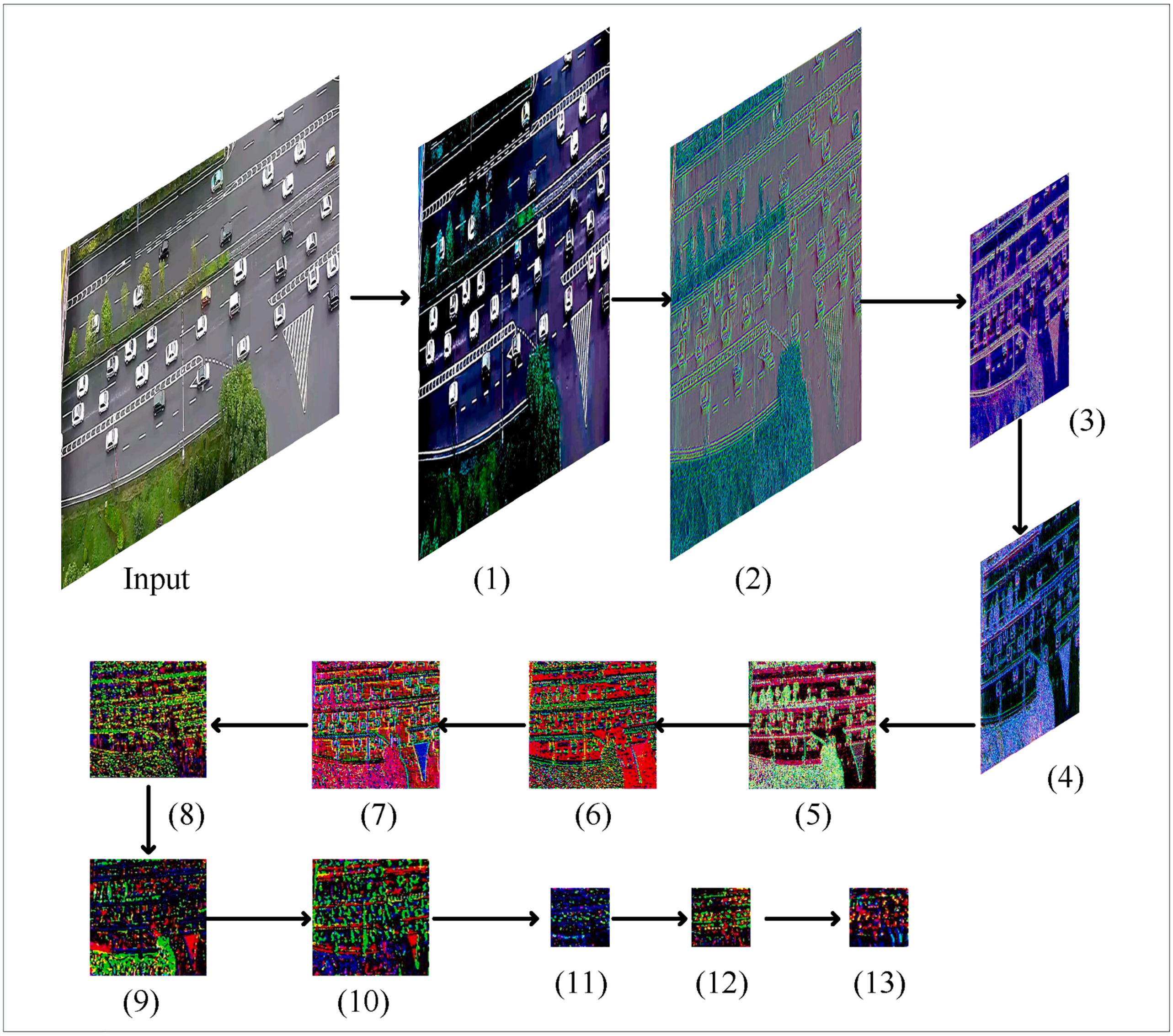


Fig. 3. Visualization of feature maps from 13 convolutional layers in VGG16.

$$f_i(L_i, i) = \varepsilon \times \frac{1}{N_{max}} + \gamma \times L_i \quad (4)$$

where ε and γ are weight coefficients that $\varepsilon \leq 1$, $\gamma \leq 1$, and $\varepsilon + \gamma = 1$. N_{max} is the maximum number of vehicles that can be accommodated on the road. It is clear that our proposed definition represents the occupancy when $\varepsilon = 0$ and $\gamma = 1$, while $\varepsilon = 1$ and $\gamma = 0$, it indicates the normalized density. The ε and γ values refer to the different capabilities and characteristics of road sections.

3.2. Framework

There are several CNN-based object detection models currently available, such as YOLO (Redmon et al., 2016), SSD (Liu et al., 2016) and Faster R-CNN (He et al., 2017). In this study, we choose Faster R-CNN model as the baseline model for traffic congestion estimation since Faster R-CNN (Ren et al., 2015) has an excellent performance in multi-object detection tasks. It is a representative two-stage CNN model that is widely used for object detection. On the feature map created by several convolution layers, it generates object-level proposals in the first step. The generated proposals with and without vehicles are then categorized. The proposals' location are further adjusted by regression in the second step and the objects' likelihood by classification are then obtained. Faster R-CNN's innovative feature is that it adopts Region Proposal Network (RPN) to produce region proposals, which makes a substantial increase in detection speed. Images, labels, and bounding boxes are taken as input into the original model. The possibilities of object classifications and the positions of objects are outputs. In our proposed model, a traffic congestion parameter layer is embedded in Faster R-CNN. The traffic congestion parameters are also taken as input to train the overall network, and congestion level is another output in the final. The training pipeline is shown in Fig. 6. To accomplish this, the overall framework of Faster R-CNN is adjusted and modified.

The pipeline of the proposed method is shown in Fig. 2. The framework includes two stages: the RPN that generates region proposals, and the other is Fast R-CNN that uses the generated region proposals for classification, location adjustment, and traffic congestion estimation. In the first stage, the VGG16, which has achieved the state-of-the-art results in the ImageNet Challenge (Simonyan & Zisserman, 2014), is used as a backbone to extract the feature map from input images, having 13 convolution layers and 3 pooling layers in the architecture. It can be seen from Fig. 6, both RPN and Fast R-CNN share the convolution layers for feature extraction to improve computation efficiency. When the image is taken as input in VGG16, the feature maps from 13 convolution layers are visible, as shown in Fig. 3. Based on these feature maps, RPN uses anchors of four scales (2^2, 4^2, 8^2, 16^2, 32^2, 48^2 pixels) and various aspect ratios (1:1, 1:2, 2:1) in a sliding window manner to generate multiple region proposals. RPN would provide the region proposals and region scores for Fast R-CNN; namely, RPN would tell the Fast R-CNN where objects are. Fast R-CNN then uses Softmax layers to binary classify those anchors whose Intersection-over-Union (IoU) overlaps with manually labeled bounding boxes are above 0.7 is set as positive, otherwise as negative. In the second stage, feature maps and proposals are taken as input to the ROI pooling layer. The generated feature maps are also taken as input to the next full connected layers and Relu layers after calculated by ROI pooling layer. Finally, the network would perform classification and regression to adjust the anchors and congestion parameters. During training, for each image, 256 anchors (128 as positive and 128 as negative) are initially entered from RPN into Roi pooling, and 2000 proposal samples can then be obtained after non-maximum suppression (NMS), which would be used for further classification and regression.

