3-2020

Domain Adaptation For Vehicle Detection In Traffic Surveillance Images From Daytime To Nighttime

Jinlong Ji

Zhigang Xu

Hongkai Yu
The University of Texas Rio Grande Valley

Lan Fu

Xuesong Zhou

DOMAIN ADAPTATION FOR VEHICLE DETECTION IN TRAFFIC SURVEILLANCE IMAGES FROM DAYTIME TO NIGHTTIME

Jinlong Li
Graduate Research Assistant
School of Information Engineering, Chang'an University
Middle Section of South 2nd-Ring Road, Xi'an, Shaanxi 710064 China
Email: lijinlong1117@chd.edu.cn

Zhigang Xu
Ph.D., Professor
School of Information Engineering, Chang'an University
Middle Section of South 2nd-Ring Road, Xi'an, Shaanxi 710064 China
Email: xuzhigang@chd.edu.cn

Hongkai Yu*
Ph.D., Assistant Professor (*corresponding author)
Department of Computer Science
University of Texas-Rio Grande Valley, Edinburg, TX 78539
Email: hongkai.yu@utrgv.edu

Lan Fu
Graduate Research Assistant
Department of Computer Science and Engineering
University of South Carolina, Columbia, SC 29201
Email: lanf@email.sc.edu

Xuesong Zhou
Ph.D., Associate Professor
School of Sustainable Engineering and the Built Environment
Arizona State University, Tempe, AZ 85287
Email: xzhou74@asu.edu

Word Count: 6,597 words + 3 tables = 7,347 words

Submitted [07/31/2019]

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 method based on Faster R-CNN with Domain Adaptation (DA) to improve the vehicle detection at nighttime. With the assistance of DA, the domain distribution discrepancy of Source and Target Domains is reduced. We collected a new dataset of 2,200 traffic images (1,200 for daytime and 1,000 for nighttime) of 57,059 vehicles for training and testing CNN. In the experiment, only using the manually labeled ground truths of daytime data, Faster R-CNN obtained 82.84% as F-measure on the nighttime vehicle detection, while the proposed method (Faster R-CNN+DA) achieved 86.39% as F-measure on the nighttime vehicle detection.

Keywords: nighttime vehicle detection, domain adaptation, deep learning, computer vision

INTRODUCTION

In recent years, more and more traffic video surveillance systems are installed, which provide more detailed traffic information, like traffic flow and vehicle speed *(1)*. Vehicles detection from these traffic surveillance images is an important part of the intelligent transportation system, safety monitoring, traffic control, and traffic simulation.

In computer vision, vehicle detection aims to discover the location of the interested object based on feature extraction and recognition, i.e., vehicle, from one single image. Some traditional methods use a variety of image processing algorithms in vehicle detection *(2)*. With the recent rapid development of deep learning, many Convolutional Neural Network (CNN) based methods are widely used for vehicle detection. 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 *(3)*, 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 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 show 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 nighttime vehicle detection. 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 *(4)*. The proposed vehicle detection problem based on DA is shown in **Figure 1**. The CNN model used in the proposed method for vehicle detection is Faster R-CNN (*5*), 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 Cycle GAN (*6*) is used for this style transfer.

To test the proposed method, we collected a new dataset, named as *CAU-UTRGV Benchmark*, 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 the traditional Background Subtraction algorithm *(7)* and the original Faster R-CNN on the *CAU-UTRGV Benchmark*, and the proposed method achieved the best F-measure performance for nighttime vehicle detection. The 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.

The main contributions of this paper are as follows: 1. For vehicle detection during daytime, a Faster R-CNN model is proposed for this task. 2. For vehicle detection during nighttime,

a new Faster R-CNN model with DA is proposed to make better usage of daytime data. Style transfer is used to realize the domain adaptation from the labeled daytime images (Source Domain) to unlabeled nighttime images (Target Domain). 3. A new dataset, named as *CAU-UTRGV Benchmark*, for this research is collected and manually labeled. The dataset contains 1,200 daytime images and 1,000 nighttime images of 57,059 vehicles, which will be publicized after paper acceptance.

