Transportation Research Part C 124 (2021) 102946

Contents lists available at ScienceDirect

Transportation Research Part C

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

Domain adaptation from daytime to nighttime: A situation-sensitive vehicle detection and traffic flow parameter estimation framework

Jinlong Li [a,d], Zhigang Xu [a,*], Lan Fu [b], Xuesong Zhou [c], Hongkai Yu [d,*]

[a] School of Information Engineering, Chang'an University, Xi'an 710064, China
[b] Department of Computer Science and Engineering, University of South Carolina, SC 29201, USA
[c] School of Sustainable Engineering and the Built Environment, Arizona State University, AZ 85287, USA
[d] Department of Electrical Engineering and Computer Science, Cleveland State University, Cleveland, OH 44115, USA

ARTICLE INFO

Keywords:
Vehicle detection
Deep learning
Domain adaptation
Traffic flow parameter

ABSTRACT

Vehicle detection in traffic surveillance images is an important approach to obtain vehicle data and rich traffic flow parameters. Recently, deep learning based methods have been widely used in vehicle detection with high accuracy and efficiency. However, deep learning based methods require a large number of manually labeled ground truths (bounding box of each vehicle in each image) to train the Convolutional Neural Networks (CNN). In the modern urban surveillance cameras, there are already many manually labeled ground truths in daytime images for training CNN, while there are little or much less manually labeled ground truths in nighttime images. In this paper, we focus on the research to make maximum usage of labeled daytime images (Source Domain) to help the vehicle detection in unlabeled nighttime images (Target Domain). For this purpose, we propose a new situation-sensitive method based on Faster R-CNN with Domain Adaptation (DA) to improve the vehicle detection at nighttime. Furthermore, a situation-sensitive traffic flow parameter estimation method is developed based on the traffic flow theory. We collected a new dataset of 2,200 traffic images (1,200 for daytime and 1,000 for nighttime) of 57,059 vehicles to evaluate the proposed method for the vehicle detection. Another new dataset with three 1,800-frame daytime videos and one 1,800-frame nighttime video of about 260 K vehicles was collected to evaluate and show the estimated traffic flow parameters in different situations. The experimental results show the accuracy and effectiveness of the proposed method.

1. Introduction

In recent years, more and more traffic video surveillance systems are installed in the city and multiple cameras are installed on one autonomous vehicle, which can provide more detailed traffic information, like the vehicle detection results introduced by Tian et al. (2015), Yang and Pun-Cheng (2018), Mertz et al. (2020), Hale et al. (2020). As described in Wan et al. (2014), Babari et al. (2012), Ma and Qian (2019), Li et al. (2020), vehicles detection from these traffic images is important to the intelligent transportation system, safety monitoring, traffic control, autonomous driving, and trajectory data-based traffic flow studies.

In computer vision, object detection aims to discover the location of the interested objects based on feature extraction and

* Corresponding authors.
E-mail addresses: xuzhigang@chd.edu.cn (Z. Xu), h.yu19@csuohio.edu (H. Yu).

https://doi.org/10.1016/j.trc.2020.102946
Received 4 June 2020; Received in revised form 20 December 2020; Accepted 21 December 2020
Available online 16 January 2021

recognition, i.e., vehicles, from one single image. Some traditional methods by Abdulrahim and Salam (2016), Coifman et al. (1998) use a variety of image processing algorithms in vehicle detection. With the recent rapid development of deep learning, many Convolutional Neural Network (CNN) based methods are widely used for vehicle detection as introduced in Wang et al. (2019b). However, deep learning based methods require a large number of manually labeled ground truths (manually annotated bounding box of each vehicle in each image) to train the CNN. Although the number of training sets can be expanded by data augmentation as that in Guo et al. (2019), including flipping, cropping, and scaling operations, there are still a large number of diverse images that need to be manually labeled. Manual labeling by human is labor-intensive and time-consuming, so it is necessary to make fully use of labeled existing data to help unlabeled new data.

In the modern urban surveillance cameras, there are already many manually labeled ground truths in daytime images for training CNN, while there are little or much less manually labeled ground truths in nighttime images. In this paper, we focus on the research to make maximum usage of labeled daytime images (Source Domain) to help the vehicle detection in unlabeled nighttime images (Target Domain). In our experiment, directly applying the CNN model trained on the Source Domain to detect the vehicles on the Target Domain shows relatively low performance. This is because of the domain distribution discrepancy of Source and Target Domains. Intuitively, the nighttime images are quite different with the daytime images: dark environment, changed road light condition, more blurred image, various road reflection, etc.

In order to reduce the domain distribution discrepancy of Source and Target Domains, we propose to use CNN with Domain Adaptation (DA) for the situation-sensitive vehicle detection which is robust in both daytime and nighttime conditions. DA is a representative method in transfer learning. Generally, when the data distribution of the source domain and target domain are different, but the task is consistent, DA can better use the combined information of the two domains to improve the task performance on the target domain described by Lin et al. (2016). The proposed vehicle detection problem based on DA is shown in Fig. 1. The CNN model used in the proposed method for vehicle detection is Faster R-CNN by Ren et al. (2015), due to its advanced accuracy and speed in object detection. The DA method used in the proposed method is actually a style transfer between daytime images and nighttime images by Generative Adversarial Networks (GAN), where the unpaired translation method CycleGAN proposed by Zhu et al. (2017) is used for this style transfer.

To test the proposed method, we collected a new dataset, named as Daytime and Nighttime Vehicle Detection (DNVD) dataset, that includes 1,200 daytime images and 1,000 nighttime images of 57,059 vehicles by a real traffic surveillance camera. We manually labeled each vehicle in the daytime images for CNN training and manually labeled each vehicle in the nighttime images for performance evaluation. We compared the proposed method with several traditional image processing based and deep learning based object detection methods, and the proposed method achieved the best F-measure and mAP performance for nighttime vehicle detection. The

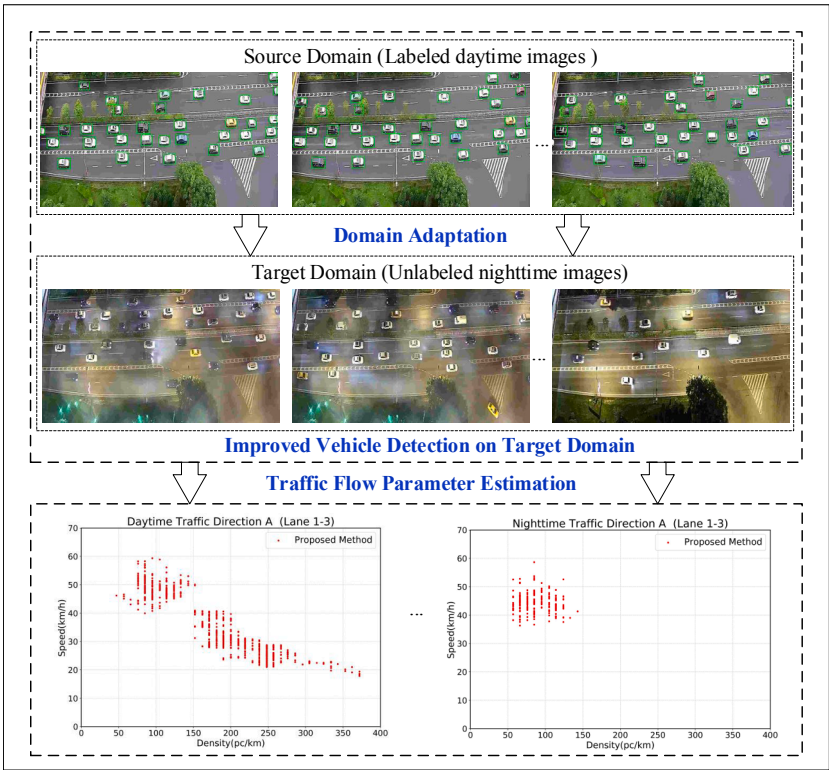


Fig. 1. The proposed situation-sensitive vehicle detection and traffic flow parameter estimation framework with Domain Adaptation from the labeled daytime images (Source Domain) to unlabeled nighttime images (Target Domain).

experiment results also show that the proposed method with DA can reduce the distribution difference of two domains and improve the performance of vehicle detection in the nighttime. Using the traffic flow theory, the proposed method can be extended for the situation-sensitive traffic flow parameter estimation in both daytime and nighttime situations. To test the proposed traffic flow parameter estimation method, we collected another new dataset (three 1,800-frame daytime videos and one 1,800-frame nighttime video) of about 260 K vehicles in total to evaluate the traffic flow parameter estimation performance. On summary, the main contributions of this paper are as follows:

- A new deep learning based pipeline for the situation-sensitive vehicle detection in daytime and nighttime with only labeled daytime images is proposed. Specifically, a Faster R-CNN model is used for vehicle detection during daytime and a new Faster R-CNN model with DA is proposed to make better usage of daytime data for vehicle detection during nighttime, where style transfer is used to realize the domain adaptation from the labeled daytime images (Source Domain) to unlabeled nighttime images (Target Domain).
- A new framework for the situation-sensitive traffic flow parameter estimation in daytime and nighttime is proposed. The proposed framework could analyze and compare daytime and nighttime traffic flow in the same location with meaningful visualizations. To the best of our knowledge, this paper is the first work for the nighttime traffic flow parameter estimation while only using the labeled daytime images to train the deep learning model. With the detected vehicles, the traffic flow parameter estimation might be straightforward, but the proposed method (Faster R-CNN+DA) could collect more accurate traffic flow parameters during nighttime than the original Faster R-CNN.
- Two new datasets for this research are collected and manually labeled for vehicle detection and traffic flow parameter estimation respectively. One new dataset for vehicle detection contains 1,200 daytime images and 1,000 nighttime images of 57,059 vehicles. Another new dataset with three 1,800-frame daytime videos and one 1,800-frame nighttime video of about 260 K vehicles was collected to evaluate and show the estimated traffic flow parameters.

2. Related work

2.1. Computer vision based vehicle detection

We consider vehicle detection from images or video based on computer vision. Generally, there are currently two approaches to obtain effective extraction of vehicle information from images or video. The first approach is to obtain moving objects (foreground) of the traffic scene, while the static part (background) of the traffic scene is separated introduced in Tian et al. (2011). The separation between background and foreground are usually by detecting the changes. Some studies proposed by Kamijo et al. (2000), Li et al. (2009) segment moving objects using space-time difference, and some other methods in Kong et al. (2007), Mandellos et al. (2011), Zhou et al. (2007), Gupte et al. (2002) use background subtraction algorithms to extract moving objects. These methods can be effectively applied to daytime traffic scenarios with good light conditions. The second approach is a feature extraction method from the object appearance, mainly using the features of color, texture and shape, which can detect stationary objects in images or video described in Lowe (1999), Tian et al. (2014). More complex features have been used in vehicle detection such as local symmetry edge operators by Agarwal et al. (2004), Scale Invariant Feature Transformation (SIFT) by Mu et al. (2016), Speeded up Robust Features (SURF) by Hsieh et al. (2014), Histogram of Oriented Gradient (HOG) by Rybski et al. (2010) and Haar-like features by Han et al. (2009). Based on feature extraction, some large-scale crowded objects with similar appearance can be detected as described in Yu et al. (2016). Recently, deep learning based CNN methods by Dong et al. (2015), Rezaei et al. (2015), Ke et al. (2018), Bautista et al. (2016), Long et al. (2015), Audebert et al. (2017), Guo et al. (2018) are widely used for vehicle detection, which have robust and advanced vehicle detection performances.