Three loss functions are included in our proposed traffic congestion estimation framework based on Faster R-CNN to compare the predictions with manually labeled ground truth. The first loss function $L_{cls\{V_i\}}$ is the object classification loss used to evaluate the classification misalignment. The second loss function $L_{reg\{T_j\}}$ represents the loss of the regression of traffic congestion, used to evaluated the misalignment of traffic congestion level. And the third loss function $L_{reg\{B_i\}}$ indicates the loss of bounding boxes regression, used to evaluate the region proposal location misalignment. $L_{cls\{V_i\}}$ and $L_{reg\{B_i\}}$ ensure traffic detection precision, while $L_{reg\{T_j\}}$ intuitively reflects congestion. $L_{cls\{V_i\}}$, $L_{reg\{B_i\}}$ and $L_{reg\{T_j\}}$ complement each other and influence each other. The overall loss function $L_{(\{V_i\},\{B_i\},\{T_j\})}$ of our proposed method contains the above three parts. They are characterized as follow:

$$L_{(\{V_i\},\{B_i\},\{T_j\})} = L_{cls\{V_i\}} + \omega L_{reg\{B_i\}} + \mu L_{reg\{T_j\}} \tag{5}$$

$$L_{cls\{V_i\}} = \frac{1}{N_{cls}} \sum_i - (logV_i^* V_i + \log[(1 - V_i^*)(1 - V_i)]) \tag{6}$$

$$L_{reg\{T_j\}} = \frac{1}{N_{lane}} \sum_j smooth_{L1}(T_j^* - T_j) \tag{7}$$

$$L_{reg\{B_i\}} = \frac{1}{N_{reg}} \sum_i V_i^* \times smooth_{L1}(B_i^* - B_i) \tag{8}$$

where the function of $smooth_{L1}(x)$ is defined as:

$$smooth_{L1}(x) = \begin{cases} 0.5x^2 if|x| < 1 \\ |x| - 0.5otherwise \end{cases} \tag{9}$$

where N_{cls} is the RPN mini-batch size (256), V_i is the probability of the i-th proposal to be a vehicle, and V_i^* is the manually labeled ground truth (the value is 1 for vehicle and 0 for non-vehicle in the proposal), N_{lane} is the number of lanes in an image, T_j is the predicted traffic congestion level of the j-th lane, T_j^* is the manually labeled ground truth of traffic congestion level. Here, both T_j and T_j^* represent traffic congestion level, with the same meaning of the left-term of Eq. (3), calculated by the right-term of Eq. (3). The $smooth_{L1}$ loss function defined as Eq. (9) is then used to calculate the error between T_j and T_j^*, thus $L_{reg\{T_j\}}$ can be calculated through Eq. (7), which will be further applied to the training process of the network. N_{reg} is the number of proposals (2000), and $smooth_{L1}$ is a kind of loss functions, which is more robust than L1 and L2 loss functions. B_i is the predicted bounding box location (4 parameterized coordinates of the bounding box) of the i-th proposal, B_i^* is the manually labeled ground truth bounding box location associated with the positive prediction. $L_{cls\{V_i\}}$ is the normalized loss for proposal classification, which is a classic two-class cross-entropy loss. For each anchor, the logarithmic value is calculated, and then summed up and divided by N_{cls} ($N_{cls} = 256$ in the training RPN stage, while $N_{cls} = 128$ in the Faster R-CNN training process). $L_{reg\{B_i\}}$ is the normalized regression loss for bounding box location adjustment. $L_{reg\{B_i\}}$ means that after calculating the actual offset relative to ground truth for each anchor, multiply it by V_i^*. As mentioned above, $V_i^* = 1$ when there is an object, otherwise $V_i^* = 0$, meaning that only the regression loss of foreground is calculated, and the background does not. $L_{reg\{T_j\}}$ represents the offset of predicted congestion value from the ground truth. Similarly, only the foreground is calculated, the background is not involved. ω and μ are balance weights. In our experiment, the ω and μ are set to 1. The four loss curves during the training processes are shown in Fig. 7.

Our proposed traffic congestion estimation framework is an end-to-end deep learning network trained by gradient descent in back propagation. The overall architecture of our proposed method has been displayed in Fig. 2.

4. Experiment

In this section, with the proposed framework and trained CNN model above, we conduct a series of experiments to compare our proposed method with four traditional algorithms and a deep learning method on traffic congestion estimation. The experimental results and corresponding analysis are depicted in this section.

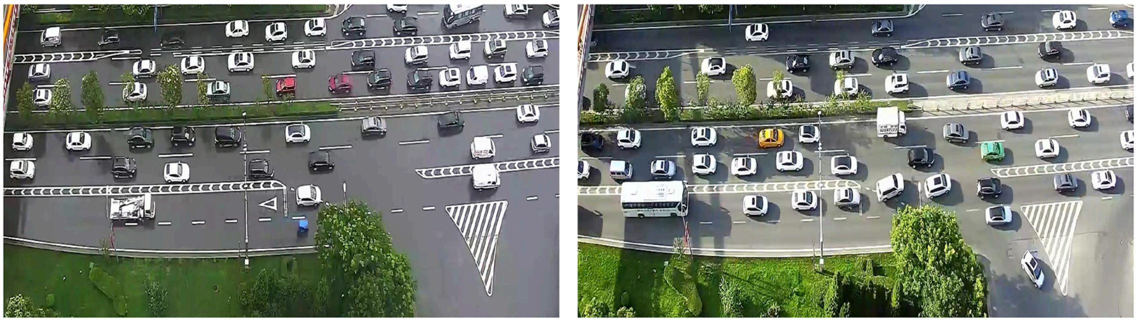
4.1. Dataset

In this study, the dataset is collected from a real traffic surveillance video located in the middle section of South Second Ring Road in Xi'an,

Table 1
Details of the collected dataset in the experiment. Two frames per second are obtained from surveillance videos.

Training Set	Status	Image Count	Vehicle Count	Duration	Date
Training-1	Rainy, Congested	700	34,770	6mins	06/16/2020
Training-2	Sunny, Congested	700	32,120	6mins	05/07/2019
Testing Set	Status	Image Count	Vehicle Count	Duration	Date
Scene-1	Rainy, Congested	600	31,173	5mins	06/16/2020
Scene-2	Rainy, Congested	600	33,488	5mins	06/16/2020
Scene-3	Sunny Congested	600	29,855	5mins	05/07/2019
Scene-4	Rainy, Non-congested	600	19,000	5mins	06/16/2020

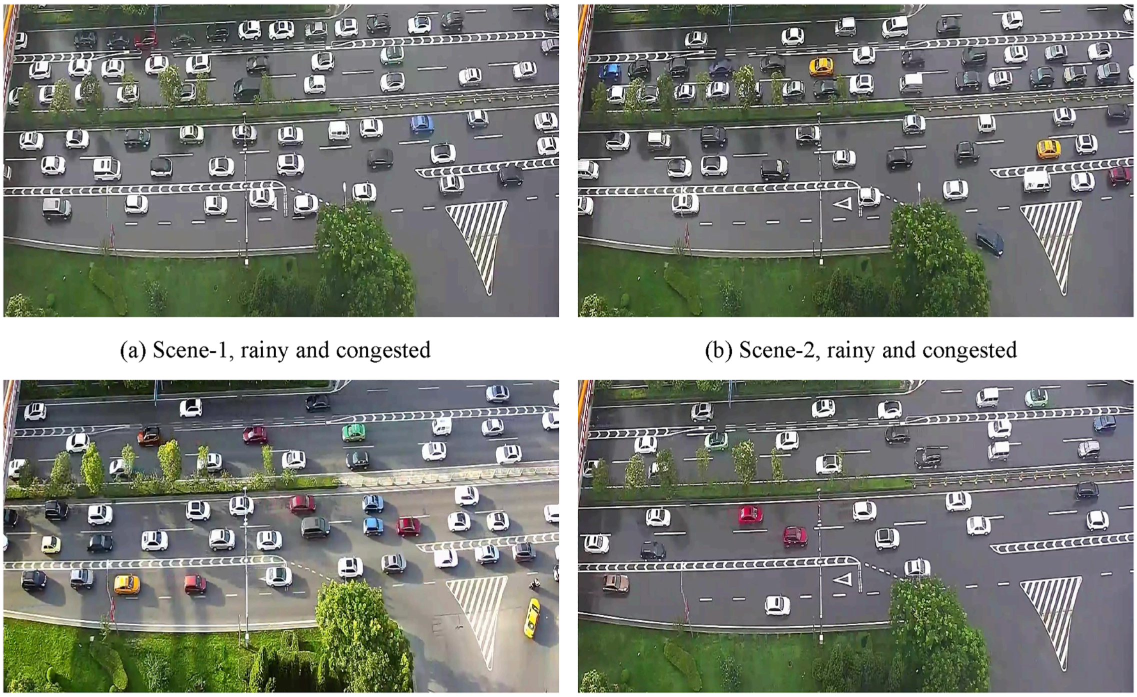
China. In our dataset, the image size is 1280 × 720 pixels. The dataset has 3800 manually labeled traffic images, which are divided into two sets: a training set and a testing set. The training set contains 1400 traffic images of two different periods (Training-1 and Training-2, 700 for each), where Training-1 (17:00 of 06/16/2020) is congested on rainy days, and Training-2 (15:13 of 05/07/2019) is congested on sunny days. There are total 66,890 vehicles in the training set. The testing set has 2400 traffic images of four different scenes (denoted as Scene-1, Scene-2, Scene-3, Scene-4, 600 for each). In the testing set, Scene-1 (17:32 of 06/16/2020) and Scene-2 (18:20 of 06/16/2020) are raining and congested, Scene-3 (18:36 of 05/07/2019) is sunny and congested, and Scene-4 (16:19 of 06/16/2020) is in the rain and free traffic condition. There are total 113,516 vehicles in the testing set. All these images are collected by two frames per second from the same video surveillance. In this study, the training set is used to train the proposed model, and each image of the testing set is manually labeled for performance evaluation only, which does not join the CNN training. The specific contents of the benchmark are shown in Table 1, and sample pictures of our two training sets and four testing sets are shown in Figs. 4 and 5, respectively.