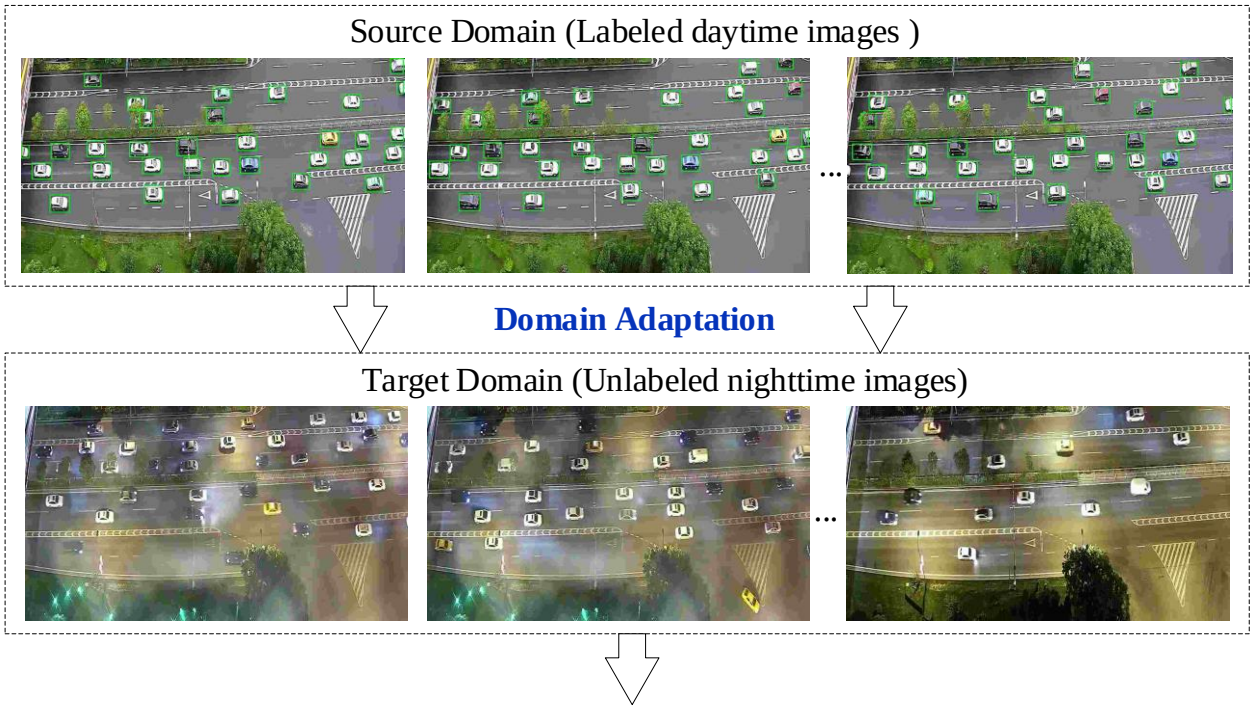


Figure 1 Vehicle detection with Domain Adaptation from the labeled daytime images (Source Domain) to unlabeled nighttime images (Target Domain).

PREVIOUS WORK

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 *(8)*. The separation between background and foreground are usually by detecting the changes. Some studies *(9, 10)* segment moving objects using space-time difference, and some other methods *(11-14)* use background subtraction algorithm 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 *(15, 16)*. More complex features have been used in vehicle detection such as local symmetry edge operators *(17)*, Scale Invariant Feature Transformation (SIFT) *(18)*, Speeded up Robust Features (SURF) *(19)*, Histogram of Oriented Gradient (HOG) *(20)* and Haar-like features *(21)*. Based on feature extraction, some large-scale crowded objects with similar appearance can be detected *(22)*. Recently, deep learning based CNN

methods (*23-29*) are widely used for vehicle detection, which have robust and advanced vehicle detection performances.

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 *(30)*. In general, in order to detect moving objects in night traffic surveillance, headlights and taillights are used as the salient features of moving vehicles. Beymer et al. *(31)* 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 regions. Huang et al. *(32)* proposed a detection method based on block-based contrast analysis and inter-frame variation information. Robert *(33)* proposed a nighttime vehicle detection system that detects pairs of headlights firstly, uses a supervised machine learning system to verify and recognize vehicles. Naoya et al. *(34)* used a detection method called center-surround extremes to detect the blobs in high speed based on the headlights and the taillights of vehicle. Chen et al. *(30)* propose a method to recognize vehicles by detecting and locating vehicle headlights and taillights using image segmentation and pattern analysis techniques. Extra hardware, like Kinect depth virtual loops, might be used for nighttime vehicle counting *(35)*. These methods are different with the proposed method in this paper. Our research goal is to improve the vehicle detection in unlabeled nighttime traffic images by making maximum usage of labeled daytime traffic image without adding extra hardware and labor costs.

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 *(36)*, which is called "Domain Adaptation" in machine learning. Although CNN achieves state-of-the-art performance in several image classification problems *(37)*, training CNN requires a large set of manually labeled images. Thus, the research of Domain Adaptation (DA) is very important to generalize the deep learning usage. To solve this DA problem, some synthetic datasets *(38, 39)* are created to improve the performance in real world. Some studies *(40, 41)* describe domain adaptation techniques by training two or more deep networks in parallel using different combinations of source and target domain samples. Ganin et al. *(42)* propose an unsupervised domain adaptation method that uses a large amount of unlabeled data from the target domain. Othman et al. *(43)* design 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 *(44)*. When there are great differences between the source and target domains, the DA method by subspace alignment can help to improve image recognition *(45)*.

METHODS

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.

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 finished: 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 (*5*), YOLO (*46*), Mask R-CNN (*47*), etc. The CNN model used in the proposed method for vehicle detection is Faster R-CNN (*5*), 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 (*48*), 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 Cycle GAN (*6*) 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 **Figure 2**. We will detail each main component of the proposed method in the next several sections.