2.2. Computer vision based vehicle detection in nighttime

Vehicle detection in nighttime is very challenging because of the light conditions, dark environment, road reflection, blurred image in the nighttime. Most of the existing methods may be unreliable for handling nighttime traffic conditions as described in Chen et al. (2010). Beymer et al. (1997) proposed a vehicle detection method for daytime and nighttime traffic conditions that extracts and tracks the corner features of moving vehicles instead of the entire vehicles, then the traffic parameters over each lane are predicted based on detected road lanes. However, Beymer et al. (1997) ignored the partial occlusions which might lead to the difficulties of detecting corners and used homography transformation for lane detection that might be not accurate enough during nighttime. Huang et al. (2008) proposed a detection method based on a block-based contrast analysis on the inter-frame variation information, but this method highly relies on the manually defined thresholds for the contrast measurement. Robert (2009) proposed a nighttime vehicle detection system that detects pairs of vehicle headlights firstly and then uses a decision tree to group them as vehicles, which might fail when the headlights are not paired (e.g., with a nearby motorcycle headlight) or occluded/invisible in the crowded scene. Kosaka and Ohashi (2015) extracted the bright, geometry and color features of headlights or taillights and then used a Support Vector Machine (SVM) to classify them for vehicle detection. However, Kosaka and Ohashi (2015)'s feature extraction could be affected by the complicated light reflection during nighttime. Chen et al. (2010) detected the vehicles by finding the bright objects during night, i.e., bright headlights or taillights. By only considering to extract bright objects, Chen et al. (2010) might not work well in the general night case. The above methods are based on traditional image processing techniques, which could be improved in the following three ways: (1) The feature extraction could be enhanced (not only considering the bright, geometry and color features of vehicle headlights or taillights), e.g., using the deep learning based CNN to extract more robust features; (2) The traditional classifiers to recognize vehicles

could be improved to some advanced deep learning based classifiers; (3) The feature extraction and final classifier could be incorporated into an end-to-end learning framework that are optimized together. Recently, several deep learning based methods for vehicle detection by Vancea et al. (2017), Ke et al. (2018), Yu et al. (2020) show improved performance than the traditional image processing based methods. The deep learning based methods could better reduce the false positive and false negative detection errors, which are more reliable in the complex real cases than the traditional image processing methods. Specifically, Vancea et al. (2017) used more robust features extracted by the deep CNN, Ke et al. (2018), Yu et al. (2020) designed different end-to-end deep learning frameworks for interested object (e.g., vehicle) detection. In this paper, the proposed method improves the traditional image processing methods into a new deep learning based pipeline in the above mentioned three ways with a special design for domain adaptation.

2.3. Domain adaptation

Typically, data distribution discrepancy always exists between different situations/domains. Multiple domain information can be used to reduce domain differences between the source and target domains as described in Fernando et al. (2013), which is called Domain Adaptation (DA) in machine learning. Although CNN achieves state-of-the-art performance in several image classification problems as introduced in Krizhevsky et al. (2012), training CNN requires a large set of manually labeled images. Thus, the research of DA is very important to generalize the deep learning usage. To solve this DA problem, some synthetic datasets by Richter et al. (2016), Wang et al. (2019a) are created to improve the performance in real world. Some studies by Chopra et al. (2013), Chen et al. (2015) describe domain adaptation techniques by training two or more deep networks in parallel using different combinations of source and target domain samples. Ganin and Lempitsky (2014) proposed an unsupervised domain adaptation method that uses a large amount of unlabeled data from the target domain. Othman et al. (2017) designed a DA network consisting of a pre-trained CNN and an additional hidden layer for handling cross-scene classification. Transfer learning method can improve the sensitivity of the model in some specific scene as described in Li et al. (2015). When there are great differences between the source and target domains, Song et al. (2019) introduced that the DA method by subspace alignment can help to improve image recognition. Another important research direction in DA is the image style transfer. For example, images in one style could be translated to the version in another style using the following methods: Pix2Pix by Isola et al. (2017), CycleGAN by Zhu et al. (2017), Coupled GAN by Liu and Tuzel (2016), instance-aware GAN (InstaGAN) by Mo et al. (2018), ComboGAN by Anoosheh et al. (2018), UNIT by Liu et al. (2017), MUNIT by Huang et al. (2018a), AttGAN by He et al. (2019), etc. Based on the image style transfer methods mentioned above, many works by Anoosheh et al. (2019), Mukherjee et al. (2019b), Mukherjee et al. (2019a), Romera et al. (2019), Sun et al. (2019), Dai and Van Gool (2018) have been proposed to narrow the gap between the daytime and nighttime situations to improve the performance of various tasks, such as semantic segmentation, retrieval-based localization, autonomous driving, and so on. Different with these works, our target in this paper is to improve the vehicle detection in nighttime with only labeled daytime data, which is accomplished by embedding the image style transfer to the deep learning based object detection model.

2.4. Traffic flow parameter estimation

Many devices and tools are widely used for the traffic parameter and state estimation as described in Seo et al. (2017), Deng et al. (2013), typically including: loop detectors by Wu et al. (2016), Coifman and Kim (2009), Liu and Sun (2014), video cameras by Malinovskiy et al. (2008), Tian et al. (2015), Wan et al. (2014), unmanned aerial vehicles (UAVs) by Khan et al. (2018), Ke et al. (2016), Ke et al. (2018), radio frequency identification (RFID) detector by Wu and Yang (2013), Huang et al. (2018b), Bluetooth devices by Bhaskar et al. (2014), GPS devices on vehicle by Simoncini et al. (2018), float car by Kong et al. (2016), light detection and ranging (LiDAR) sensors by Zhao et al. (2019), satellite remote sensing by Ahmadi et al. (2019), microwave sensors by Ma et al. (2015), etc. Based on that, a variety of traffic flow parameters can be extracted such as speed, density, quantity, queue length, travel time, etc. With the rapid development of computer vision and deep learning, the video based traffic flow parameter estimation method showing high accuracy becomes quite popular as described in Shastry and Schowengerdt (2005), Ke et al. (2015, 2016, 2018). Given the accurate vehicle detection results, the next-step traffic flow parameter estimation for daytime or nighttime is the same. However, due to the different light properties between daytime and nighttime, there could be a significant accuracy downgrade for vehicle detection when directly applying the daytime detection model to nighttime, especially when the manually labeled nighttime data is limited or unavailable for model training in many real-world scenarios. Most of existing researches in traffic flow parameter estimation assume that the accurate vehicle detection is already available, but they ignore the difficulties of vehicle detection in nighttime. It is obvious that improved vehicle detection in nighttime leads to more accurate traffic flow parameter estimation in nighttime. This paper focuses on the accurate and efficient traffic flow parameter estimation in both daytime and nighttime by a deep learning model trained with only daytime labeled images.

3. Methodology

For a better vehicle detection in traffic surveillance images during nighttime, we propose to use style transfer as the DA method to mitigate the domain difference between the source domain and the target domain, and then train a Faster R-CNN model for nighttime vehicle detection.

3.1. Framework

In this paper, we define that the set of labeled daytime traffic images (manually annotated bounding box of each vehicle in each image) is the Source Domain as **S**, and the set of unlabeled nighttime traffic images is the Target Domain as **T**. In this research problem, we have two Tasks to be addressed: 1. Detect the vehicles during daytime by Faster R-CNN; 2. Detect the vehicles during nighttime by Faster R-CNN with DA method.

For the Task 1, detecting vehicles during daytime in traffic surveillance images is a standard supervised learning problem, which can be accomplished by many CNN based object detection methods, such as Faster R-CNN by Ren et al. (2015), YOLO by Redmon et al. (2016), Mask R-CNN by He et al. (2017), etc. The CNN model used in the proposed method for vehicle detection is Faster R-CNN by Ren et al. (2015), due to its advanced accuracy and speed in object detection. The labeled daytime images of **S** are used as the training set to train a robust Faster R-CNN model for daytime vehicle detection. Faster R-CNN firstly extracts image-level features and then utilizes a Region Proposal Network (RPN) to generate object-level proposals, and then classifies the object-level proposals to be foreground/vehicle and background/non-vehicle, followed by a regression to further adjust the proposal location. One proposal is thought as a bounding-box region in the image. The backbone used for feature extraction here is VGG16 by Simonyan and Zisserman (2014), which has 16 layers in the CNN architecture. The Faster R-CNN model is an end-to-end learning system, whose network parameters can be learned by the gradient descent based backpropagation using inputs and outputs only.

For the Task 2, training a Faster R-CNN model for nighttime vehicle detection without manually labeled vehicles in nighttime training images is quite challenging. We propose a Faster R-CNN with DA method for this task. Specifically, style transfer is used to translate the real daytime images to synthetic/fake nighttime images by considering the image style of daytime and nighttime images. Image style can be translated via an unpaired image-to-image translation between two domains, so CycleGAN by Zhu et al. (2017) is used for this style transfer to reduce the domain difference. In this way, a real daytime image with manual labels can be translated to a synthetic/fake nighttime image, where the real daytime image and the synthetic/fake nighttime image have different styles but share the same manual labels. Finally, the synthetic/fake nighttime images with the shared manual labels are used to train a more robust Faster R-CNN model.

The pipeline of the proposed method is shown in Fig. 2. We will detail each main component of the proposed method in the next several sections.

3.2. Faster R-CNN based vehicle detection

Faster R-CNN proposed by Ren et al. (2015) has great performance in many object detection related tasks. It is a widely used CNN based deep learning model for object detection with a two-stage algorithm. It firstly generates object-level proposals and then classifies the generated object-level proposal as foreground/vehicle and background/non-vehicle, followed by a regression to further adjust the proposal location.

The Faster R-CNN network mainly contains two parts, one is the Region Proposal Network (RPN) that generates proposals and the other is Fast R-CNN that uses the generated proposals for classification and location adjustment by Ren et al. (2015). The backbone used for feature extraction here is VGG16 by Simonyan and Zisserman (2014), which has 13 convolutional layers in the CNN architecture. Convolutional layers for feature extraction are shared by both RPN and Fast R-CNN to improve the computation efficiency. The RPN will tell the Fast R-CNN where to look, that is, the place of the region proposals. RPN uses anchors of different scales ($32^2, 64^2, 128^2, 256^2, 512^2$ pixels) and various aspect ratios (1:1, 1:2, 2:1) in a sliding window manner to generate many object-level proposals. The anchors whose Intersection-over-Union (IoU) overlaps with manually labeled bounding box are above 0.7 or below 0.3 are set as positive and negative samples respectively during training RPN. We sample 256 anchors (128 as positive and 128 as negative) for one image during training RPN (first part). For training Fast R-CNN (second part), we fix the IoU threshold for NMS as 0.7 to generate about

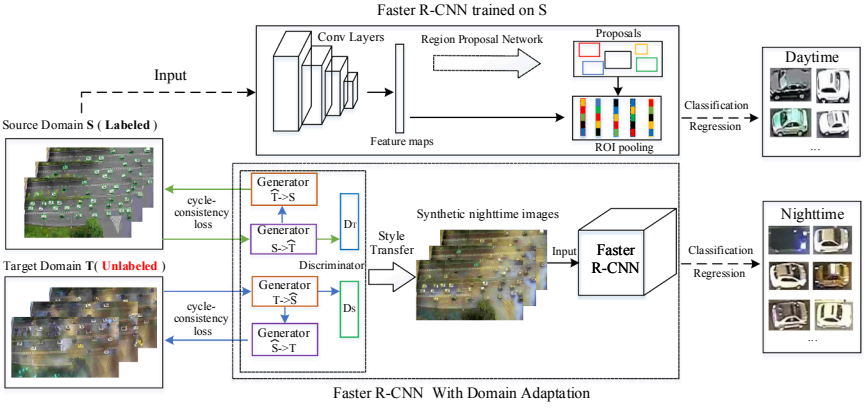


Fig. 2. Pipeline of the proposed Faster R-CNN with Domain Adaptation (DA) method for vehicle detection from labeled daytime images to unlabeled nighttime images.

2,000 proposals per image. Because each proposal has different size, region of interest pooling is implemented to pool each proposal to a fixed spatial extent, i.e., a fixed-and-same-size feature, which will be then used for later classification and regression.