(a) Training-1, rainy and congested

(b) Training-2, sunny and congested

Fig. 4. Sample pictures of our two training sets.



(a) Scene-1, rainy and congested

(b) Scene-2, rainy and congested

(c) Scene-3, sunny and congested

(d) Scene-4, rainy and non-congested

Fig. 5. Sample pictures of our four testing sets.

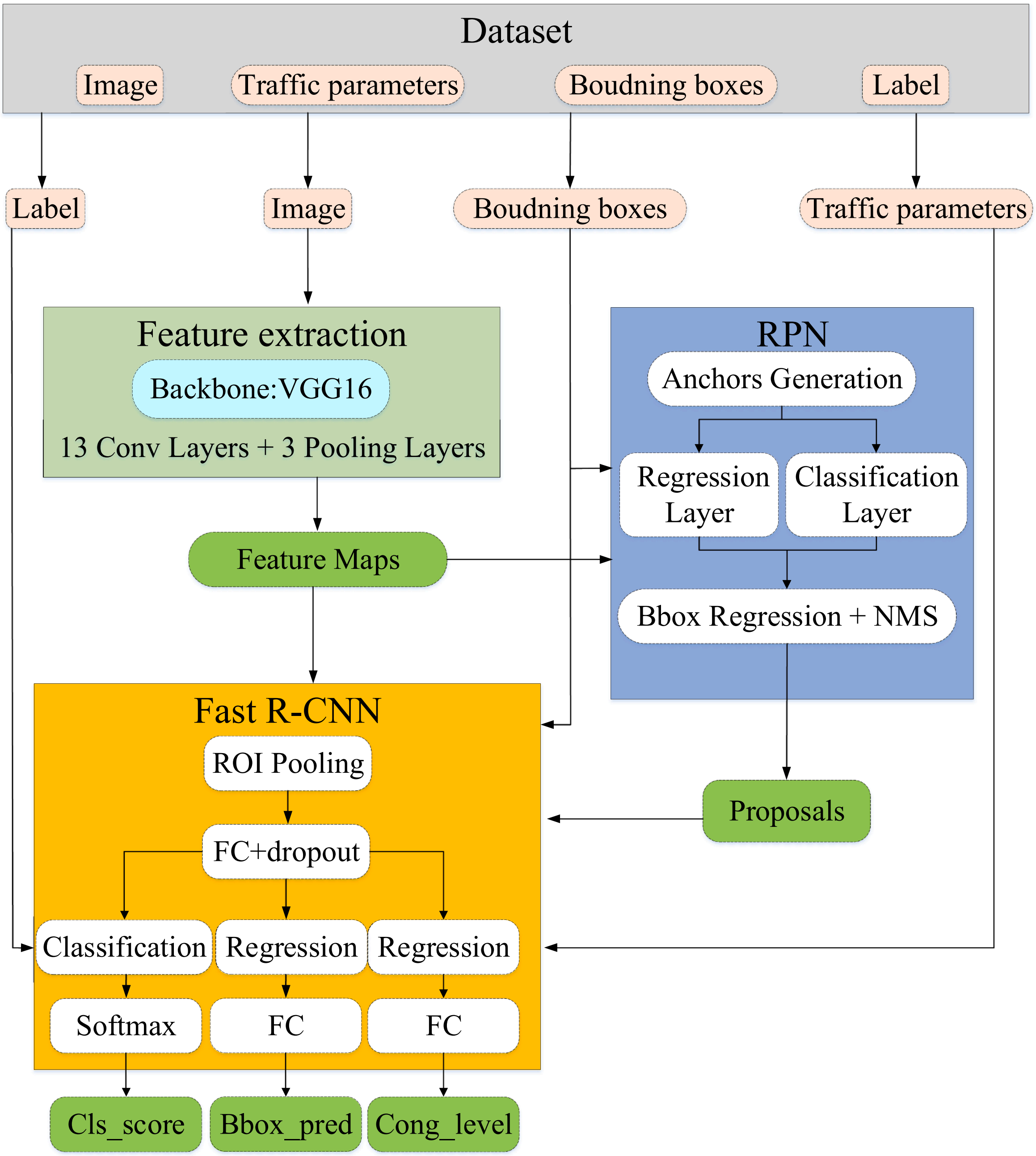


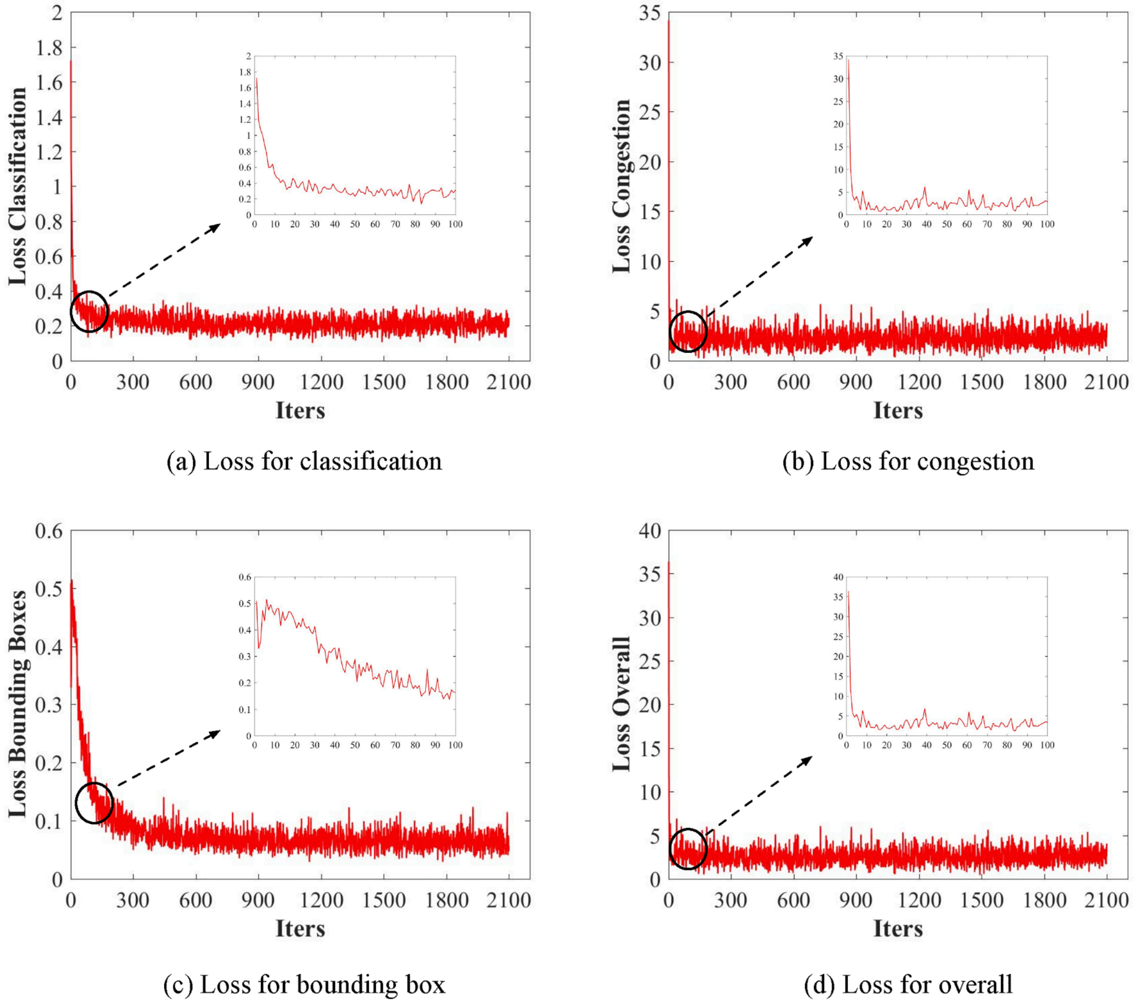
Fig. 6. The training pipeline of the proposed method. Note that FC is full connected layers.

