Advances in Multimedia

Advanced Visual Analyses for Smart and Autonomous Vehicles

Lead Guest Editor: Zhijun Fang
Guest Editors: Jenq-Neng Hwang and Shih-Chia Huang

Advanced Visual Analyses for Smart and Autonomous Vehicles

Advances in Multimedia

Advanced Visual Analyses for Smart and Autonomous Vehicles

Lead Guest Editor: Zhijun Fang
Guest Editors: Jenq-Neng Hwang and Shih-Chia Huang

Editorial Board

Contents

Hindawi
Advances in Multimedia
Volume 2018, Article ID 1762428, 2 pages
https://doi.org/10.1155/2018/1762428

Editorial

Advanced Visual Analyses for Smart and Autonomous Vehicles

Zhijun Fang,[1] **Jenq-Neng Hwang**,[2] **and Shih-Chia Huang**[3]

[1]*Shanghai University of Engineering Science, Shanghai, China*
[2]*University of Washington, Seattle, USA*
[3]*National Taipei University of Technology, Taipei, Taiwan*

Correspondence should be addressed to Zhijun Fang; zjfang@sues.edu.cn

Received 3 October 2018; Accepted 3 October 2018; Published 1 November 2018

Thanks to the major advances of sensing, communication, and computing, connected and automated vehicle (CAV) technologies have been the strong driving force behind the rapid development of intelligent transportation system (ITS). Advanced visual analyses techniques based on camera/radar/lidar/IR sensing spanning the fields of computer vision, image/video analyses, machine learning, etc. have become the indispensable tools for CAV technologies, especially for enhancing safe and autonomous operation of vehicles in traffic, mainly based on three main components (i.e., environment, vehicle, and driver—EVD) of the overall driving context.

More specifically, advanced visual analyses can infer the dynamic environment scenes outside the vehicles, such as roadway situations and weather conditions, pedestrians, and other vehicles moving trajectories, as well as traffic lights/signs, etc. Advanced visual analyses can also improve the detection accuracies of vehicles' driving states, such as the vibration, speed, acceleration, and abrupt turns. Moreover, advanced visual analyses can serve as nonintrusive monitoring of the drivers which needs to be safely maneuvered in the environment. The complex dynamics of various events and interaction of various "EVD" system components affect the overall safety and comfort of driving, as well as the condition of the traffic flow. Real-time awareness and dynamic response of these system components can proactively result in better driving safety systems and driving experiences, which can accurately, reliably, and very quickly identify the conditions that would lead to an accident and to force corrective actions so that the accident can be prevented.

The goal of this special issue is to share new advanced visual analysis techniques, bring forward challenges, and present comprehensive reviews for CAVs. More specifically, this special issue focuses on state-of-the-art researches of integrating advanced visual analysis techniques, which can be effectively applied to vehicle and driver sensing, road and pedestrian monitoring, data fusion analysis, and correction and response, etc. After several iterations of reviewing processes, five papers are accepted for this special issue, which covers the advance of visual analysis techniques for visual tracking, scene understanding, lane detection, vehicle classification and power controlling, etc.

More specifically, the paper entitled "Robust Visual Tracking with Discrimination Dictionary Learning" proposes an effective tracking algorithm based on learned discrimination dictionary. Based on the learned dictionary, target candidates to be tracked can get a more stable representation. Additionally, the observation likelihood is evaluated based on both the reconstruction error and the dictionary coefficients. The paper entitled "Scene Understanding Based on High-Order Potentials and Generative Adversarial Networks" proposes a scene understanding framework based on a generative adversarial network (GAN) to implement a fully convolutional semantic segmentation model. The high-order potentials are adopted to achieve the fine details and consistency of the segmented semantic image. The paper entitled "Lane Detection based on Connection of Various Feature Extraction Methods" presents a new preprocessing and ROI selection method for lane detection. First, in the preprocessing stage, the RGB color space is converted to the HSV color space and white features on the HSV model are also extracted. At the same time, the preliminary edge feature detection is added in the preprocessing stage, and then the part below the image is selected as the ROI area based