Faster R-CNN mainly includes two loss functions to compare the predictions with the manually labeled ground truth. The first loss function L_{cls} is the loss of classification, which is used to evaluate the misalignment of classification. The second loss function L_{reg} the loss of regression, which is used to evaluate the proposal location misalignment. The total loss function L_{total} of Faster R-CNN contains the above two loss functions, they are defined as:

$$L_{total} = L_{cls} + \omega L_{reg} \tag{1}$$

$$L_{cls} = \frac{1}{N_{cls}} \sum_{i} - \left(y_i \log P_i + \left(1 - y_i\right) \log \left(1 - P_i\right) \right) \tag{2}$$

$$L_{reg} = \frac{1}{N_{reg}} \sum_{i} y_i * \text{smooth}_{L1} \left(B_i^* - B_i \right), \tag{3}$$

where the function of $\text{smooth}_{\text{L1}}$ is defined as:

$$\text{smoooth}_{L1} = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise,} \end{cases} \tag{4}$$

where N_{cls} is the RPN batch size (256), P_i is the probability of the i-th proposal to be vehicle and y_i is its manually labeled ground truth (1 for vehicle and 0 for non-vehicle), N_{reg} is the number of proposals (about 2,000), and $\text{smooth}_{\text{L1}}$ is a type of the loss function, B_i is 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 to the positive prediction, L_{cls} is the normalized loss for proposal classification, L_{reg} is the normalized regression loss for bounding box location adjustment and ω is a balance weight. In our experiments, ω was set to 1.

The whole Faster R-CNN is an end-to-end deep learning network that can be trained by gradient descent in backpropagation. Faster R-CNN's architecture is displayed in Fig. 3.

3.3. Style transfer from daytime to nighttime

In this paper, the purpose of the DA method is to learn the translation mapping between the source domain **S** in the daytime and the target domain **T** in the nighttime. The source domain **S** provides images and labels in the daytime, and the target domain **T** only provides images in the nighttime. By learning the unpaired image-to-image translation between that two different domains, we want to train a style transformer to generate synthetic/fake nighttime images from source domain **S**. This style transfer is implemented by CycleGAN by Zhu et al. (2017).

This style transfer is finished by training two generators and two adversarial discriminators. The generator is a kind of CNN to generate a new image by taking one image as input. The discriminator is a kind of CNN to classify real or fake images. As for the translation between domain **S** and domain **T**, we define two generators $G_{S\to T}$ and $G_{T\to S}$ as the transfer functions. The former one learns a transfer function from domain **S** to **T**, and the latter one learns a transfer function from domain **T** to **S**. Meanwhile, two adversarial discriminators D_T and D_S correspond to the $G_{S\to T}$ and $G_{T\to S}$. Specifically, D_T attempts to recognize whether the image is a real image from **T** or a generated synthetic/fake image by $G_{S\to T}$, and D_S tries to discriminate whether the image is a real one from **S** or a generated synthetic/fake one by $G_{T\to S}$. The source domain **S** provides labeled images I_S, and the target domain **T** provides images I_T. Given $i_S \in I_s$ and $i_T \in I_T, i_S$ and i_T represent any image in domain **S** and **T**, respectively.

In Fig. 2, the domain for generated synthetic images is highlighted with a hat, for example $\widehat{\mathbf{T}}$ the domain for generated synthetic/fake nighttime images from real daytime images and $\widehat{\mathbf{S}}$ the domain for generated synthetic/fake daytime images from real nighttime images.

Ideally, for one image $i_S \in I_s$, it can be translated to a synthetic image in $\widehat{\mathbf{T}}$ by the generator $G_{S\to T}$. The adversarial discriminator D_T will encourage the translated image indistinguishable from the domain **T**. After translating the synthetic image back to the domain **S** by $G_{T\to S}$, leading to a reconstructed image $G_{T\to S}(G_{S\to T}(i_S))$ which should be similar to the original image i_S. In other words, the reconstruction error for i_S should be minimized when training the GAN, so is that for the image i_T. This reconstruction error is called cycle consistency loss, and this algorithm can be applied to unpaired image-to-image style transfer. Following Zhu et al. (2017), the total loss function in the style transfer architecture is defined as:

$$\begin{aligned} L_{CycleGAN}\left(G_{S\to T}, G_{T\to S}, D_S, D_T, S, T\right) = L_{GAN}\left(G_{S\to T}, D_T, S, T\right) + \\ L_{GAN}\left(G_{T\to S}, D_S, T, S\right) + \lambda L_{Cycle}\left(G_{S\to T}, G_{T\to S}, S, T\right), \end{aligned} \tag{5}$$

where λ is the balance weight, L_{Cycle} is the cycle consistency loss in the cycle architecture, L_{GAN} is the adversarial training loss. The cycle consistency loss is used to regularize the GAN training. The cycle consistent loss is an L_1 penalty in the cycle architecture, which is defined as:

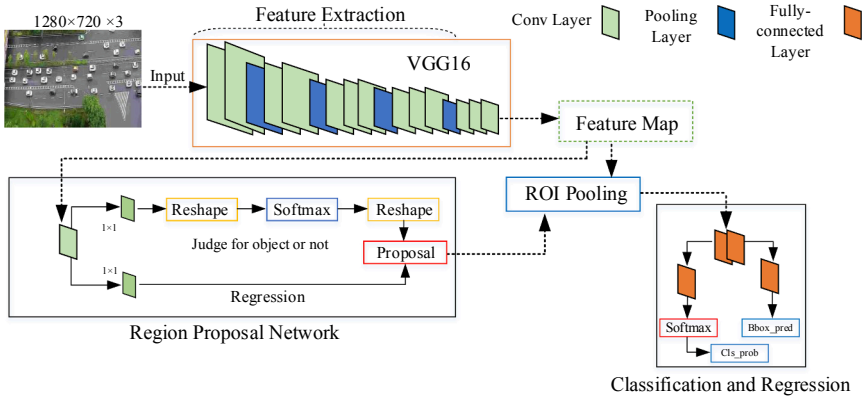


Fig. 3. Architecture of Faster R-CNN.

$$L_{Cycle}\left(G_{S\to T}, G_{T\to S}, S, T\right) = \mathrm{E}_{i_S\sim I_S}\left[\left\|G_{T\to S}\left(G_{S\to T}\left(i_S\right)\right) - i_S\right\|_1\right] + \mathrm{E}_{i_T\sim I_T}\left[\left\|G_{S\to T}\left(G_{T\to S}\left(i_T\right)\right) - i_T\right\|_1\right] \tag{6}$$

The adversarial training loss is defined as:

$$L_{GAN}\left(G_{S\to T}, D_T, S, T\right) = \mathrm{E}_{i_T\sim I_T}\left[\log\left(D_T\left(i_T\right)\right)\right] + \mathrm{E}_{i_S\sim I_S}\left[\log\left(1 - D_T\left(G_{S\to T}\left(i_S\right)\right)\right)\right]. \tag{7}$$

To train these generators and discriminators, we need to solve:

$$\begin{matrix} G^*_{S\to T} \\ G^*_{T\to S} \end{matrix} = \arg\min_{G_{S\to T}, G_{T\to S}} \max_{D_S, D_T} L_{CycleGAN}\left(G_{S\to T}, G_{T\to S}, D_S, D_T, S, T\right). \tag{8}$$

To solve the Eq. (8), in training, we alternatively update the network parameters by the ADAM optimization algorithm for the two generators and the two discriminators, which follows the official publicized code of CycleGAN by Zhu et al. (2017). After training, the learned generator $G_{S\to T}$ can be directly used to transfer the real daytime-style image to synthetic/fake nighttime-style image and also simultaneously keep the geometry and spatial relationship of vehicles in the image.

3.4. Faster R-CNN with domain adaptation

By style transfer, the synthetic/fake nighttime images from real daytime images are very similar to real nighttime images, leading to reduced domain difference, while the synthetic/fake nighttime images share the same vehicle locations. Therefore, the manually labeled ground truth bounding boxes for the vehicles in the source domain **S** can also be used for the synthetic/fake nighttime images. We then use those synthetic/fake nighttime images and corresponding labels in the source domain **S** as the training set to train a Faster R-CNN model.

3.5. Traffic flow parameter estimation

In the traffic flow theory, the volume, speed and density are the three most important parameters to describe the nature of traffic, and their relationship is given by the following equation:

$$Q = \sum_N V_i \times K_i, \tag{9}$$

where Q denotes the volume (in pc[1]/h) in the same-direction lanes, N denotes the number of lanes in the same direction, K_i denotes the density in the i-th lane defined as vehicle counts per freeway segment (in pc/km). V_i denotes the speed in the i-th lane, which is converted from pixels/frame to km/h. With the vehicle detection results, the detailed bounding boxes of vehicles in daytime and nighttime in every frame can be obtained, then traffic density and speed can be calculated. Density can be obtained by counting the vehicles in the unit freeway length. For the speed estimation, motion vectors are extracted by computing the sparse optical flow within a small Region of Interest (RoI), 3-by-4 pixels, in the center of detected vehicle between two adjacent frames as shown in Fig. 4, where

[1] pc: passenger cars.

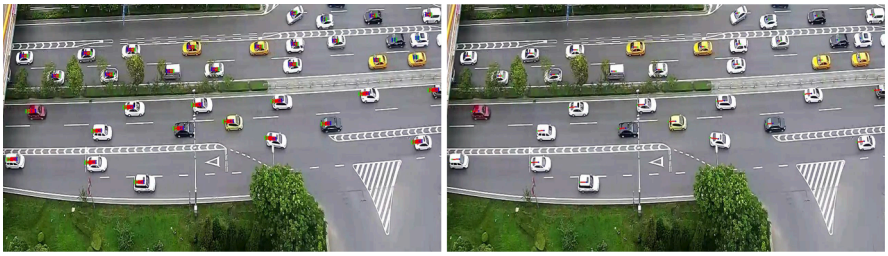


Fig. 4. Illustration of the computed motion vectors using the sparse optical flow by Bouguet et al. (2001) with the interval of five frames. The left image shows the motion vectors in the RoI of each vehicle and the right image displays the average motion vector for each vehicle. (Green dot: starting point, Blue dot: ending point, Red line: .motion vector.) (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the sparse optical flow is obtained using the pyramid Lucas Kanade feature tracker based algorithm by Bouguet et al. (2001) implemented in the OpenCV library. After obtaining all the motion vectors in the RoI for each detected vehicle, each vehicle's motion can be represented by the average motion vector in the RoI.

In each image/frame, we assume the object to have a real-world length l_1 in meters and image-world length l_2 in pixels, and the road segment length has L_s pixels in the image. To convert the pixel displacements into meters, we use the standard lane marking and urban taxi length to simply calibrate the ratio of $\frac{l_1}{l_2}$ (in m/pixel). Suppose $\overrightarrow{d_{(p,x)}}$ and $\overrightarrow{d_{(p,y)}}$ denote the p-th motion vector extracted for a vehicle in the horizontal and vertical directions, then the overall motion magnitude d_p of the p-th motion vector in pixels/frame can be calculated by the following equation:

$$d_p = \sqrt{\overrightarrow{d_{(p,x)}}^2 + \overrightarrow{d_{(p,y)}}^2}. \tag{10}$$

By taking an average for all the motion magnitudes d_p in the RoI of the q-th detected vehicle, the mean motion magnitude for q-th vehicle can be computed, defined as m_q. We average each vehicle's motion magnitude m_q in the i-th lane, defined as m_i. The frame rate is denoted as F that is fixed as 30 fps in the surveillance video. With these definitions above, the instantaneous traffic speed V_i (in km/h) and density K_i (in pc/km) in i-th lane are calculated using the following equations:

$$V_i = 3.6 \times m_i \times F \times \frac{l_1}{l_2} \tag{11}$$

$$K_i = 1000 \times \frac{T_i \times l_2}{L_s \times l_1}, \tag{12}$$

where T_i is the total number of vehicles in the i-th lane for the current frame and the constants 3.6 and 1000 are used for the transformations m/s to km/h and m to km, respectively. In this way, it is simple to compute the average speed V_i and average density K_i and also the volume Q of the lanes in the same direction.