4.2. Parameter settings

To validate the performance and accuracy of the proposed method, four traditional image processing algorithms based on background subtraction, i.e., the Mixture of Gaussians algorithm based on Adaptive Gaussian Mixture Model (MOG) (Zivkovic & Van Der Heijden, 2006), theMulti-Layer background subtraction algorithm(MultiLayer) (Yao & Odobez, 2007), the Gaussian Mixture Model background subtraction algorithm (DPGrimsonGMM) (Stauffer & Grimson, 1999), and Pixel-Based Adaptive Segmenter (PBAS) (Hofmann et al., 2012), are used as the comparative methods in our experiments. Besides, another deep learning method for vehicle detection, i.e., SSD (Liu et al., 2016) model is also introduced. Our proposed CNN-based method and SSD are trained on the same training set, including 1,400 images mentioned before. Finally, these six approaches are used in the four testing sets to estimate traffic congestion (Scene-1, Scene-2, Scene-3, Scene-4). In this study, all these 3,800 images are used directly without any self-defined pre-processing before applying them to our proposed CNN-based model and the comparative experiments.

In our experiments, five indicators are used to evaluate our proposed method and other five methods, which are Precision, Recall, F-measure, Number of False Positives per image (N_{FP}), and Number of False Negatives per image (N_{FN}), respectively. Precision is the proportion of relevant instances among the detected instances, Recall is the proportion of relevant instances detected over total instances, and F-measure is an overall indicator combining Precision and Recall together. These indicators are widely used in the performance assessment of detectors and can be described as the following equations:

$$Precision = \frac{TP}{TP + FP} \tag{10}$$



(a) Loss for classification

(b) Loss for congestion

(c) Loss for bounding box

(d) Loss for overall

Fig. 7. Four loss curves of our proposed method during 30 epoch training (20,970 iterations), including the normalized loss for proposal classification, the normalized regression loss for traffic congestion level, the normalized regression loss for bounding box location, and the overall loss.

$$Recall = \frac{TP}{TP + FN} \tag{11}$$

$$F_{measure} = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{12}$$

where *TP* is short for true positive, *FP* for false positive, and *FN* for false negative.

These six methods were implemented, and the experiments were conducted using Python3.6, OpenCV2.4, and PyTorch0.4 in Ubuntu16.04 system. During training, for our proposed CNN-based model and SSD model, we set the initial learning rate at 0.0001, the batch size as 4 images, and decayed with a factor of 0.9 of every ten epochs. The momentum is 0.9, and the training epoch is 30 in our experiments. All these experiments were conducted on a workstation with a CPU of 2.6 GHz, and a NVIDIA GTX 2080TI GPU with 12 GB memory.

4.3. Results

Most videos and image-based methods estimate traffic flow parameters based on vehicle detection currently. It is obvious that improved vehicle detection leads to more accurate traffic flow parameters in real-world applications. Therefore, the results of vehicle detection of these six methods will be compared in the performance evaluation in this study, which is intuitive that using vehicle detection to evaluate the congestion estimation performance. In our experiments, the surveillance video is captured from the roof of a 30-story building with a top view, about 100 m away from the ground. And a uniform threshold of 0.5 is determined for the IoU between the predicted bounding boxes and ground truth. Furthermore, to expand the adaptation of our proposed definition of traffic congestion to the data with different scales, we converted the pixel-level length to real-world length. Specifically, we have standard lane markings in our traffic images, six meters in length, occupying about 70 pixels. Thus we could easily calculate the real length of the road sections displayed in Figs. 4 and 5, as well as the real length of the vehicles. Based on the analysis above, we can conclude that as long as the vehicles and road markings in the captured images are clear enough, our proposed congestion estimation method would work well no matter what scale of the data is.

In Table 2, the results of the comprehensive detection performance evaluation are presented. This table illustrates that, in terms of these five metrics, vehicle detection using deep learning approaches is better than traditional methods of image processing. Four traditional image processing methods are performed in congested scenes (Scene1, Scene-2, and Scene-3), i.e., MOG, DPGrimsonGMM, MultiLayer, and PBAS, with a substantial decrease in detection performance results compared to sunny and free traffic conditions (Scene-4). The traditional image processing methods also have the worse detection performance on sunny days (Scene-3) under congested conditions than on rainy days (Scene-1 and Scene-2). The possible explanation may be that when congested, it is difficult to note the pixel change between successive frames; another is that vehicles' shadows on the road might be regarded as vehicles when

Table 2
Results of detection performance evaluation in the testing sets. On average of 4 testing subsets, the mean [Precision, F-measure] by different methods are: MOG [85.36%, 18.78%], DPGrimsonGMM [78.55%, 72.66%], MultiLayer [87.30%, 81.25%], PBAS [74.70%, 62.54%], SSD [98.31%, 95.16%], our proposed method [98.74%, 98.91%].

Scene-1						
	MOG	DPGrimsonGMM	MultiLayer	PBAS	SSD	Proposed
Precision	81.81%	76.63%	85.66%	76.07%	97.89%	**99.15%**
Recall	9.75%	69.17%	72.42%	57.05%	91.07%	**98.75%**
F-measure	17.43%	72.71%	78.48%	65.20%	94.36%	**98.95%**
N_{FP}	**0.12**	10.94	6.28	9.30	1.01	0.43
N_{FN}	46.80	15.98	14.30	22.27	4.63	**0.64**
Scene-2						
	MOG	DPGrimsonGMM	MultiLayer	PBAS	SSD	Proposed
Precision	90.18%	76.95%	85.70%	74.22%	97.95%	**98.69%**
Recall	9.10%	61.78%	66.34%	47.92%	89.84%	**99.16%**
F-measure	16.53%	68.54%	74.79%	58.24%	93.72%	**98.92%**
N_{FP}	**0.55**	10.32	6.17	9.29	1.05	0.73
N_{FN}	50.73	21.33	18.78	29.06	5.67	**0.465**
Scene-3						
	MOG	DPGrimsonGMM	MultiLayer	PBAS	SSD	Proposed
Precision	73.99%	74.32%	86.71%	62.06%	98.47%	**98.73%**
Recall	12.19%	61.13%	82.38%	33.45%	95.07%	**99.09%**
F-measure	20.93%	67.09%	84.49%	43.47%	96.74%	**98.91%**
N_{FP}	2.13	10.51	6.28	10.17	0.73	**0.63**
N_{FN}	43.69	19.33	8.76	33.11	2.45	**0.45**
Scene-4						
	MOG	DPGrimsonGMM	MultiLayer	PBAS	SSD	Proposed
Precision	95.47%	86.32%	91.18%	86.46%	**98.95%**	98.40%
Recall	11.32%	78.68%	83.69%	80.34%	92.92%	**99.35%**
F-measure	20.25%	82.32%	87.28%	83.29%	95.84%	**98.87%**
N_{FP}	**0.17**	3.94	2.56	3.98	0.31	0.51
N_{FN}	28.08	6.75	5.16	6.22	2.24	**0.20**