on the proposed preprocessing. The paper entitled "Pretraining Convolutional Neural Networks for Image-based Vehicle Classification" proposes a convolution neural networks (CNN) for image-based vehicle classification with four categories, including motorcycle, transporter, and passenger. An unsupervised pretraining approach is introduced to initialize CNN parameters for better classification performance. Finally, the paper entitled "A Power Control Algorithm Based on Outage Probability Awareness in Vehicular Ad Hoc Networks" addresses a power control algorithm to overcome the problems of random mobility of nodes, interference in multiusers, and high outage probability. The proposed power control method aims at minimizing the outage probability, taking advantage of available cumulative interference at the transmitter of each terminal. Furthermore, the interference model is built by a stochastic geometric theory, from which the expression between outage probability and cumulative interference can be derived.

As we conclude the introduction to this special issue and the contents of the selected five papers, we would like to thank all authors for their valuable contributions. We also express our deep gratitude to all the reviewers for their timely and insightful comments on all submitted papers. It is our sincere expectation that the contents in this special issue are informative and useful from various aspects related to connected and automated vehicle (CAV) technologies.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Zhijun Fang
Jenq-Neng Hwang
Shih-Chia Huang

Hindawi
Advances in Multimedia
Volume 2018, Article ID 3138278, 10 pages
https://doi.org/10.1155/2018/3138278

Research Article

Pretraining Convolutional Neural Networks for Image-Based Vehicle Classification

Yunfei Han,[1,2,3] **Tonghai Jiang**,[1,2,3] **Yupeng Ma**,[1,2,3] **and Chunxiang Xu**[1,2,3]

[1]*The Xinjiang Technical Institute of Physics & Chemistry, Urumqi 830011, China*
[2]*Xinjiang Laboratory of Minority Speech and Language Information Processing, Urumqi 830011, China*
[3]*University of Chinese Academy of Sciences, Beijing 100049, China*

Correspondence should be addressed to Tonghai Jiang; jth@ms.xjb.ac.cn

Received 18 May 2018; Accepted 13 September 2018; Published 2 October 2018

Guest Editor: Zhijun Fang

Vehicle detection and classification are very important for analysis of vehicle behavior in intelligent transportation system, urban computing, etc. In this paper, an approach based on convolutional neural networks (CNNs) has been applied for vehicle classification. In order to achieve a more accurate classification, we removed the unrelated background as much as possible based on a trained object detection model. In addition, an unsupervised pretraining approach has been introduced to better initialize CNNs parameters to enhance the classification performance. Through the data enhancement on manual labeled images, we got 2000 labeled images in each category of motorcycle, transporter, passenger, and others, with 1400 samples for training and 600 samples for testing. Then, we got 17395 unlabeled images for layer-wise unsupervised pretraining convolutional layers. A remarkable accuracy of 93.50% is obtained, demonstrating the high classification potential of our approach.

1. Introduction

Vehicle is one of the greatest inventions in human history. The vehicle has become an indispensable part of modern people's life. The use of a huge large number of vehicles can reflect the population's mobility, intimacy, economic, and so on, and the analysis of vehicle behavior is very meaningful for urban development and government decision-making. In order to collect refueling vehicle information, such as license plate, picture, time, location, volume, type, and so on, we have deployed data collecting equipment in many of refueling stations in Xinjiang, which is mainly responsible for safety supervision and analysis of refueling behavior. Till now, many vehicle profile information such as vehicle color and vehicle type is entered into the system by hand; this is inefficient and not uniform. The accurate, various, and volume of data are the key to dig the value of refueling data. Hence, it has become a problem to be solved urgently that how to obtain the vehicle profile information through the vehicle picture automatically. In this paper, we focus on how to get the vehicle type from the picture. This problem is regarded as image classification, which means we should classify the images containing vehicles into the right type by image processing. Due to the environment in which images are taken is quite varied and complex and the impact of irrelevant background, the vehicles in images are very difficult to recognize.