4. Experiment

4.1. DNVD and TFPE datasets

In this paper, two new datasets were collected from a real traffic surveillance camera located in the middle section of South Second Ring Road in Xi'an, China. It is an urban expressway in the large city. The first dataset contains 2,200 traffic images (1,200 for daytime, 1,000 for nighttime) of different periods and dates. There are total 57,059 vehicles in the first dataset. The collected images are always with the same quality of 720p (size: $1,280 \times 720$ pixels) and the same scale due to the fixed camera position and internal parameters, whose only differences are the light conditions in daytime and nighttime. This dataset is named as Daytime and Nighttime Vehicle Detection (DNVD) dataset. The dataset is divided into two parts: training set and testing set. The training set has 1,000 manually labeled traffic images in daytime (denoted as Day-training). The testing set has 1,200 images, including a subset of 100 images in normal daytime traffic (denoted as Day-normal), a subset of 100 images in congested daytime traffic (denoted as Day-congested), 4 subsets of nighttime traffic images (denoted as Night-1, Night-2, Night-3, Night-4). Each image of the testing set is manually labeled for performance evaluation only, whose labels do not join the CNN training. The specific contents of the benchmark are shown in Table 1. In the experiment, the labeled daytime traffic images (Day-training) is the Source Domain as **S**, and the unlabeled nighttime traffic images (a combination of Night-1, Night-2, Night-3 and Night-4) is the Target Domain as **T**.

To evaluate the traffic flow parameter estimation performance of the proposed method, we collected the second dataset from the same traffic surveillance camera. Four videos were collected to test the proposed method: three 1,800-frame videos (60 s for each) in daytime (17:45 of 06/24/2019, 18:42 of 07/05/2019, 17:19 of 06/24/2019) and one 1,800-frame video (60 s) in nighttime (20:37 of 07/05/2019) of about 260 K vehicles in total, which is denoted as Traffic Flow Parameter Estimation (TFPE) datset in this paper. The ground truths of vehicle count and speed were manually labeled on one daytime 1,800-frame video and the nighttime 1,800-frame

Table 1
Details of the collected DNVD dataset in the experiment.

Training set	Image Number	Vehicle Number	Date	Time
Day-training	1,000	32,456	05/16/2019	19:10
Training set	Image Number	Vehicle Number	Date	Time
Day-normal	100	3,173	05/16/2019	19:00
Day-congested	100	4,539	04/14/2019	14:30
Night-1	250	7,322	06/01/2019	21:30
Night-2	250	5,554	06/02/2019	21:30
Night-3	250	1,738	06/02/2019	23:50
Night-4	250	2,277	07/05/2019	00:20

video respectively for the performance evaluation. For the ground-truth vehicle count, it was labeled by manually counting the vehicles from the each lane frame by frame. For the ground-truth vehicle speed, the moving distance of one sampled vehicle in each lane was manually labeled over an time interval, where five frames were used same as that in Ke et al. (2016, 2018). In the collected videos, the actual time interval of five frames was 0.167 s, the vehicle speed could still be viewed as a constant in this instant period. We did not label the ground truths for other two daytime 1,800-frame videos, which were used to show the estimated traffic flow parameters only.

4.2. Experimental setting

To validate the accuracy of the proposed method, our experiment mainly includes two parts: vehicle detection and traffic flow parameter estimation. In the vehicle detection experiment, two different scenarios are considered separately: 1. Detect the vehicles during daytime by Faster R-CNN; 2. Detect the vehicles during nighttime by Faster R-CNN with DA method. The following is the detailed setting.

1) Scenario I: We directly train a Faster R-CNN model on the set of Day-training using the images and manually labeled ground-truth. Then, we test the trained model on the sets of Day-normal and Day-congested.

2) Scenario II: We firstly use the proposed style transfer method to translate the image style from source domain **S** (Day-training) to the target domain **T** (a combination of Night-1, Night-2, Night-3 and Night-4) for domain difference reduction. In this way, each image in daytime style in the set of Day-training will be translated to a synthetic/fake image in nighttime style but with the same contents. As we defined before, the set of generated synthetic/fake images from **S** is $\widehat{\mathbf{T}}$ and then the manually labeled ground truth of **S** and corresponding synthetic/fake image in $\widehat{\mathbf{T}}$ are used to train a new Faster R-CNN model. This new Faster R-CNN with DA model can be used to detect the vehicles in the nighttime images (Night-1, Night-2, Night-3 and Night-4). In addition, we directly use the trained Faster R-CNN model in Scenario I to test the vehicle detection in nighttime as the comparison methods.

Besides, three traditional image processing methods based on background subtraction for vehicle detection, i.e., Mean-BGS by Li et al. (2013), Multi-LayerBGS by Yao and Odobez (2007), and DPGrimsonGMM by Stauffer and Grimson (1999), are used as the comparison methods in the experiments. Furthermore, two more deep learning methods for vehicle detection, i.e., SSD by Liu et al. (2016) and Faster R-CNN$_L$, are also utilized as comparison methods in the experiments. Specifically, the SSD model is trained on the set of Day-training (1000 images), and the Faster R-CNN$_L$ model is a Faster R-CNN model trained on another dataset collected by the same camera in different days (1000 daytime images) with four different light conditions during daytime (250 images: weak lights in cloudy weather, 250 images: middle lights in cloudy weather, 250 images: strong lights with shadows in sunny weather, 250 images: extra strong lights with more shadows in sunny weather). These new images are manually labeled for each vehicle location to train the Faster R-CNN$_L$ model.

We implement these methods and conduct the experiments using Python, OpenCV and PyTorch. During training, for Faster R-CNN based methods and SSD, we set the initial learning rate at 0.0001 and decayed with a factor of 0.9 of every 10 epochs. We set the batch size as 4 images and the momentum is 0.9 and the training epoch is 40 in our all experiments. For the CycleGAN, the balance weight λ is set to 10 for the cycle consistency loss, and we use the default setting for other hyper-parameters in its publicized code.[2] For the traditional image processing methods using background subtraction, we follow the default setting in their publicized code.[3] The experiments are conducted on a workstation with a CPU of 2.6 GHz, a memory of 12 GB and a NVIDIA GTX 2080 TI GPU.

In the evaluation of detection experimental results, there are six metrics used to evaluate those methods including Mean-BGS, Multi-LayerBGS, DPGrimsonGMM, SSD, Faster R-CNN, Faster R-CNN$_L$, and the Proposed Method. They include mean Average Precision(mAP), Precision, Recall, F-measure, Number of False Positives per image (N_{FP} error/image), and Number of False Negatives per image (N_{FN} error/image):

$$\text{Precision} = \frac{TP}{TP + FP} \tag{13}$$

[2] https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix.
[3] https://github.com/andrewssobral/bgslibrary.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{14}$$

$$\text{Fmeasure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{15}$$

where *TP* is short for true positive, *FP* for false positive, and *FN* for false negative. F-measure is an overall metric combining precision and recall together, so we use F-measure to report the overall performance. The mAP (%) metric is the precision averaged across all values of recall between 0 and 1 for the vehicles, which is considered as a comprehensive metric to well demonstrate the detection performance as used in Ren et al. (2015). For all the methods, the performance evaluation uses a uniform threshold of 0.5 for the IoU between the predicted bounding box and ground truth.

For the experiment of traffic flow parameter estimation, vehicle speed estimation and vehicle count were evaluated. Accuracy is used as a metric to evaluate vehicle count, and Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) are the metrics to evaluate vehicle speed. They are calculated using the following equations:

$$\text{MAE} = \frac{1}{n}\sum_{n}|y_i - y_i^*|, \tag{16}$$

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{n}(y_i - y_i^*)^2}, \tag{17}$$

where y_i denotes the ground truth value, y_i^* denotes the estimation value. Smaller MAE and RMSE errors indicate the better performance. At the same time, the percentage of error reduction (PER) is used to evaluate the error reduction of MAE by the proposed method, which is calculated by the difference between the previous MAE and the latter MAE over the previous MAE ($\frac{|P_{MAE} - L_{MAE}|}{P_{MAE}} \times 100\%$).

4.3. Experimental results

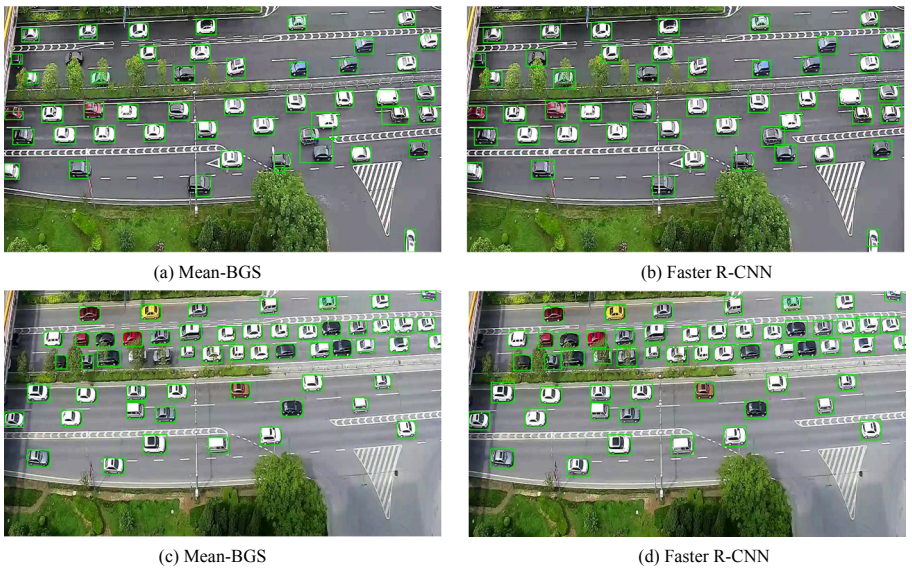
4.3.1. Results of vehicle detection

The experimental results for Scenario I are shown in Table 2. This table shows that the deep learning methods achieved better performance than the traditional image processing methods on our dataset. We can also see that the performances of all the methods slightly drop in the case of congested traffic. Among the deep learning methods, Faster R-CNN obtained best [96.41%, 93.79%] and SSD got [96.28%, 93.70%] as the mean [F-measure, mAP] on the two daytime testing sets, which is comparable and similar. Because Faster R-CNN got slightly better performance on our dataset, we choose Faster R-CNN as the baseline model for our algorithm development. Faster R-CNN$_L$ trained with different light conditions got [94.10%, 93.14%] as the mean [F-measure, mAP] on the two daytime testing sets, which is slightly lower than Faster R-CNN's performance. The possible reason might be the light condition differences in the training and testing sets. Among the traditional image processing methods, Mean-BGS achieved better performance, which is [94.46%, 90.18%] as the mean [F-measure, mAP] on the two daytime testing sets. The visualized detection results in Scenario I are shown in Fig. 5, where we only display the results by the most representative methods, i.e., Mean-BGS and Faster R-CNN. It is obvious that Faster R-CNN obtains better detection than Mean-BGS.

The experimental results for Scenario II are shown in Table 3. This table shows that the traditional image processing methods based

Table 2
Results in Scenario I on the collected DNVD dataset: daytime vehicle detection. On the two daytime testing sets, the mean [F-measure, mAP] by different methods are: Multi-LayerBGS [90.64%,82.98%], DPGrimsonGMM [90.00%,81.84%], Mean-BGS [94.46%,90.18%], SSD [96.28%,93.70%], Faster R-CNN [96.41%, 93.79%], Faster R-CNN$_L$ [94.10%, 93.14%].