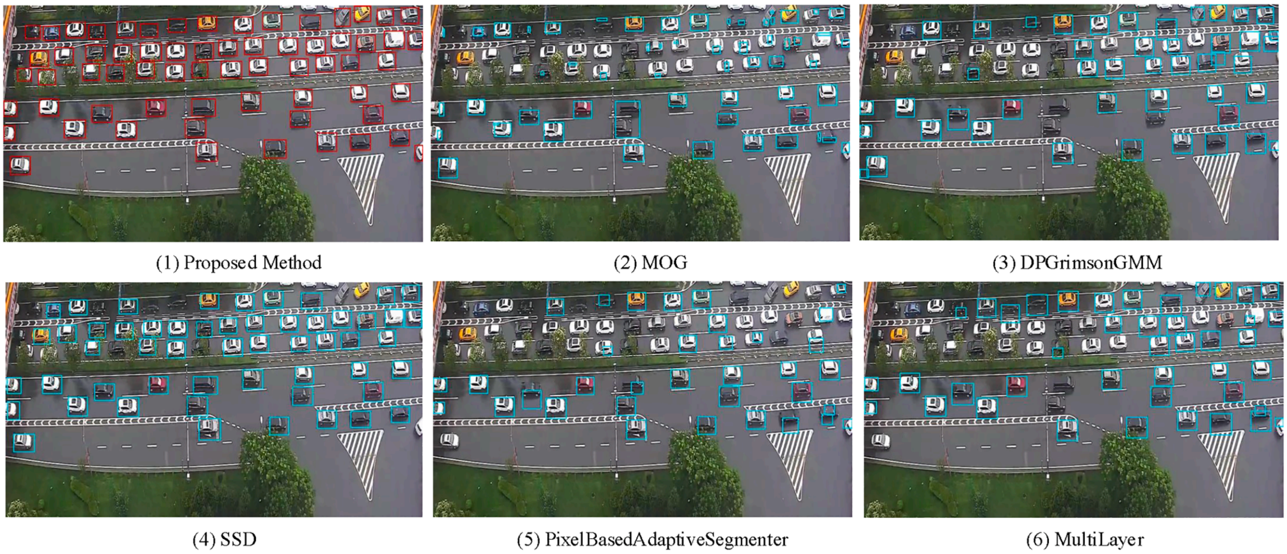
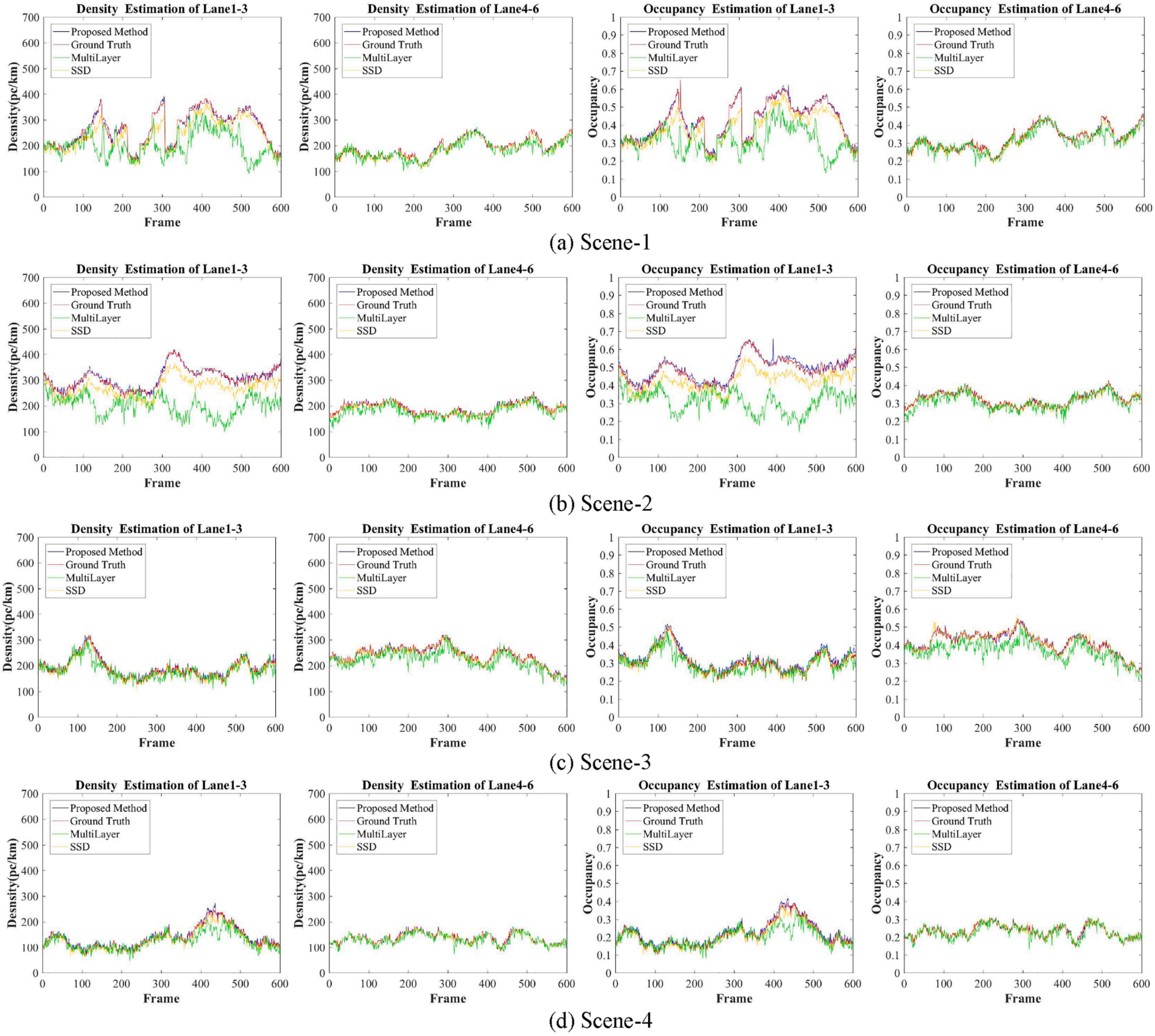

(1) Proposed Method
(2) MOG
(3) DPGrimsonGMM
(4) SSD
(5) PixelBasedAdaptiveSegmenter
(6) MultiLayer

Fig. 8. Vehicle detection results in Scene-2 with six methods. Red rectangle indicates our proposed method, and cyan rectangle represents the other five methods.

detected on sunny days. Although the recall of traditional methods is much smaller than that of deep learning techniques, MultiLayer has achieved the best results than the other three traditional methods. It also gets 87.30% and 81.25% as precision and F-measure on the average of four testing sets. Among all these six methods, our proposed method obtains the best performance on precision 98.74% and F-measure 98.91%.