Thanks to the success of deep learning, we present a combination of approaches for vehicle detection and classification based on convolutional neural networks in this paper. To detect the vehicle in the image more efficiently, a successful object detection approach is used to detect the objects in an image, then the target vehicle waiting for entering refueling station is filtered out. Next, we designed a convolutional neural networks which contains 4 convolutional layers, 3 max pooling layers, and 2 full connected layers for vehicle classification. We trained our model on labeled vehicles images dataset. Comparing it with other five state-of-the-art approaches verified our approach achieves the highest accuracy than others. In order to pursue better classification performance, we taken advantage of unsupervised pretraining to better initialize classification model parameters under the circumstance of a shortage of labeled images. The unsupervised pretraining method was implemented based on deconvolution. After pretraining, the convolutional layers

were initialized by the pretrained parameters and trained the model on our labeled images data set; thus, we got a better classification performance than the previous without pretraining.

This paper is organized as follows. The related works are introduced in Section 2. Vehicle detection and classification based on CNNs and a pretraining approach are described in Section 3. In Section 4, the vehicles data set is presented, and we evaluated the presented approaches on our data, and the experimental results and a performance evaluation are given. Finally, Section 5 concludes the paper.

2. Related Works

Existing methods use various types of signal for vehicle detection and classification, including acoustic signal [2–5], radar signal [6, 7], ultrasonic signal [8], infrared thermal signal [9], magnetic signal [10], 3D lidar signal [11] and image/video signal [12–16]. Furthermore, some of the methods can be combined with a variety of signals, such as radar&vision signal [7] and audio&vision signal [17]. Usually, the detection and classification performance is excellent in these methods because of the precise signal data, but there are many hardware devices involved in these methods, resulting in larger deployment cost and even higher failure rate.

The evolution of image processing techniques, together with wide deployment of surveillance cameras, facilitates image-based vehicle detection and classification. Various approaches to image-based vehicle detection and classification have been proposed over the last few years. Kazemi et al. [13] used 3 different kinds of feature extractors, Fourier transform, Wavelet transform, and Curvelet transform, to recognize and classify 5 models of vehicles; k-nearest neighbor is used as classifier. They compare the 3 proposed approaches and find that the Curvelet transform can extract better features. Chen et al. [18] presented a system for vehicle detection, tracking, and classification from roadside closed circuit television (CCTV). First, a Kalman filter tracked a vehicle to enable classification by majority voting over several consecutive frames, then they trained a support vector machine (SVM) using a combination of a vehicle silhouette and intensity-based pyramid histogram of oriented gradient (HOG) features extracted following background subtraction, classifying foreground blobs with majority voting. Wen et al. [19] used Haar-like feature pool on a $32*32$ grayscale image patch to represent a vehicle's appearance and then proposed a rapid incremental learning algorithm of AdaBoost to improve the performance of AdaBoost. Arrospide and Salgado [16] analyzed the individual performance of popular techniques for vehicle verification and found that classifiers based on Gabor and HOG features achieve the best results and outperform principal component analysis (PCA) and other classifiers based on features as symmetry and gradient. Mishra and Banerjee [20] detected vehicle using background, extracted Haar, pyramidal histogram of oriented gradients, shape and scale-invariant feature transform features, designed a multiple kernel classifier based on k-nearest neighbor to divide the vehicles into 4 categories. Tourani and Shahbahrami [21] combined different image/video processing methods including object detection, edge detection, frame differentiation, and Kalman filter to propose a method which resulted in about 95 percent accuracy for classification and about 4 percent error in vehicle detection targets. In these methods, the classification results are very good; however, there are still some problems. First, the image features are limited by the hand-crafted features algorithms to represent rich information. Second, the hand-crafted features algorithms require a lot of calculation, so they are not suitable for real-time applications, especially for embedding in front-end camera devices. Third, most of them are used in fixed scenes and background environments; it is difficult for them to deal with complex environment.