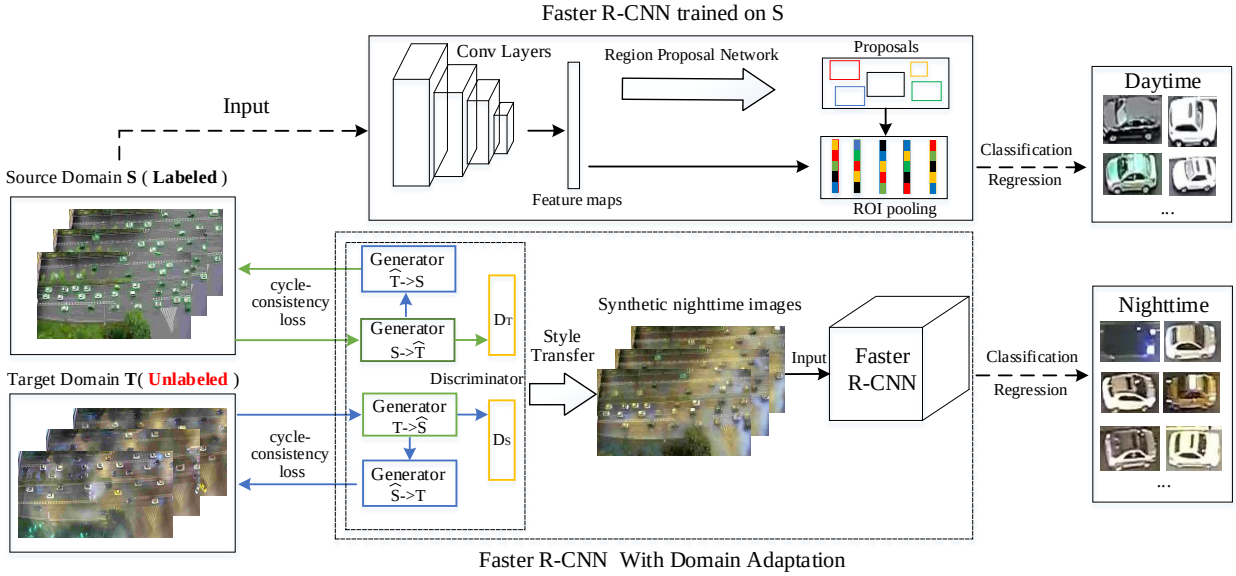


Figure 2 Pipeline of the proposed Faster R-CNN with Domain Adaptation (DA) method for vehicle detection from labeled daytime images to unlabeled nighttime images.

Faster R-CNN based vehicle detection

Faster R-CNN *(5)* 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 (*5*). The backbone used for feature extraction here is VGG16 (*48*), 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 2,000 proposals per image. Because each proposal has different size, region of interest (ROI) 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} is 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:

$$L_{total} = L_{cls} + \omega L_{reg} \quad (1)$$

$$L_{cls} = \frac{1}{N_{cls}} \sum_i -(y_i \log P_i + (1 - y_i)\log(1 - P_i)) \quad (2)$$

$$L_{reg} = \frac{1}{N_{reg}} \sum_i y_i * \mathrm{smooth}_{L1}(B_i^* - B_i), \quad (3)$$

where the function of smooth_{L1} is defined as:

$$\mathrm{smooth}_{L1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise,} \end{cases} \quad (4)$$

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 smooth_{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 **Figure 3**.

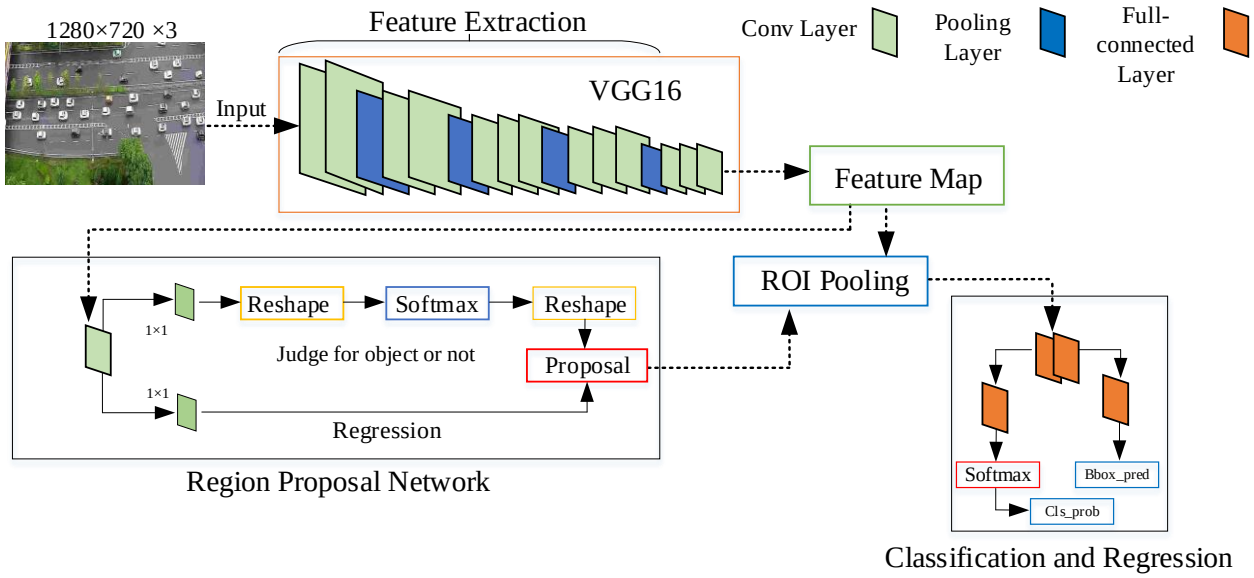


Figure 3 Architecture of Faster R-CNN.