From this table, it can be seen that the proposed method also achieves the best performance on *FN* among all the testing sets in this paper. It can also be observed intuitively from Fig. 8 that our proposed method has very few missed detections. In addition, our proposed method achieves the best results on sunny days (Scene-3) on *FP*, and still gets excellent results in rainy conditions (Scene-1, Scene-2, Scene-4). For the other three indicators including Precision, F-Measure, and Recall, our proposed method has achieved the best results without exception. It is evident that, in all congested, free, rainy, and sunny settings, the proposed method achieves the available and robust performance for vehicle detection. Additionally, we have to commit that the accuracy of



(a) Scene-1

(b) Scene-2

(c) Scene-3

(d) Scene-4

Fig. 9. Plots present the traffic density estimation and occupancy estimation in four Scenes. Traffic density of Lane1-3/Lane4-6 in a plot is the sum of three lanes in the same direction, while the occupancy of Lane13/Lane4-6 in a plot is three-lane in the same direction. Each scene has four result plots displayed in a row and from top to bottom: (a) Scene-1, (b) Scene-2, (c) Scene-3, (d) Scene-4. Proposed Method, MultiLayer, SSD, and Ground truth overlay in each of the 16 plots, where the blue curve represents the proposed method, red curve represents the ground truths, green curve is MultiLater method, and yellow curve is SSD method. Note that the duration of each plot is 5 min, two frames are obtained per second.

detection could be affected by some exceptions such as extreme weather or camera faults. However, the state of traffic flow is a continuously changing variable, it would not change with a big fluctuation. Therefore, the false detection holding a short time can be eliminated by a filtering algorithm such as the Kalman filter. For long-term false detection, for instance, caused by the foggy weather, the proposed method will definitely fall, which can be solved by non-vision sensors such as microwave Radar or loops.

The results of the visualized detections in Scene-2 are shown by these six methods in Fig. 8, where we only present the sample results of the detection in congested and rainy conditions. Our proposed method achieves the best performance compared to the other five approaches. The tests show that traditional image processing methods, i.e., MOG, DPGrimsonGMM, MultiLayer, and PBAS, are not stable and have limited performance in congested and lousy weather conditions. In contrast, in various traffic conditions, deep learning methods, i.e., SSD and the proposed method, are accurate and robust.

Traffic density and occupancy are usually used in practice to describe real-time traffic congestion over a while. Generally, after the vehicle detection results obtained by these traditional methods of image processing, or the deep learning models trained with traffic images, i.e., the SSD model, the estimation of traffic congestion could be simple and easy through existing knowledgeable methods. Our proposed method, unlike them, explicitly estimates traffic congestion once the model is trained, which is more effective and labor-saving. As seen in Fig. 8, it is clear that vehicles are traveling in two directions in our traffic images. Therefore, to approximate traffic density and occupancy (i.e., three lanes in each direction), six lanes (Lane1 to Lane6) are chosen from top to bottom in traffic images, where Lane1-3 indicates that traffic is moving towards left and Lane4-6 indicates that traffic is moving towards right.

Fig. 9 shows the traffic density estimation and occupancy estimation using these six methods in four testing scenes, where the ground truth of density and occupancy are calculated via the manually labeled images. Apparently, the proposed method curves (blue curves) and ground truth curves (red curves) are almost overlapped, while the MultiLayer method obtains the better detection performance in these traditional image

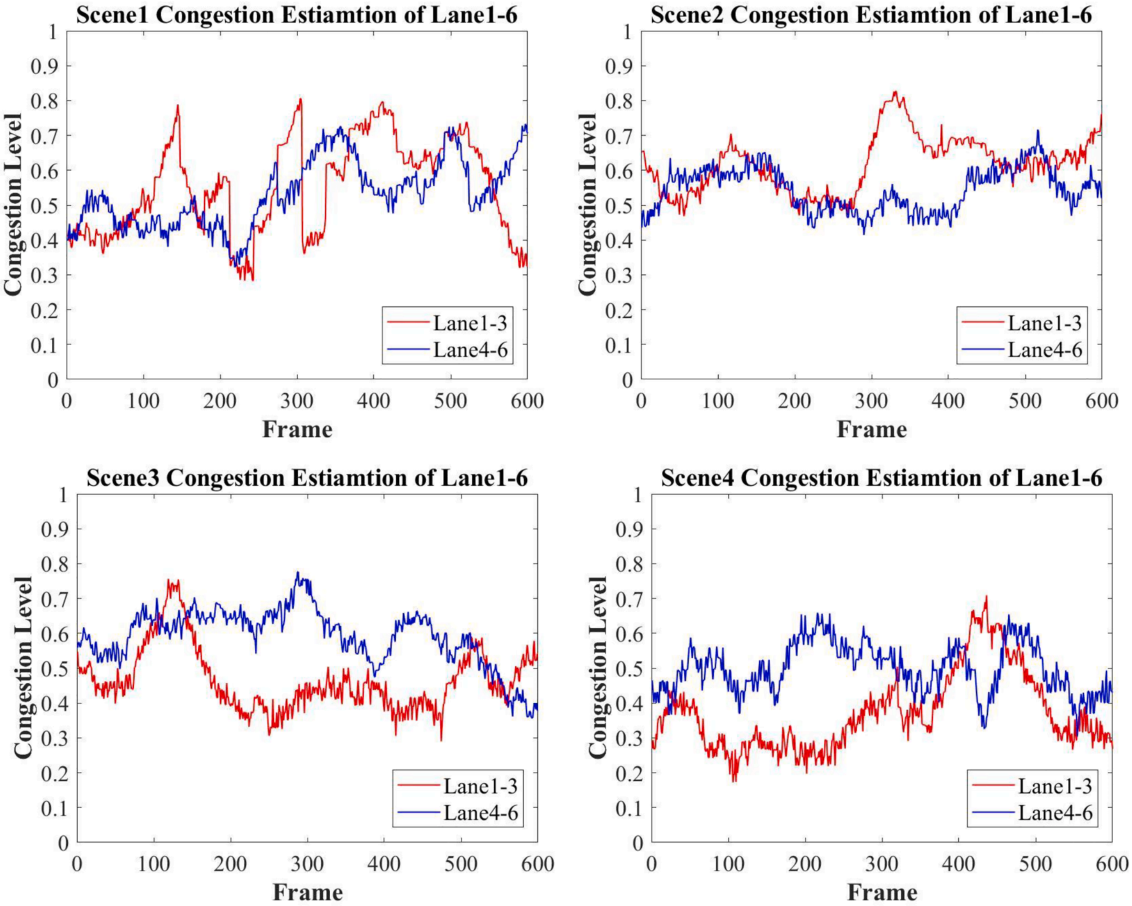


Fig. 10. Plots present the traffic congestion estimation based on our congestion definition in four testing scenes. The defined congestion level is from 0 to 1. Each scene has one plot, where red curve is the congestion estimation in Lane1-3, blue curve is the congestion estimation in Lane4-6.

processing methods. Specifically, the traffic density of Lane1-3 and Lane4-6 shown in Scene-1 are significantly different at the same time. The actual average density of Lane1-3 and Lane4-6 is 266.07 pc/km (passenger cars per kilometer in the same direction) and 192.74 pc/km. These can also be reflected in the same way with occupancy. The actual average occupancy of Lane1-3 and Lane4-6 are 0.4212 and 0.3246, respectively. The results of these four scenes in Fig. 9 show that Lane1-3 in Scene-2 is always congested, and the actual average occupancy is 0.4836. The minimum occupancy exists in Lane1-3 in Scene-4, which is 0.2163. The maximum average density is 306.03 pc/km in Lane1-3 in Scene-2, and the minimum average density is 135.96 pc/km in Lane1-3 in Scene-4. It is obvious that the maximum average traffic density and occupancy values are in Lane1-3 in Scene-2.

Traffic congestion estimation on the four testing sets based on our proposed definition of congestion is carried out, and the results are shown in Fig. 10, where ε and γ are set to 0.5. When "$\varepsilon = 0$ and $\gamma = 1$" holds, our proposed definition of congestion represents the occupancy, while "$\varepsilon = 1$ and $\gamma = 0$" holds, it indicates the normalized density. The results of occupancy are displayed in Fig. 9 on the right two columns, while the left two columns represent density without normalization. It can be seen that for different indicators describing congestion in the same scenario, their curve trends are the same, and the difference is the calculated value. Taking these three indicators for comparison, it is clear that the congestion definition we proposed can better reflect congestion, since when traffic congestion occurs and the estimated outcome is 0.7 or higher based on the proposed definition of traffic congestion, the occupancy is about 0.5 and the density is 320 pc/km or so.