More recently, deep learning has become a hot topic in object detection and object classification area. Wang et al. [22] proposed a novel deep learning based vehicle detection algorithm with 2D deep belief network; the 2D-DBN architecture uses second-order planes instead of first-order vector as input and uses bilinear projection for retaining discriminative information so as to determine the size of the deep architecture which enhances the success rate of vehicle detection. Their algorithm preforms very good in their datasets. He et al. [1] proposed a new efficient vehicle detection and classification approach based on convolutional neural network, the features extracted by this method outperform those generated by traditional approaches. Yi et al. [23] proposed a deep convolution network based on pretrained AlexNet model for deciding whether a certain image patch contains a vehicle or not in Wide Area Motion Imagery (WAMI) imagery analysis. Li et al. [24] presented the 3D range scan data in a 2D point map and used a single 2D end-to-end fully convolutional network to predict the vehicle confidence and the bounding boxes simultaneously, and they got the state-of-the-art performance on the KITTI dataset.

Meantime, object detection and classification based on Convolutional neural networks (CNNs) [25–27] are very successful in the field of computer vision recently. The first work about object detection and classification based on deep learning has been done in 2013; Sermanet et al. [28] present an integrated framework for using deep learning for object detection, localization, and classification; this framework obtains very competitive results for the detection and classifications tasks. Up to now, excellent object detection and classification models based on deep learning include R-CNN [29], Fast R-CNN [30], YOLO [31], Faster R-CNN [32], SSD [33], and R-FCN [34]; these models achieve state-of-the-art results on several data sets. Before the YOLO, many approaches on object detection, for example, R-CNN and Faster R-CNN, repurpose classifiers to perform detection. Instead, YOLO frame object detection is a regression problem to separated bounding boxes and associated class probabilities. The YOLO framework uses a custom network based on the Googlenet architecture, using 8.52 billion operations for a forward pass. However, a more recent improved model called YOLOv2 [35] achieves comparable results on standard tasks like PASCAL VOC and COCO. In YOLOv2 network, it uses a new model, called Darknet-19, and has 19 convolutional layers and 5 maxpooling layers; the model only requires 5.58

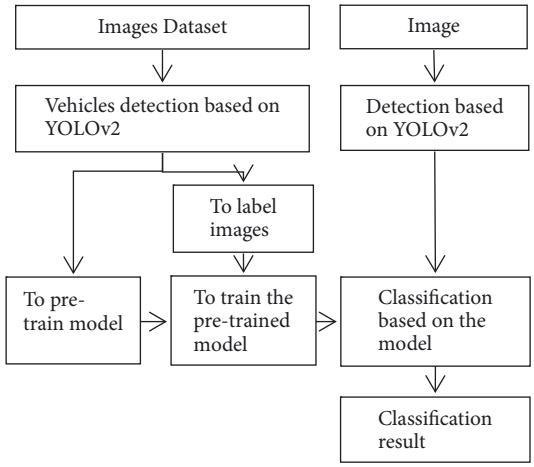


Figure 1: Flowchart of vehicle classification.

billion operations. In conclusion, YOLOv2 is a state-of-the-art detection system, it is better, faster, and stronger than others and applied for object detection tasks in this work.

At last, unsupervised pretraining initializes the model to a point in the parameter space that somehow renders the optimization process more effective, in the sense of achieving a lower minimum of the empirical loss function [36]. Much recent research has been devoted to learning algorithms for deep architectures such as Deep Belief Networks [37, 38] and stacks of autoencoder variants [39]. After vehicle detection, we can easily get a lot of unlabeled images of vehicles and optimize the classification model parameters initialization by unsupervised pretraining.