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 Cycle GAN (*6*).

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 **Figure 2**, the domain for generated synthetic images is highlighted with a hat, for example $\hat{\mathbf{T}}$ means the domain for generated synthetic/fake nighttime images from real daytime images and $\hat{\mathbf{S}}$ means 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 $\hat{\mathbf{T}}$ 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 (*6*), the total loss function in the style transfer architecture is defined as:

$$L_{CycleGAN}(G_{S\to T}, G_{T\to S}, D_S, D_T, S, T) = L_{GAN}(G_{S\to T}, D_T, S, T) + L_{GAN}(G_{T\to S}, D_S, T, S) + \lambda L_{Cycle}(G_{S\to T}, G_{T\to S}, S, T), \quad (5)$$

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:

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

The adversarial training loss is defined as:

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

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

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

After solving **Equation 8** by gradient descent and backpropagation, 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.

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.

EXPERIMENTS

CAU-UTRGV Benchmark

In this paper, a new dataset is 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 dataset labeled 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 our dataset. The image size is 1,280 ×720 pixels. Since these data were collected and processed by the researchers of Chang'an University (CAU) and University of Texas-Rio Grande Valley (UTRGV) together, this dataset is named as *CAU-UTRGV Benchmark*.

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**.

TABLE 1 Details of the collected *CAU-UTRGV Benchmark* in the experiment.

Training set	Number	Vehicle Number	Date	Detail time
Day-training	1000	32456	05/16/2019	19:10
Testing set	Number	Vehicle Number	Date	Detail time
Day-normal	100	3173	05/16/2019	19:00
Day-congested	100	4539	04/14/2019	14:30
Night-1	250	7322	06/01/2019	21:30
Night-2	250	5554	06/02/2019	21:30
Night-3	250	1738	06/02/2019	23:50
Night-4	250	2277	07/05/2019	00:20

Experimental Setting

In the 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.

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. The traditional method for vehicle detection by background subtraction algorithm *(7)* is used as the comparison method.

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 $\widehat{\mathbf{T}}$ will be used to train a new Faster R-CNN model. This new Faster R-CNN with DA model, short as *Proposed Method*, can be used to detect the vehicles in the nighttime images (Night-1, Night-2, Night-3 and Night-4).

The traditional method for vehicle detection by background subtraction algorithm *(7)* is also used as the comparison method. In addition, we directly use the trained Faster R-CNN model in Scenario I to test the vehicle detection in nighttime as another comparison method.

We implement these methods and conduct the experiments using Python, OpenCV and PyTorch. During training, we set the initial learning rate at 0.0001 and decayed with a factor of 0.9 of every ten epochs. The momentum is 0.9 and the training epoch is 40 in our experiments. We set the batch size as 4 images in all the experiments. The balance weight λ is set to 10 for the cycle

consistency loss. The experiments are conducted on a workstation with a CPU of 2.6GHz, a memory of 12 GB and a NVIDIA GTX 2080 TI GPU.

In the evaluation of experimental results, there are five metrics used to evaluate those three methods including Background Subtraction, Faster R-CNN, and the Proposed Method. They include 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):

$$Precision = \frac{TP}{TP+FP} \quad (7)$$

$$Recall = \frac{TP}{TP+FN} \quad (8)$$

$$Fmeasure = \frac{2*Precision*Recall}{Precision+Recall}, \quad (9)$$

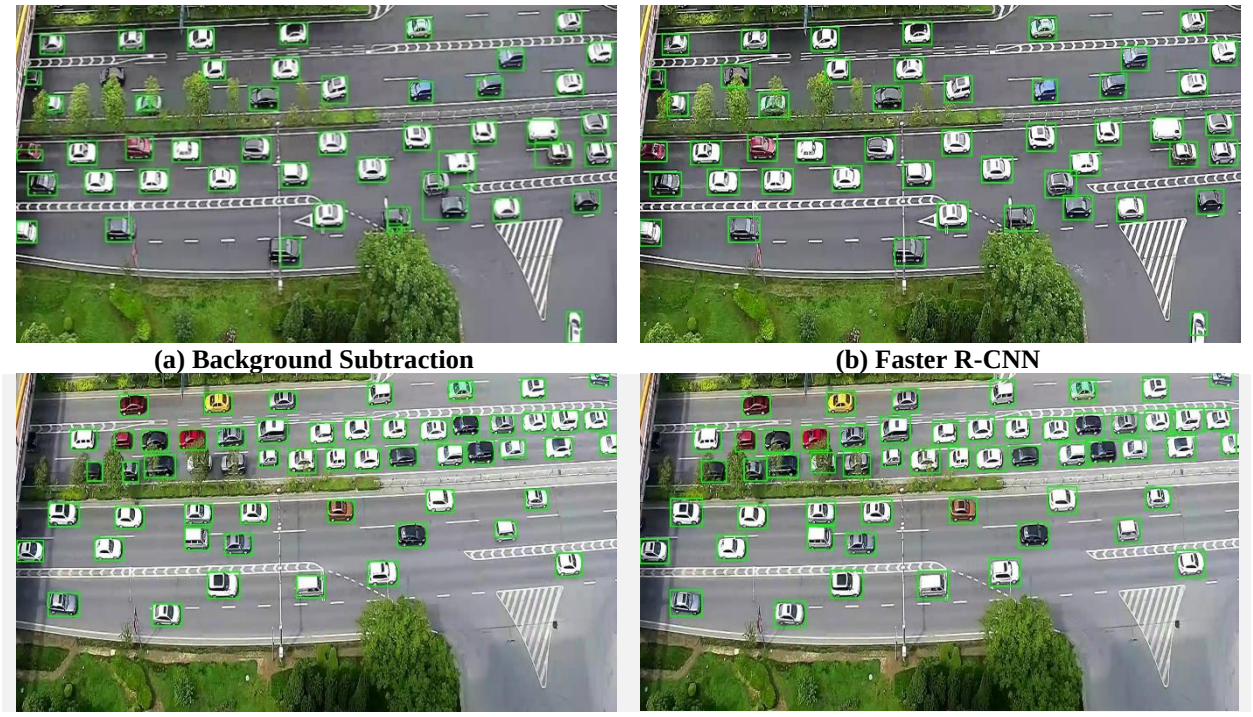
where *TP* is short for true positive, *FP* for false positive, and *FN* for false negative. *F-measure* is an overall metric combining precision and recall together, so we use *F-measure* to report the overall performance. For all the methods, the performance evaluation uses a uniform threshold of 0.5 for the IoU between the predicted bounding box and ground truth.