To visualize the traffic congestion in our four testing sets, a heat map is finally performed and the results are shown in Fig. 11. It visually shows the estimated traffic congestion in each lane of the four scenes. The color bar of the heat map in Fig. 11 converts the color from cool blue to warm red, representing an increase in the corresponding rate of traffic congestion level from 0 to 1. In the color shift of the heat map, the six lanes' traffic congestion levels are mirrored. In Scene-1, Lane1-3 shows congested in the middle and second half of the video, while Lane4-6 shows less congested. Scene-2 could be spotted that severe congestion came out on Lane1, Lane2, and Lane3. Most of the time, they are in a more congested condition, while the other three lanes could be considered free, which is also in line with our expected results. The congestion estimation of Scene-3 is similar to that of Scene-1 and Scene-2, which is consistent with video shooting yet. Scene-4 is shot on the same road section during a non-congested period, which is well reflected from the heat map's cool color.

5. Conclusion

In this paper, a unified and accurate definition of traffic congestion is proposed to quantify the traffic congestion, and then a framework of traffic congestion estimation for video surveillance using the CNN model is proposed. For traffic congestion estimation, a modified Faster R-CNN model with a traffic parameter layer integrated, can be used to directly estimate traffic congestion. Based on our four different testing subsets collected including 2400 traffic images, our proposed method achieves the best available performance and robustness compared with the other four traditional image processing methods and a deep learning method introduced in our experiments.

In the future, we will concentrate on the following topics. First, more vehicle attributes will be considered to construct the congestion-related

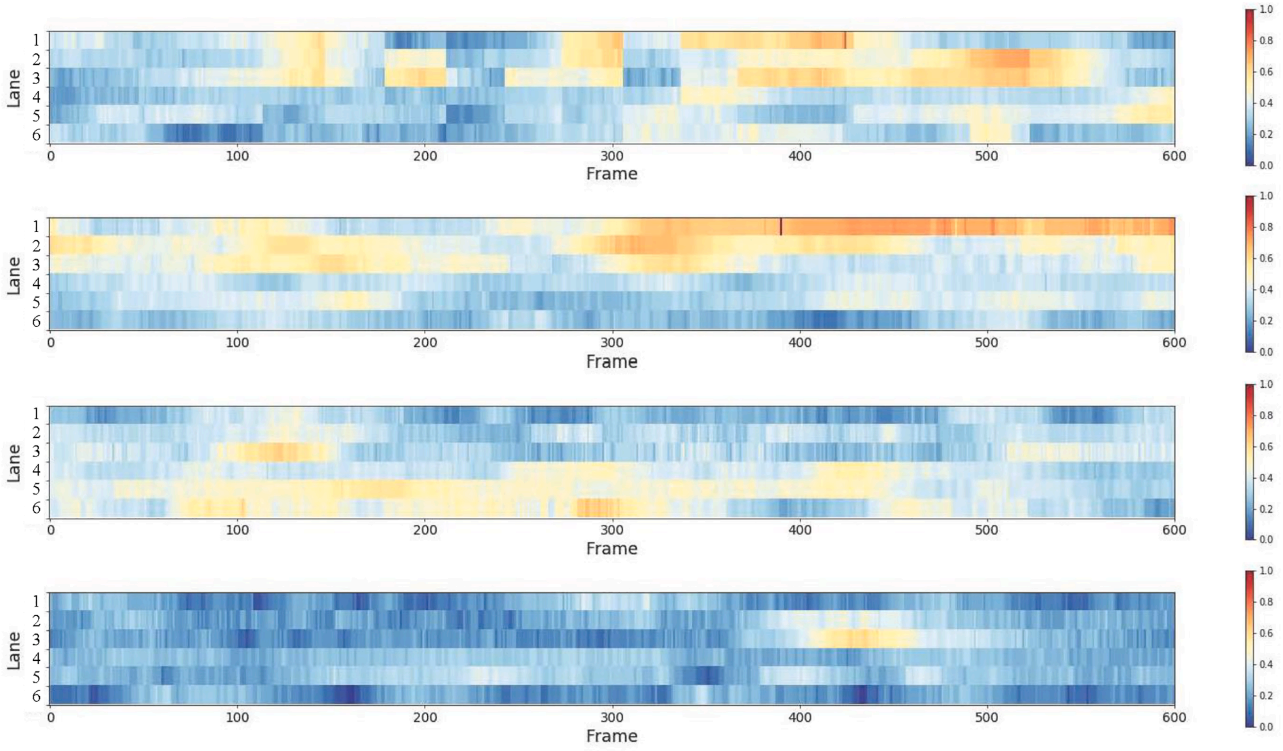


Fig. 11. Heat maps show the results of traffic congestion estimation based on our definition in four scenes. From top to bottom: Scene-1, Scene-2, Scene-3, Scene-4. Note that the duration of each scene is 5 min, two frames are obtained per second.

estimation methods, such as velocity and driving behavior. Second, the present study concentrates on estimating traffic congestion. We plans to use deep learning methods to predict the upcoming traffic congestion based on the historical and current data from surveillance cameras.

CRediT authorship contribution statement

Ying Gao: Data curation, Methodology, Software, Writing - original draft. **Jinlong Li:** Data curation, Methodology, Software, Writing - original draft. **Zhigang Xu:** Methodology, Writing - review & editing, Supervision. **Zhangqi Liu:** Data curation, Methodology. **Xiangmo Zhao:** Methodology, Writing - review & editing, Supervision. **Jianhua Chen:** Methodology.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by National Key Research and Development Program of China (No. 2019YFB1600100), National Natural Science Foundation of China (No. 61973045), Shaanxi Province Key Development Project (No. S2018-YF-ZDGY-0300), Fundamental Research Funds for the Central Universities (No. 300102248403), Joint Laboratory of Internet of Vehicles sponsored by Ministry of Education and China Mobile (No. 213024170015), Application of Basic Research Project for National Ministry of Transport (No. 2015319812060).

References

Ahonen, T., Hadid, A., & Pietikainen, M. (2006). Face description with local binary patterns: Application to face recognition. *IEEE transactions on Pattern Analysis and Machine Intelligence, 28*(12), 2037–2041.

Bacon, J., Bejan, A. I., Beresford, A. R., Evans, D., Gibbens, R. J., & Moody, K. (2011). Using real-time road traffic data to evaluate congestion. In *Dependable and Historic Computing* (pp. 93–117). Springer.

Bautista, C. M., Dy, C. A., & Mañalac, M. I., Orbe, R. A., and Cordel, M.. (2016). In *Convolutional neural network for vehicle detection in low resolution traffic videos* (pp. 277–281). IEEE.

Borden, J. (1993). Hicomp report: Statewide highway congestion monitoring program. *California Department of.* Transportation.

Bouwmans, T., El Baf, F., & Vachon, B. (2008). Background modeling using mixture of gaussians for foreground detection-a survey. *Recent patents on computer science, 1*(3), 219–237.

Calderoni, L., Maio, D., & Rovis, S. (2014). Deploying a network of smart cameras for traffic monitoring on a "city kernel". *Expert Systems with Applications, 41*(2), 502–507.

Cao, X., Lan, J., Yan, P., & Li, X. (2012). Vehicle detection and tracking in airborne videos by multi-motion layer analysis. *Machine Vision and Applications, 23*(5), 921–935.

Chen, Y., & andWu, Q.. (2015). In *Moving vehicle detection based on optical flow estimation of edge* (pp. 754–758). IEEE.

Chung, J., & Sohn, K. (2017). Image-based learning to measure traffic density using a deep convolutional neural network. *IEEE Transactions on Intelligent Transportation Systems, 19*(5), 1670–1675.

Dai, J., Li, Y., He, K., & Sun, J. (2016). R-fcn: Object detection via region-based fully convolutional networks. *Advances in neural information processing systems*, 379–387.

Dallalzadeh, E., Guru, D., & Harish, B. (2013). Symbolic classification of traffic video shots. In *Advances in Computational Science, Engineering and Information Technology* (pp. 11–22). Springer.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). In *Imagenet: A large-scale hierarchical image database* (pp. 248–255). IEEE.

Ding, D., Tong, J., & Kong, L. (2020). A deep learning approach for quality enhancement of surveillance video. *Journal of Intelligent Transportation Systems, 24*(3), 304–314.