Day-normal	Precision	Recall	F-measure	N_{FP}/image	N_{FN}/image	mAP
Multi-LayerBGS	97.37%	92.31%	94.77%	0.79	2.44	90.40%
DPGrimsonGMM	95.54%	87.80%	91.51%	1.3	3.87	84.86%
Mean-BGS	95.19%	96.63%	95.90%	1.55	1.07	92.45%
SSD	98.15%	98.80%	98.48%	0.59	0.38	**99.05%**
Faster R-CNN	**98.19%**	**99.02%**	**98.60%**	**0.58**	**0.31**	99.01%
Faster R-CNN$_L$	94.79%	98.95%	96.83%	1.71	0.33	98.29%
Day-congested	**Precision**	**Recall**	**F-measure**	N_{FP}/image	N_{FN}/image	**mAP**
Multi-LayerBGS	92.52%	81.23%	86.51%	2.98	8.52	75.56%
DPGrimsonGMM	93.39%	84.09%	88.50%	2.7	7.22	78.82%
Mean-BGS	92.77%	93.26%	93.01%	3.3	3.06	87.91%
SSD	**95.69%**	92.51%	94.07%	**1.89**	3.4	88.35%
Faster R-CNN	93.74%	**94.71%**	**94.22%**	2.87	**2.4**	**88.57%**
Faster R-CNN$_L$	89.32%	93.53%	91.38%	4.81	2.78	87.99%



(a) Mean-BGS (b) Faster R-CNN

(c) Mean-BGS (d) Faster R-CNN

Fig. 5. Detection results in Scenario I on the collected DNVD dataset: daytime vehicle detection. Top: normal traffic, Bottom: congested traffic.

Table 3
Results in Scenario II on the collected DNVD dataset: nighttime vehicle detection. Note that the training of all the deep learning methods did not use any labels in nighttime. On average of 4 nighttime testing sets, the mean [F-measure, mAP] by different methods are: Multi-LayerBGS [71.18%, 52.26%], DPGrimsonGMM [59.32%,37.10%], Mean-BGS [71.72%,52.71%], SSD [81.95%,79.72%], Faster R-CNN [82.85%,80.39%], Faster R-CNN$_L$ [80.30%,76.62%], Proposed method [86.40%, 84.62%].

Night-1	Precision	Recall	F-measure	N_{FP}/image	N_{FN}/image	mAP
Multi-LayerBGS	86.05%	67.07%	75.39%	3.18	9.644	58.09%
DPGrimsonGMM	63.48%	54.12%	58.43%	9.12	13.436	35.81%
Mean-BGS	79.40%	66.98%	72.66%	5.08	9.672	54.03%
SSD	88.93%	73.33%	80.38%	2.67	7.812	74.06%
Faster R-CNN	**95.60%**	75.08%	84.10%	**1.01**	7.3	74.84%
Faster R-CNN$_L$	87.57%	70.49%	78.11%	2.81	8.304	68.03%
Proposed	92.54%	**85.54%**	**88.90%**	2.02	**4.236**	**79.39%**
Night-2	Precision	Recall	F-measure	N_{FP}/image	N_{FN}/image	mAP
Multi-LayerBGS	87.21%	66.65%	75.56%	2.17	7.40	58.60%
DPGrimsonGMM	74.63%	64.62%	69.27%	4.88	7.86	49.25%
Mean-BGS	78.40%	62.48%	69.54%	3.82	8.33	49.09%
SSD	89.55%	73.42%	80.69%	1.90	5.90	73.78%
Faster R-CNN	**96.47%**	72.87%	83.02%	**0.59**	6.02	74.05%
Faster R-CNN$_L$	87.14%	68.97%	77.00%	2.18	6.65	66.99%
Proposed	93.36%	**84.82%**	**88.89%**	1.34	**3.37**	**80.72%**
Night-3	Precision	Recall	F-measure	N_{FP}/image	N_{FN}/image	mAP
Multi-LayerBGS	65.13%	61.16%	63.09%	2.27	2.7	40.76%
DPGrimsonGMM	60.36%	51.44%	55.55%	2.34	3.37	33.03%
Mean-BGS	62.72%	81.99%	71.07%	3.38	1.25	52.16%
SSD	**75.59%**	86.77%	80.79%	**1.94**	0.92	84.02%
Faster R-CNN	66.98%	**94.42%**	78.37%	3.23	**0.38**	85.63%
Faste R-CNN$_L$	72.50%	91.16%	80.77%	2.26	0.58	85.57%
Proposed	72.64%	93.33%	**81.69%**	2.44	0.46	**88.72%**
Night-4	Precision	Recall	F-measure	N_{FP}/image	N_{FN}/image	mAP
Multi-LayerBGS	74.50%	67.24%	70.68%	2.09	2.98	51.57%
DPGrimsonGMM	59.81%	49.28%	54.03%	3.01	4.62	30.31%
Mean-BGS	68.24%	79.93%	73.62%	3.38	1.82	55.56%
SSD	**84.10%**	87.83%	85.93%	**1.51**	1.10	87.00%
Faster R-CNN	78.60%	**94.69%**	85.90%	2.34	**0.48**	87.05%
Faster R-CNN$_L$	81.19%	89.91%	85.33%	1.8	0.87	85.89%
Proposed	79.41%	94.03%	**86.10%**	2.22	0.54	**89.66%**

on background subtraction performed worse than deep learning methods obviously. Among traditional image processing methods, Mean-BGS obtained better [71.72%, 52.71%] as mean [F-measure, mAP] on the 4 nighttime testing sets. Among the deep learning methods, SSD got [81.95%,79.72%], and Faster R-CNN$_L$ got [80.30%, 76.62%], and Faster R-CNN obtained [82.85%, 80.39%], and the proposed method achieved [86.40%,84.62%] (best performance) as the mean [F-measure, mAP] on the 4 nighttime testing sets. During nighttime, many vehicles are blurred and visually similar as the background road, so the traditional image processing methods based on background subtraction cannot effectively extract the moving vehicles. For example, a black or dark-color vehicle is extremely hard to be detected by background subtraction in the nighttime. The deep learning based methods obtained better performance due to the powerful discriminative feature extraction by the CNN frameworks. With the labeled daytime data only, SSD and Faster R-CNN models trained on the daytime data cannot well detect the vehicles in the nighttime because of the significant domain distribution discrepancy between daytime training and nighttime testing data. Even using different light conditions in daytime for the model training, Faster R-CNN$_L$ still cannot accurately detect the vehicles in the nighttime. However, the proposed method (Faster R-CNN with DA) could reduce the domain difference between daytime and nighttime by the style transfer, leading to the highest detection performance. In object detection, F-measure (considering both Precision and Recall) and mAP are the overall performance evaluation metrics. Although the proposed method sometimes might not obtain the highest Precision or Recall on each nighttime testing set, the proposed method achieved the highest overall F-measure and mAP on each of the four nighttime testing sets. Fig. 6 shows some translation demos of the original real daytime images and the corresponding synthetic/fake mages after the proposed style transfer. The synthetic/fake images are visually similar to the real nighttime images. The light conditions, road reflections, blurred air conditions in the synthetic/fake images are quite close to the real nighttime traffic images. Therefore, the domain difference between two domains are certainly reduced by the proposed style transfer. Fig. 7 shows the visualized results for the vehicle detection on real nighttime images, where we only display the results by the most representative methods, i.e., Mean-BGS, Faster R-CNN, and the proposed method. The Mean-BGS method has many missed detection during nighttime. Faster R-CNN is better than the Mean-BGS method, but it still has significant false positive and false negative errors. After style transfer based domain adaptation, the proposed method gets less false positive and false negative errors, which improves the vehicle detection in the nighttime.

Because there are many existing manually labeled ground truth for vehicle detection in the daytime images by the current urban traffic surveillance cameras, the research outcome of the proposed method is able to make maximum usage of the existing labeled daytime data to help the vehicle detection in the nighttime.

4.3.2. *Discussion of style transfer for vehicle detection*

In this section, we discuss the performance change if we apply a different style transfer method to Faster R-CNN. The proposed method applies CycleGAN method as the style transfer to Faster R-CNN, so we replace the CycleGAN method with another unpaired image-to-image translation method to test the performance change. The UNsupervised Image-to-image Translation Networks (UNIT) by Liu et al. (2017) can learn a joint distribution of images in different domains through its designed GAN-based deep learning framework for the unpaired image-to-image translation with good performance in many computer vision tasks, so we choose the UNIT method as the comparison in this study.

Similar as the proposed method, we implement the style transfer by UNIT to translate the daytime-style images to nighttime-style images, and we keep the same experimental setting with the proposed method but only replacing CycleGAN to UNIT. By using the daytime labeled images and corresponding transferred/fake nighttime-style images for model training, we denote the two methods as "Faster R-CNN+CycleGAN" and "Faster R-CNN+UNIT". Table 4 shows the detection results for nighttime vehicle detection and Fig. 8 displays an illustration of style transfer by CycleGAN and UNIT from daytime to nighttime. On average of 4 nighttime testing sets, using the mAP metric, Faster R-CNN+UNIT gets 79.69%, while the proposed method (Faster R-CNN+CycleGAN) achieves 84.62% as the best performance. Based on Fig. 8, the transferred nighttime-style images by CycleGAN maintain more structure information of vehicles than the UNIT method. Possibly because UNIT assumes the Gaussian latent space in its translation model as described by Liu et al. (2017), UNIT's transferred images are not as good as those by CycleGAN in our collected DNVD dataset. Therefore, the Faster R-

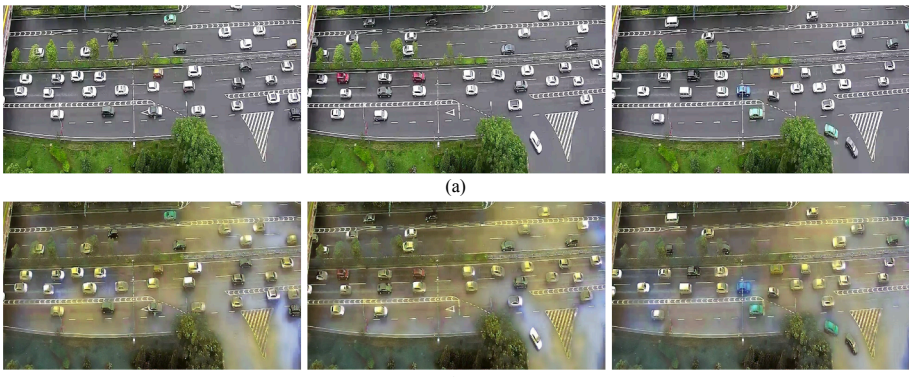


Fig. 6. Style transfer from daytime to nighttime. (a) real daytime traffic images, (b) corresponding synthetic/fake traffic images generated by CycleGAN.

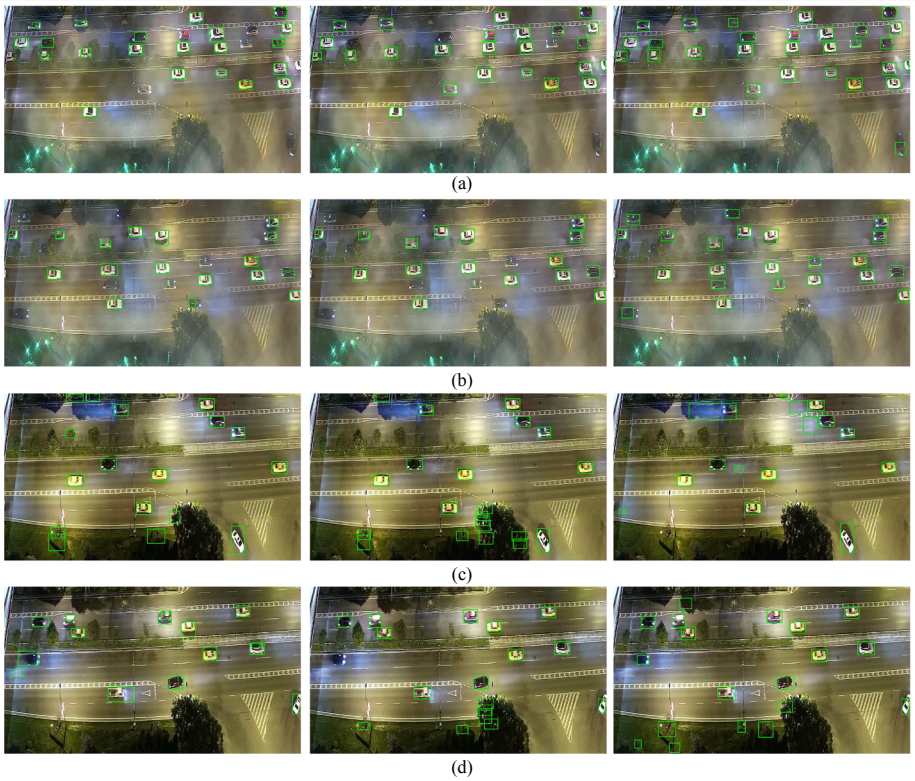


Fig. 7. Detection results in Scenario II on the collected DNVD dataset: nighttime vehicle detection. Detections on one example image of (a) Night-1, (b) Night-2, (c) Night-3 and (d) Night-4 are displayed. From left to right: results of Mean-BGS by Li et al. (2013), Faster R-CNN by Ren et al. (2015), Proposed Method.