Experimental Results

The experimental results for Scenario I are shown in **Table 2**. This table shows that vehicle detection using the deep learning method, Faster R-CNN, is better than the traditional image processing method, Background Subtraction, in terms of five metrics. In congested traffic conditions, the performances of both two methods slightly drop. In the daytime, Faster R-CNN achieved 98.60% and 94.22% as F-measures for normal and congested traffic conditions. The visualized detection results by the two methods in Scenario I are shown in **Figure 4**. It is obvious that Faster R-CNN obtains better detections than Background Subtraction.

TABLE 2 Results in Scenario I: daytime vehicle detection.

	Background Subtraction (*7*)		Faster R-CNN (*5*)	
	Day-normal	Day-congested	Day-normal	Day-congested
Precision	95.18%	92.76%	**98.18%**	**93.74%**
Recall	96.62%	93.25%	**99.02%**	**94.71%**
F-measure	95.90%	93.01%	**98.60%**	**94.22%**
N_{FP} error/image	1.55	3.30	**0.58**	**2.87**
N_{FN} error/image	1.07	3.06	**0.31**	**2.40**



(a) Background Subtraction **(b) Faster R-CNN**

(c) Background Subtraction **(d) Faster R-CNN**

Figure 4 Detection results in Scenario I: daytime vehicle detection. Top: normal traffic, Bottom: congested traffic.

The experimental results for Scenario II are shown in **Table 3**. This table shows that the Background Subtraction method obtained the worst performance. During nighttime, many vehicles are blurred and visually similar as the background road, so the Background Subtraction method cannot effectively extract the moving vehicles. For example, a black or dark-color vehicle is extremely hard to be detected in the nighttime. Compared to Background Subtraction, the other two deep learning based methods obtained much better detection performance. Only using the manually labeled ground truths of daytime data, Faster R-CNN obtained 82.84% as F-measure on the 4 nighttime subsets, while the proposed method (Faster R-CNN+DA) achieved 86.39% as overall F-measure on the 4 nighttime subsets. Compared to the high F-measure (>94%) by Faster R-CNN in daytime in Scenario I, Faster R-CNN dropped to a lower overall F-measure (82.84%) on the 4 nighttime subsets because of the domain distribution discrepancy. After the style transfer, the domain difference is reduced, leading to a better overall F-measure (86.39%). **Figure 5** shows some translation demos of the original real daytime images and the corresponding synthetic/fake images 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. **Figure 6** shows the visualized results for the vehicle detection by the three methods on real nighttime images. The Background Subtraction method has many missed detections during nighttime. Faster R-CNN is better than the Background Subtraction 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 of Faster R-CNN in the nighttime.

TABLE 3 Results in Scenario II: nighttime vehicle detection. Note that the trainings of Faster R-CNN and Proposed Method did not use any labels in nighttime. On average of 4 nighttime subsets, Faster R-CNN obtains 82.84% as F-measure, while the Proposed method achieved 86.39% as F-measure.

Night-1	Precision	Recall	F-measure	N_{FP} error/image	N_{FN} error/image
Background Subtraction (*7*)	79.40%	66.97%	72.66%	5.08	9.67
Faster R-CNN (*5*)	**95.60%**	75.07%	84.10%	**1.01**	7.30
Proposed Method	92.53%	**85.53%**	**88.89%**	2.02	**4.23**
Night-2	Precision	Recall	F-measure	N_{FP} error/image	N_{FN} error/image
Background Subtraction (*7*)	78.40%	62.47%	69.53%	3.82	8.33
Faster R-CNN (*5*)	**96.47%**	72.86%	83.02%	**0.59**	6.02
Proposed Method	93.30%	**84.80%**	**88.88%**	1.34	**3.37**
Night-3	Precision	Recall	F-measure	N_{FP} error/image	N_{FN} error/image
Background Subtraction (*7*)	62.72%	81.99%	71.07%	3.38	1.25
Faster R-CNN (*5*)	66.97%	**94.41%**	78.36%	3.23	**0.38**
Proposed Method	**72.63%**	93.32%	**81.69%**	**2.44**	0.46
Night-4	Precision	Recall	F-measure	N_{FP} error/image	N_{FN} error/image
Background Subtraction (*7*)	68.24%	79.92%	73.62%	3.38	1.82
Faster R-CNN (*5*)	78.60%	**94.68%**	85.89%	2.34	**0.48**
Proposed Method	**79.40%**	94.02%	**86.10%**	**2.22**	0.54

(a)

(b)

Figure 5 Style transfer from daytime to nighttime. (a) real daytime traffic images, and (b) corresponding synthetic/fake traffic images in nighttime style.