Gupte, S., Masoud, O., Martin, R. F., & Papanikolopoulos, N. P. (2002). Detection and classification of vehicles. *IEEE Transactions on Intelligent Transportation Systems, 3*(1), 37–47.

Han, S., Han, Y., & Hahn, H. (2009). Vehicle detection method using haar-like feature on real time system. *World Academy of Science, Engineering and Technology, 59*, 455–459.

He, K., Gkioxari, G., & Dolĺar, P., and Girshick, R.. (2017). Mask r-cnn. In *In IEEE International Conference on Computer Vision* (pp. 2961–2969).

Hofmann, M., Tiefenbacher, P., & Rigoll, G. (2012). Background segmentation with feedback: The pixel-based adaptive segmenter. In *IEEE computer society conference on computer vision and pattern recognition workshops* (pp. 38–43). IEEE.

Hsieh, J.-W., Chen, L.-C., & Chen, D.-Y. (2014). Symmetrical surf and its applications to vehicle detection and vehicle make and model recognition. *IEEE Transactions on Intelligent Transportation Systems, 15*(1), 6–20.

Hu, S., Wu, J., & Xu, L. (2012). Real-time traffic congestion detection based on video analysis. *Journal of Information and Computational Science, 9*(10), 2907–2914.

Ke, R., Li, Z., Tang, J., Pan, Z., & Wang, Y. (2018). Real-time traffic flow parameter estimation from uav video based on ensemble classifier and optical flow. *IEEE Transactions on Intelligent Transportation Systems, 20*(1), 54–64.

Keller, J. M., Gray, M. R., & Givens, J. A. (1985). A fuzzy k-nearest neighbor algorithm. *IEEE transactions on systems, man, and cybernetics, 4*, 580–585.

Kong, J., Zheng, Y., Lu, Y., & Zhang, B. (2007). *A novel background extraction and updating algorithm for vehicle detection and tracking* (volume 3,, 464–468.

Kong, X., Xu, Z., Shen, G., Wang, J., Yang, Q., & Zhang, B. (2016). Urban traffic congestion estimation and prediction based on floating car trajectory data. *Future Generation Computer Systems, 61*, 97–107.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Communications of the ACM, 60*(6), 84–90.

Li, J., Xu, Z., Fu, L., Zhou, X., & Yu, H. (2021). Domain adaptation from daytime to nighttime: A situation-sensitive vehicle detection and traffic flow parameter estimation framework. *Transportation Research Part C: Emerging Technologies, 124*, Article 102946.

Li, S., Yu, H., Zhang, J., Yang, K., & Bin, R. (2013). Video-based traffic data collection system for multiple vehicle types. *IET Intelligent Transport Systems, 8*(2), 164–174.

Li, X., Li, Z., Han, J., & Lee, J.-G. (2009). In *Temporal outlier detection in vehicle traffic data* (pp. 1319–1322). IEEE.

Liu, H. X., & Sun, J. (2014). Length-based vehicle classification using event-based loop detector data. *Transportation Research Part C: Emerging Technologies, 38*, 156–166.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). In *Ssd: Single shot multibox detector* (pp. 21–37). Springer.

Lomax, T. J. (1997). *Quantifying congestion, Number 398*. Transportation Research Board.

Lozano, A., Manfredi, G., & Nieddu, L. (2009). An algorithm for the recognition of levels of congestion in road traffic problems. *Mathematics and Computers in Simulation, 79* (6), 1926–1934.

Ma, X., Tao, Z., Wang, Y., Yu, H., & Wang, Y. (2015). Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transportation Research Part C: Emerging Technologies, 54*, 187–197.

Mu, K., Hui, F., & Zhao, X. (2016). Multiple vehicle detection and tracking in highway traffic surveillance video based on sift feature matching. *Journal of Information Processing Systems, 12*(2).

Oliva, A., Torralba, A. B., Gúerin-Dugúe, A., and H́erault, J. (1999). Global semantic classification of scenes using power spectrum templates. In Challenge of image retrieval, pages 1–11.

Pan, X., Guo, Y., & Men, A. (2010). *Traffic surveillance system for vehicle flow detection* (volume 1,, 314–318.

Papandreou, G., & Maragos, P. (2006). Multigrid geometric active contour models. *IEEE transactions on image processing, 16*(1), 229–240.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *IEEE conference on Computer Vision and Pattern Recognition*, 779–788.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 91–99.

Roess, R. P. and Prassas, E. S. (2014). The highway capacity manual: a conceptual and research history.

Rybski, P. E., Huber, D., Morris, D. D., & Hoffman, R. (2010). In *Visual classification of coarse vehicle orientation using histogram of oriented gradients features* (pp. 921–928). IEEE.

Sadollah, A., Gao, K., Zhang, Y., Zhang, Y., & Su, R. (2019). Management of traffic congestion in adaptive traffic signals using a novel classification-based approach. *Engineering Optimization, 51*(9), 1509–1528.

Simoncini, M., Taccari, L., Sambo, F., Bravi, L., Salti, S., & Lori, A. (2018). Vehicle classification from low-frequency gps data with recurrent neural networks. *Transportation Research Part C: Emerging Technologies, 91*, 176–191.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for largescale image recognition. arXiv preprint arXiv:1409.1556.

Song, B., & HAN, L.-Q., ZHONG, Y.-X., and WANG, X.-J.. (2011). All-day traffic states recognition system without vehicle segmentation. *The Journal of China Universities of Posts and Telecommunications, 18*, 1–11.

Stauffer, C., & Grimson, W. E. L. (1999). *Adaptive background mixture models for real-time tracking, No PR00149), volume 2*, 246–252.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., … Rabinovich, A. (2015). Going deeper with convolutions. *IEEE conference on Computer Vision and Pattern Recognition*, 1–9.

Tahmid, T., & Hossain, E. (2017). In *Density based smart traffic control system using canny edge detection algorithm for congregating traffic information* (pp. 1–5). IEEE.

Venkatesvara Rao, N., Venkatavara Prasad, D., & Sugumaran, M. (2018). Realtime video object detection and classification using hybrid texture feature extraction. *International Journal of Computers and Applications*, 1–8.

Wang, Q., Wan, J., & Yuan, Y. (2018). Locality constraint distance metric learning for traffic congestion detection. *Pattern Recognition, 75*, 272–281.

Wei, L., & Hong-ying, D. (2016). Real-time road congestion detection based on image texture analysis. *Procedia engineering, 137*, 196–201.

Willis, C., Harborne, D., Tomsett, R., and Alzantot, M. (2017). A deep convolutional network for traffic congestion classification. In Proc NATO IST-158/RSM-010 Specialists' Meeting on Content Based Real-Time Analytics of Multi-Media Streams, pages 1–11.

Wu, Y.-J., Chen, F., Lu, C.-T., & Yang, S. (2016). Urban traffic flow prediction using a spatio-temporal random effects model. *Journal of Intelligent Transportation Systems, 20*(3), 282–293.

Yao, J., & Odobez, J.-M. (2007). In *Multi-layer background subtraction based on color and texture* (pp. 1–8). IEEE.

Yuan, Y., Wan, J., & Wang, Q. (2016). Congested scene classification via efficient unsupervised feature learning and density estimation. *Pattern Recognition, 56*, 159–169.

Zhao, J., Xu, H., Liu, H., Wu, J., Zheng, Y., & Wu, D. (2019a). Detection and tracking of pedestrians and vehicles using roadside lidar sensors. Transportation Research Part C: Emerging Technologies, 100:68–87.

Zhao, X., Dawson, D., Sarasua, W. A., & Birchfield, S. T. (2017). Automated traffic surveillance system with aerial camera arrays imagery: Macroscopic data collection with vehicle tracking. *Journal of Computing in Civil Engineering, 31*(3), 04016072.

Zhao, Z.-Q., Zheng, P., Xu, S.-t., & Wu, X. (2019b). Object detection with deep learning: A review. IEEE transactions on neural networks and learning systems, 30(11): 3212–3232.

Zivkovic, Z., & Van Der Heijden, F. (2006). Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recognition Letters, 27*(7), 773–780.