3. Methodology

In this section, we will present the details of the method based on CNNs for image-based vehicle detection and vehicle classification. This section contains three parts: vehicle detection, vehicle classification, and pretraining approach. The relations between each part and the overall framework of the entire idea is shown in Figure 1.

3.1. Vehicle Detection. Our images taken from static cameras in different refueling station contain front views of vehicles or side views of vehicles at any point. The vehicles in images are very indeterminacy; this makes the vehicle detection more difficult in traditional methods based on hand-crafted features.

The YOLOv2 model is trained on COCO data sets, it can detect 80 common objects in life, such as person, bicycle, car, bus, train, truck, boat, bird, cat, etc., and therefore we can perform vehicle detection based on YOLOv2. In a picture of the vehicle which is waiting for entering the refueling station shown in Figure 2, YOLOv2 can well detect many objects, for example, the security guards, drivers, the vehicle, and queued vehicles, and even vehicles on the side of road. Here, our goal is to pick up the vehicle which is waiting for entering in picture.

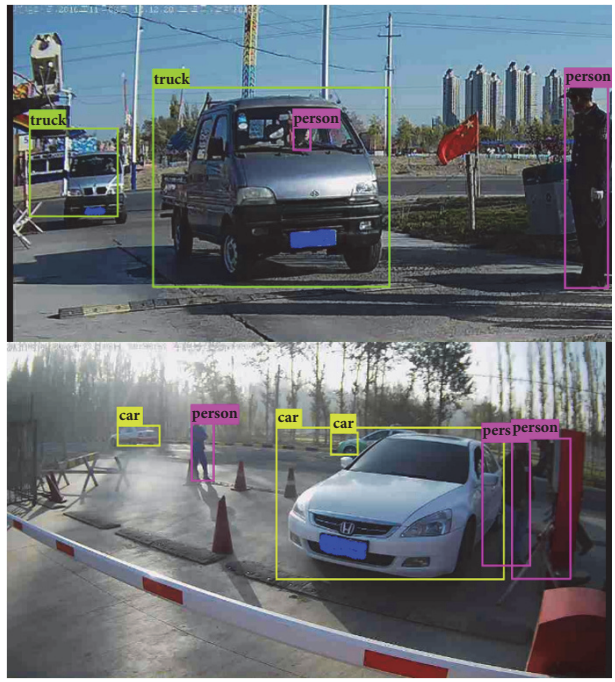


Figure 2: YOLOv2 detection. Top: truck. Down: car.

Although trained YOLOv2 can detect the vehicle and divide it into bicycle, car, motorbike, bus, and truck, it does not meet our classification categories. In order to solve this problem, to fine-tune the YOLOv2 on our data could be a solution, but this method needs amount of manual labeled data and massive computation, it is not a preferable method for us, and then, we presented a rule-based method to detect four categories vehicles more accurately. First, from the YOLOv2 detection results, we select the objects which are very similar to our targets, such as car, bus, truck, and motorbike; second, according to the distance between the vehicle and the camera, the closer the vehicle is, the bigger the target is, we choose the most similar vehicle in picture as the target vehicle entering the refueling station for further vehicle behavior analysis.

3.2. Vehicle Classification. According to the function and size of vehicles, vehicles will be divided into four categories of motorcycle, transporter, passenger, and others. Motorcycle includes motorcycle and motor tricycle; transporter includes truck and container car; passenger includes sedan, hatchback, coupe, van, SUV, and MPV; others include vehicles used in agricultural production and infrastructure, such as tractor and crane, and other types of vehicles. Figure 3 shows the sheared samples in four categories in each column. As we can see, samples images are very different in shape, color, size, and angle of camera, even the samples images in the same category. And Figure 3(b) bottom and Figure 3(c) bottom are not in the same category, but they are very similar, especially on front face, shape, and color, which makes the classification between transporter and passenger more difficult.

(a) (b) (c) (d)

Figure 3: Sheared vehicle image examples for four categories. (a) Motorcycle; (b) transporter; (c) passenger; (d) others.