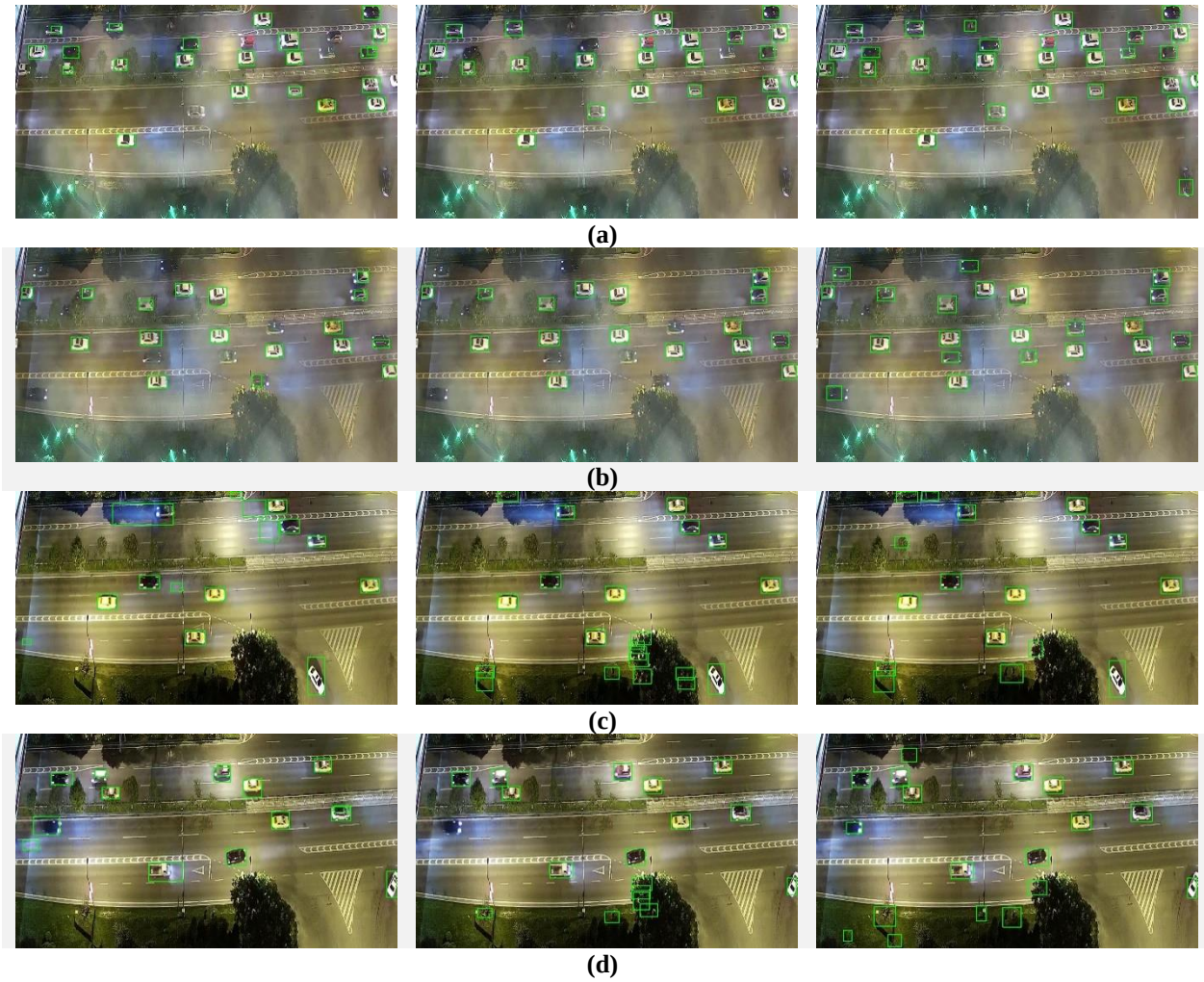


Figure 6 Detection results in Scenario II: nighttime vehicle detection. Detections on one example image of (a) Night-1, (b) Night-2, (c) Night-3 and (d) Night-4 are displayed. From left to right: results by Background Subtraction (*7*), Faster R-CNN (*5*), Proposed Method.

DISCUSSION

The experiments show that traditional image processing based methods, like Background Subtraction, for vehicle detection is not stable in congested daytime traffic conditions and much worse in nighttime traffic conditions. Deep learning based methods, like Faster R-CNN, is more accurate and robust to detect vehicles in daytime and nighttime. Directly applying the trained CNN model using daytime traffic data to detect vehicles in nighttime traffic images does not perform well with a significant performance drop. This performance drop is due to the domain difference between daytime domain and night domains. By the proposed style transfer based domain adaptation, the domain difference can be relieved so as to improve the CNN model.