Table 4
Results (mAPs) of Faster R-CNN with different style transfer methods for nighttime vehicle detection on the collected DNVD dataset.

Testing Sets	Faster R-CNN+UNIT	Proposed (Faster R-CNN+CycleGAN)
Night-1	70.56%	**79.39%**
Night-2	77.13%	**80.72%**
Night-3	82.87%	**88.72%**
Night-4	88.19%	**89.66%**
Mean mAP	79.69%	**84.62%**

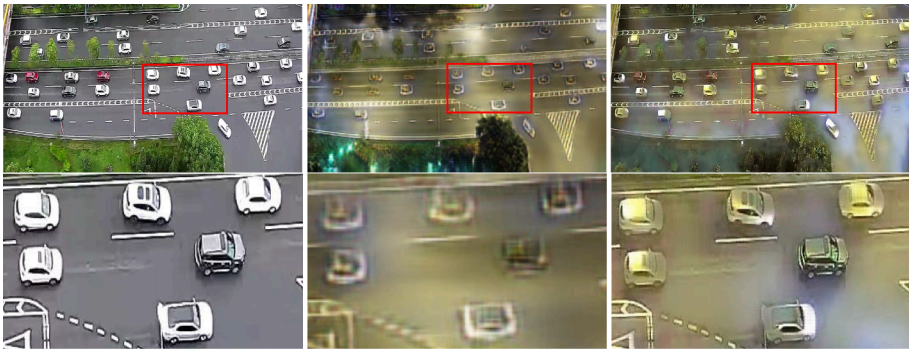
CNN+UNIT method gets lower mAP than the proposed method (Faster R-CNN+CycleGAN).

On summary, the proposed method with better style transfer method, like CycleGAN, could obtain higher vehicle detection performance.

4.3.3. Results of traffic flow parameter estimation

Because Faster R-CNN is the baseline model we choose for the experiments, we only use Faster R-CNN as the comparison method in this section. It is worth mentioning that the proposed method comes back to the Faster R-CNN method during the daytime.

The proposed method achieved a satisfactory performance in the vehicle count estimation in daytime and nighttime as shown in Table 5. During the daytime, the proposed method is just the Faster R-CNN method, which could reach the mean accuracy of 97.58% for vehicle count estimation in all the lanes. During the nighttime, the proposed method is the Faster R-CNN with DA method, which could reach the mean accuracy of 85.26% for vehicle count estimation in all the lanes, compared to 75.04% only by Faster R-CNN. It shows that the proposed method using DA could greatly improve the accuracy for vehicle counting during nighttime. In general, the proposed method achieved high vehicle counting accuracy in daytime and very good vehicle counting accuracy in nighttime. Because the deep learning model we trained did not use any nighttime manual labels as supervisions, the great accuracy improvement during nighttime is quite promising. Fig. 9 and Fig. 10 show the estimated and ground-truth counts for the daytime and nighttime traffic conditions. Because the position and internal parameters of the surveillance camera is fixed during the data collection, the prior lane



(a) (b) (c)

Fig. 8. An illustration of style transfer by different methods from daytime to nighttime. (a) real daytime traffic images, (b) transferred nighttime-style image of (a) by UNIT, and (c) transferred nighttime-style image of (a) by CycleGAN. Second row shows the zoomed images of the cropped region (red box) in the first row. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 5
Results in vehicle counting accuracy of each lane in daytime and nighttime on the labeled TFPE dataset. During the daytime, the average accuracy of Faster R-CNN over all six lanes is 97.58%. During the nighttime, the average accuracy of Faster R-CNN is 75.04%, while the proposed method improved it to be 85.26%. Lane distribution can be found in Fig. 9.

Vehicle Counting Accuracy in Daytime						
Method	Lane1	Lane2	Lane3	Lane4	Lane5	Lane6
Faster R-CNN	95.84%	98.22%	97.25%	99.41%	98.57%	96.19%
Vehicle Counting Accuracy in Nighttime						
Method	Lane1	Lane2	Lane3	Lane4	Lane5	Lane6
Faster R-CNN	76.34%	79.20%	52.63%	85.38%	85.54%	71.17%
Proposed	**85.85%**	**88.59%**	**68.28%**	**91.63%**	**93.78%**	**83.43%**

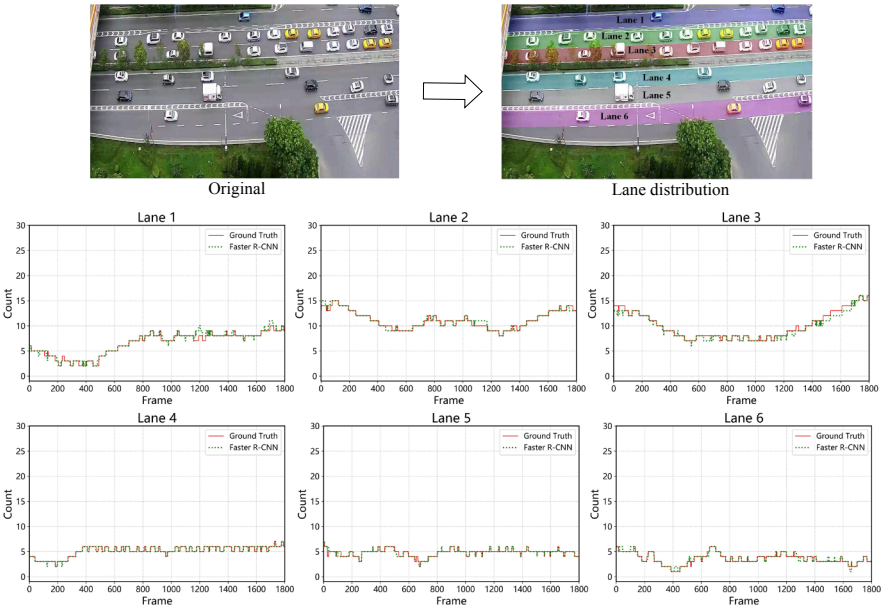


Fig. 9. The daytime vehicle count estimation on each lane in the labeled TFPE dataset. One example image with the detailed lane distribution is shown in the top.

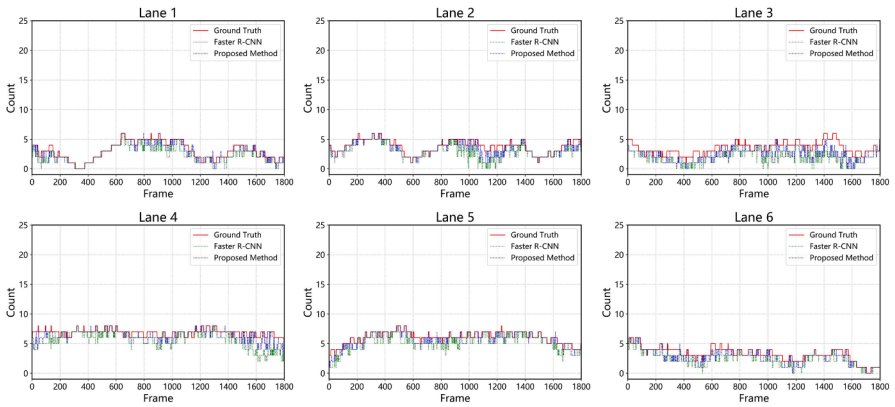


Fig. 10. The nighttime vehicle count estimation on each lane in the labeled TFPE dataset.

distribution can be manually segmented as that in Fig. 9. Lane3 has relatively low vehicle counting accuracy during nighttime because it has many occlusions caused by the trees. These occlusions do not affect the detection in daytime too much, leading to 97.25% accuracy in Lane3 during daytime, but the nighttime condition with the occlusions drops it to 68.28%. Experimental results on each lane show the significant accuracy increase by the proposed method compared to Faster R-CNN.

Table 6 shows the daytime and nighttime vehicle speed estimation results in the collected TFPE dataset. The estimated speed by computer vision can be compared with the loop detected speed or the empirical and analytical result, as introduced by Bickel et al. (2007), Lu and Coifman (2007). Similar to them, we compare the estimated speed by the proposed method with the manual ground truth. During the daytime, Faster R-CNN obtains average MAE of 1.87 km/h and average RMSE of 3.00. During the nighttime, Faster R-CNN obtains average MAE of 5.07 km/h and average RMSE of 8.77, while the proposed method improved the performance to average MAE of 4.22 and average RMSE of 7.61. Fig. 11 shows the estimated vehicle speed and the ground-truth speed on each lane during the daytime. Fig. 12 shows the nighttime estimated and the ground-truth speed on each line. It is obvious that the estimated vehicle speed by the proposed method could follow the changes of the ground-truth vehicle speed with relatively small MAE and RMSE errors. Lane3 has the largest speed estimation error during nighttime over all the lanes because it has occlusions caused by trees, as shown in Fig. 7, making both the vehicle detection and optical flow association much more difficult.

Furthermore, we could infer the speed, density and volume estimated by the proposed method for each lane in the collected TFPE dataset, which is shown in Table 7. Specifically, we sample the traffic surveillance video every five frames and estimate the speed by Eq. (11), density by Eq. (12), and then compute the volume by Eq. (9). We can see that the estimated speed is much related to the estimated count and density in each lane. When the vehicle count and density are small, the vehicle speed is high. The traffic flow parameters changed significantly in daytime and nighttime. Fig. 13 displays the relation of speed and density estimated by the proposed method. It shows that the collected traffic has free flow (high speed and low density) and congested flow (low speed and high density) situations during the daytime. While in the nighttime, the traffic changes to be only free flow with high speed and low density. As shown in Fig. 13, the proposed method provides an effective visualization to analyze and compare daytime and nighttime traffic flow in the same location.

5. Conclusions

In this paper, we proposed a new deep learning method for the situation-sensitive vehicle detection and traffic flow parameter estimation (i.e., speed, density and volume) in daytime and nighttime for urban surveillance cameras. The main contribution is that the

Table 6
Results in vehicle speed estimation of every lane in the labeled TFPE dataset. During the daytime, Faster R-CNN obtains average MAE of 1.87 km/h and average RMSE of 3.00. During the nighttime, Faster R-CNN obtains average MAE of 5.07 km/h and average RMSE of 8.77, while the proposed method reduced the errors to average MAE of 4.22 km/h and average RMSE of 7.61. The average percentage of error reduction (PER) is 15.13% by the proposed method.

Vehicle Speed Estimation								
Time	Method	Metric	Lane1	Lane2	Lane3	Lane4	Lane5	Lane6
Day	Faster R-CNN	MAE	1.42	4.09	1.07	1.72	1.40	1.52
		RMSE	2.50	5.60	1.66	3.89	2.15	2.21
Night	Faster R-CNN	MAE	5.21	4.24	14.78	1.57	1.61	3.04
		RMSE	9.65	9.40	21.31	2.56	2.49	7.26
	Proposed	MAE	**4.94**	**3.90**	**11.66**	**1.30**	**1.48**	**2.09**
		RMSE	**9.46**	**9.00**	**18.08**	**2.20**	**2.35**	**4.58**
		PER	5.18%	8.01%	21.11%	17.19%	8.07%	31.25%

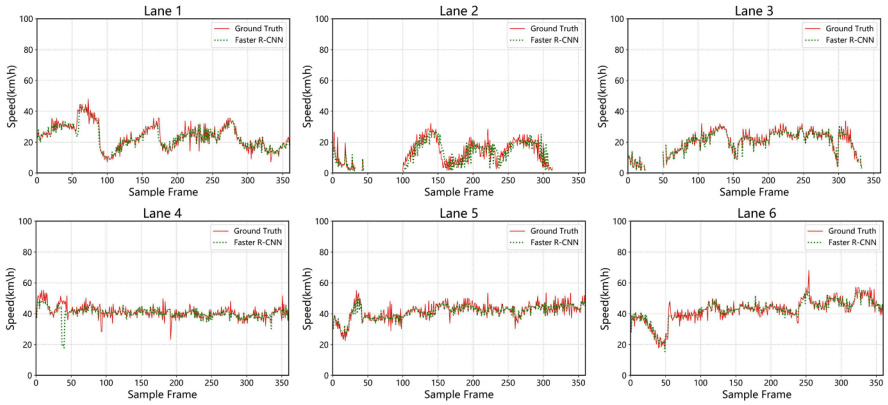


Fig. 11. Results in daytime vehicle speed estimation on each lane in the labeled TFPE dataset.