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.

CONCLUSIONS

In this paper, we propose a Faster R-CNN with Domain Adaptation method for vehicle detection in the nighttime to use labeled daytime data (source domain) to help the unlabeled nighttime data (target domain). For experiments, we collected a new dataset *CAU-UTRGV Benchmark* containing 2,200 labeled traffic images to test the proposed method and other comparison methods. Using style transfer based domain adaptation, the deep learning based method Faster R-CNN can be improved for vehicle detection in the nighttime traffic surveillance. Without using any extra hardware and labor costs, the proposed method can improve the current vehicle detection in the nighttime traffic surveillance. Our future work will be focused on traffic flow parameter extraction, like flow, speed and density, in daytime and nighttime together.

ACKNOWLEDGMENTS

This work is supported by NVIDIA GPU Grant and the National Natural Science Foundation of China (No.51278058), 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), and Application of Basic Research Project for National Ministry of Transport (No.2015319812060).

AUTHOR CONTRIBUTIONS

The authors confirm contribution to the paper as follows: system design and algorithm development: Hongkai Yu, Jinlong Li, Zhigang Xu and Xuesong Zhou; experiment design: Hongkai Yu, Zhigang Xu; experiments and programming: Jinlong Li, Lan Fu; draft manuscript preparation: Hongkai Yu, Jinlong Li, Zhigang Xu and Xuesong Zhou. All authors reviewed the results and approved the final version of the manuscript.

REFERENCES

1. Yang Z, Pun-Cheng L S C. Vehicle detection in intelligent transportation systems and its applications under varying environments: A review[J]. Image and Vision Computing, 2018, 69: 143-154.
2. Abdulrahim K, Salam R A. Traffic surveillance: A review of vision based vehicle detection, recognition and tracking [J]. International Journal of Applied Engineering Research, 2016, 11(1): 713-726.
3. Guo H, Zheng K, Fan X, et al. Visual attention consistency under image transforms for multi-label image classification[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019: 729-739.
4. Lin Y, Chen J, Cao Y, et al. Cross-domain recognition by identifying joint subspaces of source domain and target domain[J]. IEEE Transactions on Cybernetics, 2016, 47(4): 1090-1101.
5. Ren S, He K, Girshick R, et al. Faster R-CNN: Towards real-time object detection with region proposal networks[C]//Advances in Neural Information Processing Systems. 2015: 91-99.
6. Zhu J Y, Park T, Isola P, et al. Unpaired image-to-image translation using cycle-consistent adversarial networks[C]//Proceedings of the IEEE International Conference on Computer Vision. 2017: 2223-2232.
7. Li S, Yu H, Zhang J, et al. Video-based traffic data collection system for multiple vehicle types[J]. IET Intelligent Transport Systems, 2013, 8(2): 164-174.
8. Tian B, Yao Q, Gu Y, et al. Video processing techniques for traffic flow monitoring: A survey[C]//2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC). IEEE, 2011: 1103-1108.
9. Kamijo S, Matsushita Y, Ikeuchi K, et al. Traffic monitoring and accident detection at intersections[J]. IEEE Transactions on Intelligent Transportation Systems, 2000, 1(2): 108-118.
10. Li X, Li Z, Han J, et al. Temporal outlier detection in vehicle traffic data[C]//2009 IEEE 25th International Conference on Data Engineering. IEEE, 2009: 1319-1322.
11. Kong J, Zheng Y, Lu Y, et al. A novel background extraction and updating algorithm for vehicle detection and tracking[C]//Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007). IEEE, 2007, 3: 464-468.
12. Mandellos N A, Keramitsoglou I, Kiranoudis C T. A background subtraction algorithm for detecting and tracking vehicles[J]. Expert Systems with Applications, 2011, 38(3): 1619-1631.
13. Zhou J, Gao D, Zhang D. Moving vehicle detection for automatic traffic monitoring[J]. IEEE Transactions on Vehicular Technology, 2007, 56(1): 51-59.

14. Gupte S, Masoud O, Martin R F K, et al. Detection and classification of vehicles[J]. IEEE Transactions on Intelligent Transportation Systems, 2002, 3(1): 37-47.
15. Lowe D G. Object recognition from local scale-invariant features[C]//Proceedings of the IEEE International Conference on Computer Vision. 1999: 1150-1157.
16. Tian B, Morris B T, Tang M, et al. Hierarchical and networked vehicle surveillance in ITS: a survey[J]. IEEE Transactions on Intelligent Transportation Systems, 2014, 16(2): 557-580.
17. Agarwal S, Awan A, Roth D. Learning to detect objects in images via a sparse, part-based representation[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2004 (11): 1475-1490.
18. Mu K, Hui F, Zhao X. Multiple Vehicle Detection and Tracking in Highway Traffic Surveillance Video Based on SIFT Feature Matching[J]. Journal of Information Processing Systems, 2016, 12(2).
19. Hsieh J W, Chen L C, Chen D Y. Symmetrical SURF and its applications to vehicle detection and vehicle make and model recognition[J]. IEEE Transactions on Intelligent Transportation Systems, 2014, 15(1): 6-20.
20. Rybski P E, Huber D, Morris D D, et al. Visual classification of coarse vehicle orientation using histogram of oriented gradients features[C]//2010 IEEE Intelligent Vehicles Symposium. IEEE, 2010: 921-928.
21. Han S, Han Y, Hahn H. Vehicle detection method using Haar-like feature on real time system[J]. World Academy of Science, Engineering and Technology, 2009, 59: 455-459.
22. Yu H, Zhou Y, Simmons J, et al. Groupwise tracking of crowded similar-appearance targets from low-continuity image sequences[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 952-960.
23. Dong Z, Wu Y, Pei M, et al. Vehicle type classification using a semisupervised convolutional neural network[J]. IEEE Transactions on Intelligent Transportation Systems, 2015, 16(4): 2247-2256.
24. Rezaei M, Terauchi M, Klette R. Robust vehicle detection and distance estimation under challenging lighting conditions[J]. IEEE Transactions on Intelligent Transportation Systems, 2015, 16(5): 2723-2743.
25. Ke R, Li Z, Tang J, et al. Real-time traffic flow parameter estimation from UAV video based on ensemble classifier and optical flow[J]. IEEE Transactions on Intelligent Transportation

Systems, 2018, 20(1): 54-64.

26. Bautista C M, Dy C A, Mañalac M I, et al. Convolutional neural network for vehicle detection in low resolution traffic videos[C]//2016 IEEE Region 10 Symposium (TENSYMP). IEEE, 2016: 277-281.
27. Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015: 3431-3440.
28. Audebert N, Le Saux B, Lefèvre S. Segment-before-detect: Vehicle detection and classification through semantic segmentation of aerial images[J]. Remote Sensing, 2017, 9(4): 368.
29. Guo D, Zhu L, Lu Y, et al. Small Object Sensitive Segmentation of Urban Street Scene With Spatial Adjacency Between Object Classes[J]. IEEE Transactions on Image Processing, 2018, 28(6): 2643-2653.
30. Chen Y L, Wu B F, Huang H Y, et al. A real-time vision system for nighttime vehicle detection and traffic surveillance[J]. IEEE Transactions on Industrial Electronics, 2010, 58(5): 2030-2044.
31. Beymer D, McLauchlan P, Coifman B, et al. A real-time computer vision system for measuring traffic parameters[C]//Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 1997: 495-501.
32. Huang K, Wang L, Tan T, et al. A real-time object detecting and tracking system for outdoor night surveillance[J]. Pattern Recognition, 2008, 41(1): 432-444.
33. Robert K. Night-time traffic surveillance: A robust framework for multi-vehicle detection, classification and tracking[C]//2009 IEEE International Conference on Advanced Video and Signal Based Surveillance. IEEE, 2009: 1-6.
34. Kosaka N, Ohashi G. Vision-based nighttime vehicle detection using CenSurE and SVM[J]. IEEE Transactions on Intelligent Transportation Systems, 2015, 16(5): 2599-2608.
35. Hu, Zhaozheng, Rufeng Zhang, and Mengchao Mu. Nighttime Vehicle Detection, Counting, and Classification Using Kinect Depth Virtual Loops[C]//TRB 2018 Annual Meeting, No. 18-02724, 2018.
36. Fernando B, Habrard A, Sebban M, et al. Unsupervised visual domain adaptation using subspace alignment[C]//Proceedings of the IEEE International Conference on Computer Vision. 2013: 2960-2967.
37. Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural

networks[C]//Advances in Neural Information Processing Systems. 2012: 1097-1105.

38. Richter S R, Vineet V, Roth S, et al. Playing for data: Ground truth from computer games[C]//European Conference on Computer Vision. Springer, Cham, 2016: 102-118.
39. Wang Q, Gao J, Lin W, et al. Learning from synthetic data for crowd counting in the wild[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019: 8198-8207.
40. Chopra S, Balakrishnan S, Gopalan R. Dlid: Deep learning for domain adaptation by interpolating between domains[C]//ICML workshop on challenges in representation learning. 2013, 2(6).
41. Chen Q, Huang J, Feris R, et al. Deep domain adaptation for describing people based on fine-grained clothing attributes[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015: 5315-5324.
42. Ganin Y, Lempitsky V. Unsupervised domain adaptation by backpropagation[J]. arXiv preprint arXiv:1409.7495, 2014.
43. Othman E, Bazi Y, Melgani F, et al. Domain adaptation network for cross-scene classification[J]. IEEE Transactions on Geoscience and Remote Sensing, 2017, 55(8): 4441-4456.
44. Li X, Ye M, Fu M, et al. Domain adaption of vehicle detector based on convolutional neural networks[J]. International Journal of Control, Automation and Systems, 2015, 13(4): 1020-1031.
45. Song S, Yu H, Miao Z, et al. Domain Adaptation for Convolutional Neural Networks-Based Remote Sensing Scene Classification[J]. IEEE Geoscience and Remote Sensing Letters, 2019.
46. Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection[C]//Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. 2016: 779-788.
47. He K, Gkioxari G, Dollár P, et al. Mask r-cnn[C]//Proceedings of the IEEE international Conference on Computer Vision. 2017: 2961-2969.
48. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint arXiv:1409.1556, 2014.