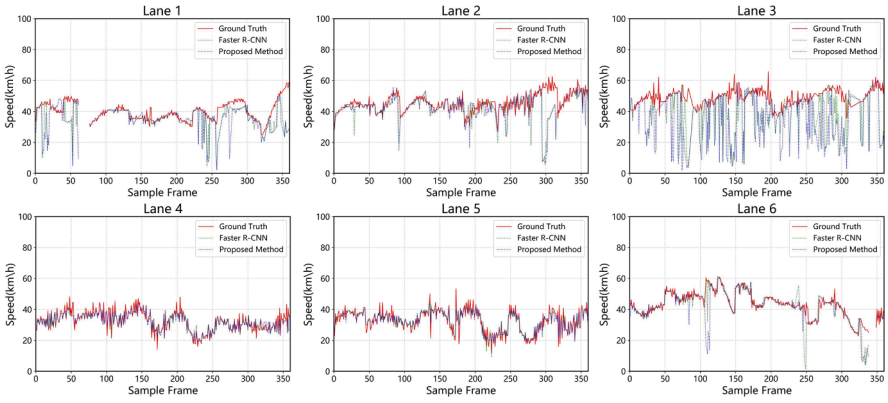


Fig. 12. Results in nighttime vehicle speed estimation on each lane in the labeled TFPE dataset.

Table 7
Summary of the estimated traffic flow parameters by the proposed method on each lane in the labeled TFPE dataset.

Time	Traffic Parameter	Lane1	Lane2	Lane3	Lane4	Lane5	Lane6
Daytime	Count (pc/frame)	6.5	11.1	9.7	5.1	4.7	3.5
	Speed (km/h)	23.4	9.2	17	40.3	41.2	42.3
	Density (pc/km)	62.5	106.8	93.1	49	45.7	34.2
	Volume (pc/h)	1,412.3	910.5	1,412.7	1,978.7	1,883.3	1,432.5
Nighttime	Count (pc/frame)	2.7	3.4	2.4	6.3	5.7	2.7
	Speed (km/h)	33.8	42.9	37.4	31.9	32.5	41.3
	Density (pc/km)	25.8	32.5	23.4	61	55	26.2
	Volume (pc/h)	936.5	1,401.1	908.8	1,951.2	1,770.3	1,138.6

proposed deep learning method only using the manual labels in daytime for training and a style transfer based domain adaptation method to improve the performance in the nighttime. Another contribution is that the proposed method could analyze and compare daytime and nighttime traffic flow in the same location with meaningful visualizations. The proposed method could make the maximum usage of available daytime labeled data by the traffic surveillance videos, which is quite promising to improve traffic data collection to avoid more manual annotations in training the deep learning models. In this paper, two datasets were collected and manually annotated for performance evaluation. The experimental results show that the proposed deep learning method could greatly improve the performances of vehicle detection, counting, speed estimation and collect better traffic flow parameters.

Future research work can be focused on continually improving the performance with more advanced domain adaptation methods, collecting multi-type traffic flow parameters (i.e., car, bus, truck), and mining diverse traffic conditions in daytime and nighttime.

CRediT authorship contribution statement

Jinlong Li: Data curation, Methodology, Software, Writing - original draft. **Zhigang Xu:** Methodology, Writing - review & editing,

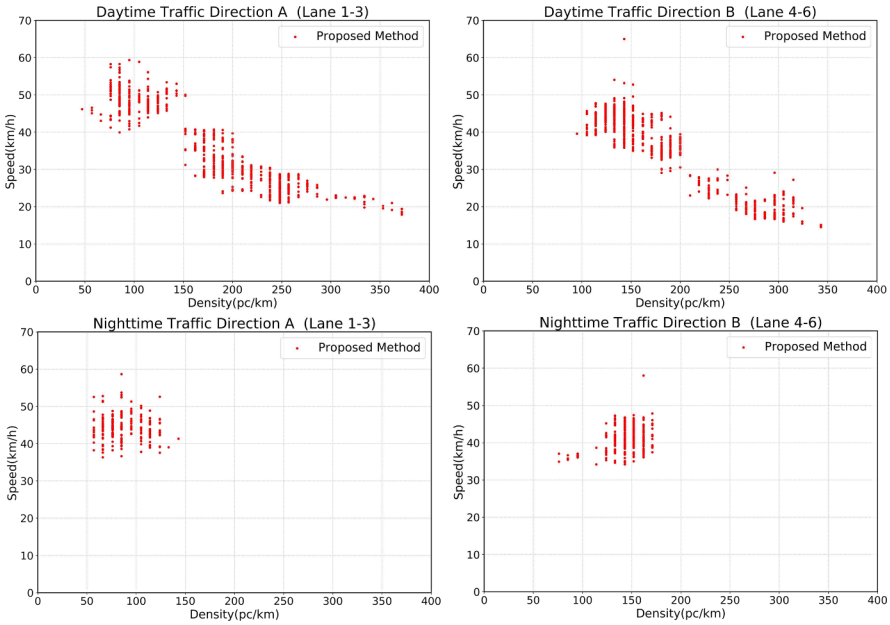


Fig. 13. The relation of estimated speed and density by the proposed method (daytime and nighttime) in the whole TFPE dataset.

Supervision. **Lan Fu:** Methodology, Software. **Xuesong Zhou:** Conceptualization, Writing - review & editing. **Hongkai Yu:** Methodology, Software, Writing - review & editing, Supervision.

Acknowledgments

Zhigang Xu 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). Hongkai Yu is supported by NVIDIA GPU Grant, Amazon Web Services (AWS) Cloud Credits for Research Award.

References

Abdulrahim, K., Salam, R.A., 2016. Traffic surveillance: A review of vision based vehicle detection, recognition and tracking. Int. J. Appl. Eng. Res. 11 (1), 713–726.
Agarwal, S., Awan, A., Roth, D., 2004. Learning to detect objects in images via a sparse, part-based representation. IEEE Trans. Pattern Anal. Mach. Intell. 26 (11), 1475–1490.
Ahmadi, S.A., Ghorbanian, A., Mohammadzadeh, A., 2019. Moving vehicle detection, tracking and traffic parameter estimation from a satellite video: a perspective on a smarter city. Int. J. Remote Sens. 40 (22), 8379–8394.
Anoosheh, A., Agustsson, E., Timofte, R., Van Gool, L., 2018. Combogan: Unrestrained scalability for image domain translation. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 783–790.
Anoosheh, A., Sattler, T., Timofte, R., Pollefeys, M., Van Gool, L., 2019. Night-to-day image translation for retrieval-based localization. In: International Conference on Robotics and Automation. IEEE, pp. 5958–5964.
Audebert, N., Le Saux, B., Lefèvre, S., 2017. Segment-before-detect: Vehicle detection and classification through semantic segmentation of aerial images. Remote Sens. 9 (4), 368.
Babari, R., Hautière, N., Dumont, É., Paparoditis, N., Misener, J., 2012. Visibility monitoring using conventional roadside cameras–emerging applications. Transp. Res. Part C: Emerg. Technol. 22, 17–28.
Bautista, C.M., Dy, C.A., Mañalac, M.I., Orbe, R.A., Cordel, M., 2016. Convolutional neural network for vehicle detection in low resolution traffic videos. In: IEEE Region 10 Symposium. IEEE, pp. 277–281.
Beymer, D., McLauchlan, P., Coifman, B., Malik, J., 1997. A real-time computer vision system for measuring traffic parameters. In: IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp. 495–501.
Bhaskar, A., Tsubota, T., Chung, E., et al., 2014. Urban traffic state estimation: Fusing point and zone based data. Transp. Res. Part C: Emerg. Technol. 48, 120–142.
Bickel, P.J., Chen, C., Kwon, J., Rice, J., Van Zwet, E., Varaiya, P., 2007. Measuring traffic. Statistical Sci. 581–597.
Bouguet, J.-Y., et al., 2001. Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm. Intel Corp. 5 (1–10), 4.
Chen, Q., Huang, J., Feris, R., Brown, L.M., Dong, J., Yan, S., 2015. Deep domain adaptation for describing people based on fine-grained clothing attributes. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 5315–5324.
Chen, Y.-L., Wu, B.-F., Huang, H.-Y., Fan, C.-J., 2010. A real-time vision system for nighttime vehicle detection and traffic surveillance. IEEE Trans. Industr. Electron. 58 (5), 2030–2044.
Chopra, S., Balakrishnan, S., Gopalan, R., 2013. Dlid: Deep learning for domain adaptation by interpolating between domains. In: International Conference on Machine Learning Workshop.
Coifman, B., Kim, S., 2009. Speed estimation and length based vehicle classification from freeway single-loop detectors. Transp. Res. Part C: Emerg. Technol. 17 (4), 349–364.
Coifman, B., Beymer, D., McLauchlan, P., Malik, J., 1998. A real-time computer vision system for vehicle tracking and traffic surveillance. Transp. Res. Part C: Emerg. Technol. 6 (4), 271–288.

Dai, D., Van Gool, L., 2018. Dark model adaptation: Semantic image segmentation from daytime to nighttime. In: International Conference on Intelligent Transportation Systems. IEEE, pp. 3819–3824.
Deng, W., Lei, H., Zhou, X., 2013. Traffic state estimation and uncertainty quantification based on heterogeneous data sources: A three detector approach. Transp. Res. Part B: Methodol. 57, 132–157.
Dong, Z., Wu, Y., Pei, M., Jia, Y., 2015. Vehicle type classification using a semisupervised convolutional neural network. IEEE Trans. Intell. Transp. Syst. 16 (4), 2247–2256.
Fernando, B., Habrard, A., Sebban, M., Tuytelaars, T., 2013. Unsupervised visual domain adaptation using subspace alignment. In: IEEE International Conference on Computer Vision, pp. 2960–2967.
Ganin, Y., Lempitsky, V., 2014. Unsupervised domain adaptation by backpropagation. arXiv preprint arXiv:1409.7495.
Guo, D., Zhu, L., Lu, Y., Yu, H., Wang, S., 2018. Small object sensitive segmentation of urban street scene with spatial adjacency between object classes. IEEE Trans. Image Process. 28 (6), 2643–2653.
Guo, H., Zheng, K., Fan, X., Yu, H., Wang, S., 2019. Visual attention consistency under image transforms for multi-label image classification. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 729–739.
Gupte, S., Masoud, O., Martin, R.F., Papanikolopoulos, N.P., 2002. Detection and classification of vehicles. IEEE Trans. Intell. Transp. Syst. 3 (1), 37–47.
Hale, D., Li, X., Ghiasi, A., Zhao, D., James, R., 2020. A methodology for trajectory-based calibration of microsimulation models. In: Transportation Research Board Annual Meeting.
Han, S., Han, Y., Hahn, H., 2009. Vehicle detection method using haar-like feature on real time system. World Acad. Sci. Eng. Technol. 59, 455–459.
He, K., Gkioxari, G., Dollár, P., Girshick, R., 2017. Mask r-cnn. In: IEEE International Conference on Computer Vision, pp. 2961–2969.
He, Z., Zuo, W., Kan, M., Shan, S., Chen, X., 2019. Attgan: Facial attribute editing by only changing what you want. IEEE Trans. Image Process. 28 (11), 5464–5478.
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 Trans. Intell. Transp. Syst. 15 (1), 6–20.
Huang, K., Wang, L., Tan, T., Maybank, S., 2008. A real-time object detecting and tracking system for outdoor night surveillance. Pattern Recogn. 41 (1), 432–444.
Huang, X., Liu, M.-Y., Belongie, S., Kautz, J., 2018a. Multimodal unsupervised image-to-image translation. In: European Conference on Computer Vision, pp. 172–189.
Huang, Y., Qian, L., Feng, A., Wu, Y., Zhu, W., 2018b. Rfid data-driven vehicle speed prediction via adaptive extended kalman filter. Sensors 18 (9), 2787.
Isola, P., Zhu, J.-Y., Zhou, T., Efros, A.A., 2017. Image-to-image translation with conditional adversarial networks. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1125–1134.
Kamijo, S., Matsushita, Y., Ikeuchi, K., Sakauchi, M., 2000. Traffic monitoring and accident detection at intersections. IEEE Trans. Intell. Transp. Syst. 1 (2), 108–118.
Ke, R., Kim, S., Li, Z., Wang, Y., 2015. Motion-vector clustering for traffic speed detection from uav video. In: IEEE International Smart Cities Conference. IEEE. pp. 1–5.
Ke, R., Li, Z., Kim, S., Ash, J., Cui, Z., Wang, Y., 2016. Real-time bidirectional traffic flow parameter estimation from aerial videos. IEEE Trans. Intell. Transp. Syst. 18 (4), 890–901.
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 Trans. Intell. Transp. Syst. 20 (1), 54–64.
Khan, M.A., Ectors, W., Bellemans, T., Janssens, D., Wets, G., 2018. Unmanned aerial vehicle-based traffic analysis: A case study for shockwave identification and flow parameters estimation at signalized intersections. Remote Sens. 10 (3), 458.
Kong, J., Zheng, Y., Lu, Y., Zhang, B., 2007. A novel background extraction and updating algorithm for vehicle detection and tracking. In: International Conference on Fuzzy Systems and Knowledge Discovery, vol. 3. IEEE, pp. 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 Gener. Comput. Syst. 61, 97–107.
Kosaka, N., Ohashi, G., 2015. Vision-based nighttime vehicle detection using censure and svm. IEEE Trans. Intell. Transp. Syst. 16 (5), 2599–2608.
Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105.
Li, L., Jiang, R., He, Z., Chen, X.M., Zhou, X., 2020. Trajectory data-based traffic flow studies: A revisit. Transp. Res. Part C: Emerg. Technol. 114, 225–240.
Li, S., Yu, H., Zhang, J., Yang, K., Bin, R., 2013. Video-based traffic data collection system for multiple vehicle types. IET Intel. Transp. Syst. 8 (2), 164–174.
Li, X., Li, Z., Han, J., Lee, J.-G., 2009. Temporal outlier detection in vehicle traffic data. In: IEEE International Conference on Data Engineering. IEEE, pp. 1319–1322.
Li, X., Ye, M., Fu, M., Xu, P., Li, T., 2015. Domain adaption of vehicle detector based on convolutional neural networks. Int. J. Control Autom. Syst. 13 (4), 1020–1031.
Lin, Y., Chen, J., Cao, Y., Zhou, Y., Zhang, L., Tang, Y.Y., Wang, S., 2016. Cross-domain recognition by identifying joint subspaces of source domain and target domain. IEEE Trans. Cybernet. 47 (4), 1090–1101.
Liu, H.X., Sun, J., 2014. Length-based vehicle classification using event-based loop detector data. Transp. Res. Part C: Emerg. Technol. 38, 156–166.
Liu, M.-Y., Tuzel, O., 2016. Coupled generative adversarial networks. In: Advances in Neural Information Processing Systems, pp. 469–477.
Liu, M.-Y., Breuel, T., Kautz, J., 2017. Unsupervised image-to-image translation networks. In: Advances in Neural Information Processing Systems, pp. 700–708.
Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C., 2016. Ssd: Single shot multibox detector. In: European Conference on Computer Vision. Springer, pp. 21–37.
Long, J., Shelhamer, E., Darrell, T., 2015. Fully convolutional networks for semantic segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440.
Lowe, D.G., 1999. Object recognition from local scale-invariant features. In: IEEE International Conference on Computer Vision, vol. 2. IEEE, pp. 1150–1157.
Lu, X.-Y., Coifman, B., 2007. Highway traffic data sensitivity analysis. Technical report. California PATH Research Report.
Ma, W., Qian, S., 2019. High-resolution traffic sensing with autonomous vehicles. arXiv preprint arXiv: 1910.02376.
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. Transp. Res. Part C: Emerg. Technol. 54, 187–197.
Malinovskiy, Y., Wu, Y.-J., Wang, Y., 2008. Video-based monitoring of pedestrian movements at signalized intersections. Transp. Res. Rec. 2073 (1), 11–17.
Mandellos, N.A., Keramitsoglou, I., Kiranoudis, C.T., 2011. A background subtraction algorithm for detecting and tracking vehicles. Expert Syst. Appl. 38 (3), 1619–1631.
Mertz, C., Qian, S., Chiang, J., 2020. Improving rush hour traffic flow by computer-vision-based parking detection and regulations.
Mo, S., Cho, M., Shin, J., 2018. Instagan: Instance-aware image-to-image translation. arXiv preprint arXiv:1812.10889.
Mu, K., Hui, F., Zhao, X., 2016. Multiple vehicle detection and tracking in highway traffic surveillance video based on sift feature matching. J. Inform. Process. Syst. 12 (2).
Mukherjee, A., Joshi, A., Hegde, C., Sarkar, S., 2019a. Semantic domain adaptation for deep classifiers via gan-based data augmentation. In: Conference on Neural Information Processing Systems Workshops, pp. 1–7.
Mukherjee, A., Joshi, A., Sarkar, S., Hegde, C., 2019b. Attribute-controlled traffic data augmentation using conditional generative models. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 83–87.
Othman, E., Bazi, Y., Melgani, F., Alhichri, H., Alajlan, N., Zuair, M., 2017. Domain adaptation network for cross-scene classification. IEEE Trans. Geosci. Remote Sens. 55 (8), 4441–4456.
Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You only look once: Unified, real-time object detection. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788.
Ren, S., He, K., Girshick, R., Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems, pp. 91–99.
Rezaei, M., Terauchi, M., Klette, R., 2015. Robust vehicle detection and distance estimation under challenging lighting conditions. IEEE Trans. Intell. Transp. Syst. 16 (5), 2723–2743.

Richter, S.R., Vineet, V., Roth, S., Koltun, V., 2016. Playing for data: Ground truth from computer games. In: European Conference on Computer Vision. Springer, pp. 102–118.
Robert, K., 2009. Night-time traffic surveillance: A robust framework for multi-vehicle detection, classification and tracking. In: IEEE International Conference on Advanced Video and Signal Based Surveillance. IEEE, pp. 1–6.
Romera, E., Bergasa, L.M., Yang, K., Alvarez, J.M., Barea, R., 2019. Bridging the day and night domain gap for semantic segmentation. In: IEEE Intelligent Vehicles Symposium. IEEE, pp. 1312–1318.
Rybski, P.E., Huber, D., Morris, D.D., Hoffman, R., 2010. Visual classification of coarse vehicle orientation using histogram of oriented gradients features. In: IEEE Intelligent vehicles symposium. IEEE, pp. 921–928.
Seo, T., Bayen, A.M., Kusakabe, T., Asakura, Y., 2017. Traffic state estimation on highway: A comprehensive survey. Ann. Rev. Control 43, 128–151.
Shastry, A.C., Schowengerdt, R.A., 2005. Airborne video registration and traffic-flow parameter estimation. IEEE Trans. Intell. Transp. Syst. 6 (4), 391–405.
Simoncini, M., Taccari, L., Sambo, F., Bravi, L., Salti, S., Lori, A., 2018. Vehicle classification from low-frequency gps data with recurrent neural networks. Transp. Res. Part C: Emerg. Technol. 91, 176–191.
Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
Song, S., Yu, H., Miao, Z., Zhang, Q., Lin, Y., Wang, S., 2019. Domain adaptation for convolutional neural networks-based remote sensing scene classification. IEEE Geosci. Remote Sens. Lett. 16 (8), 1324–1328.
Stauffer, C., Grimson, W.E.L., 1999. Adaptive background mixture models for real-time tracking. In: IEEE Conference on Computer Vision and Pattern Recognition, vol. 2. IEEE. pp. 246–252.
Sun, L., Wang, K., Yang, K., Xiang, K., 2019. See clearer at night: towards robust nighttime semantic segmentation through day-night image conversion. In: Artificial Intelligence and Machine Learning in Defense Applications, vol. 11169. International Society for Optics and Photonics, p. 111690A.
Tian, B., Yao, Q., Gu, Y., Wang, K., Li, Y., 2011. Video processing techniques for traffic flow monitoring: A survey. In: IEEE Conference on Intelligent Transportation Systems. IEEE, pp. 1103–1108.
Tian, B., Morris, B.T., Tang, M., Liu, Y., Yao, Y., Gou, C., Shen, D., Tang, S., 2014. Hierarchical and networked vehicle surveillance in its: a survey. IEEE Trans. Intell. Transp. Syst. 16 (2), 557–580.
Tian, B., Tang, M., Wang, F.-Y., 2015. Vehicle detection grammars with partial occlusion handling for traffic surveillance. Transp. Res. Part C: Emerg. Technol. 56, 80–93.
Vancea, F.I., Costea, A.D., Nedevschi, S., 2017. Vehicle taillight detection and tracking using deep learning and thresholding for candidate generation. In: IEEE International Conference on Intelligent Computer Communication and Processing. IEEE, pp. 267–272.
Wan, Y., Huang, Y., Buckles, B., 2014. Camera calibration and vehicle tracking: Highway traffic video analytics. Transp. Res. Part C: Emerg. Technol. 44, 202–213.
Wang, Q., Gao, J., Lin, W., Yuan, Y., 2019a. Learning from synthetic data for crowd counting in the wild. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 8198–8207.
Wang, Y., Zhang, D., Liu, Y., Dai, B., Lee, L.H., 2019b. Enhancing transportation systems via deep learning: A survey. Transp. Res. Part C: Emerg. Technol. 99, 144–163.
Wu, A., Yang, X., 2013. Real-time queue length estimation of signalized intersections based on rfid data. Procedia-Soc. Behav. Sci. 96, 1477–1484.
Wu, Y.-J., Chen, F., Lu, C.-T., Yang, S., 2016. Urban traffic flow prediction using a spatio-temporal random effects model. J. Intell. Transp. Syst. 20 (3), 282–293.
Yang, Z., Pun-Cheng, L.S., 2018. Vehicle detection in intelligent transportation systems and its applications under varying environments: A review. Image Vis. Comput. 69, 143–154.
Yao, J., Odobez, J.-M., 2007. Multi-layer background subtraction based on color and texture. In: IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp. 1–8.
Yu, H., Zhou, Y., Simmons, J., Przybyla, C.P., Lin, Y., Fan, X., Mi, Y., Wang, S., 2016. Groupwise tracking of crowded similar-appearance targets from low-continuity image sequences. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 952–960.
Yu, H., Guo, D., Yan, Z., Fu, L., Simmons, J., Przybyla, C.P., Wang, S., 2020. Weakly supervised easy-to-hard learning for object detection in image sequences. Neurocomputing 398, 71–82.
Zhao, J., Xu, H., Liu, H., Wu, J., Zheng, Y., Wu, D., 2019. Detection and tracking of pedestrians and vehicles using roadside lidar sensors. Transp. Res. Part C: Emerg. Technol. 100, 68–87.
Zhou, J., Gao, D., Zhang, D., 2007. Moving vehicle detection for automatic traffic monitoring. IEEE Trans. Veh. Technol. 56 (1), 51–59.
Zhu, J.-Y., Park, T., Isola, P., Efros, A.A., 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In: IEEE International Conference on Computer Vision, pp. 2223–2232.