Table 1: The structure of C4M3F2.

Layer Type/Activation	Size/Stride	Filters
Convolutional/ReLU	3×3/1	32
Max Pooling	2×2/2	
Convolutional/ReLU	3×3/1	64
Convolutional/ReLU	3×3/1	64
Max Pooling	2×2/2	
Convolutional/ReLU	3×3/1	64
Max Pooling	2×2/2	
Fully Connected/ReLU	1024	
Fully Connected	1024	
Softmax	4	

To solve this difficult problem, we presented a convolutional classification model which is effective and requires little amount of operations. Our model, called C4M3F2, has 4 convolutional layers, 3 max pooling layers, and 2 fully connected layers.

Each convolutional layer contains multiple (32 or 64) 3∗3 kernels, and each kernel represents a filter connected to the outputs of the previous layer. Each max pooling layer contains multiple max pooling with 2×2 filters and stride 2; it effectively reduces the feature dimension and avoids overfitting. For fully connected layers, each layer contains 1024 neurons, each neuron makes prediction from its all input, and it is connected to all the neurons in previous layers. For each sheared vehicle image detected from YOLOv2, it has been resized to 48×48 and then passed into C4M3F2. Eventually, all the features are passed to softmax layer, and what we need to do is just minimizing the cross entropy loss between softmax outputs and the input labels. Table 1 shows the structure of our model C4M3F2.

3.3. Pretraining Approach. With the purpose of achieving a satisfactory classification result, we need more labeled images for training our model, but there is a shortage of labeled images; however, there are plenty of images collected easily, and how to use the plenty of unlabeled images for optimization of our classification model has become the main content in this subsection.

The motivation of this unsupervised pretraining approach is to optimize the parameters in convolutional kernel

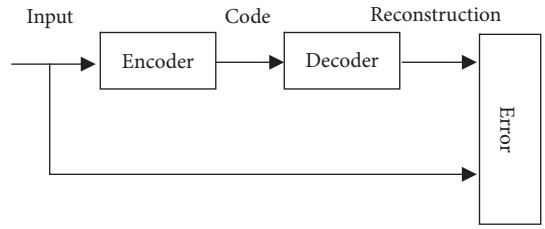


Figure 4: Autoencoder.

parameters. Kernel training in C4M3F2 starts from a random initialization, and we hope that the kernel training procedure can be optimized and accelerated using a good initial value obtained by unsupervised pretraining. In addition, pretraining initializes the model to a point in the parameter space that somehow renders the optimization process more effective, in the sense of achieving a lower minimum of the loss function [36]. Next, we will explain how to greedy layer-wise pretrain the convolution layers and the fully connected layers in the C4M3F2 model. Max pooling layer function is subsampling; it is not included in layer-wise pretraining process.

An autoencoder [40] neural network is an unsupervised learning algorithm that applies backpropagation, setting the target values to be equal to the inputs. It uses a set of recognition weights to convert the input into code and then uses a set of generative weights to convert the code into an approximate reconstruction of the input. Autoencoder must try to reconstruct the input, which aims to minimize the reconstruction error as shown in Figure 4.

According to the aim of autoencoder, our approach for unsupervised pretraining will be explained. In every convolutional layer, convolution can be regarded as the encoder, and deconvolution [41] is taken as the decoder, which is a very unfortunate name and also called transposed convolution. An input image is passed into the encoder, then the output code getting from encoder is passed into the decoder to reconstruct the input image. Here, Euclidean distance, which means the reconstruct error, is used to measure the similarity between the input image and reconstructed image, so the aim of our approach is minimizing the Euclidean paradigm. For pretraining the next layer, first, we should drop the decoder and freeze the weights in the encoder of previous layers and then take the output code of previous layer as the input in this

layer and do the same things as in previous layer. Next how to use transposed convolution to construct and minimize the loss function in one convolutional layer will be described in detail as follows.

The convolution of a feature maps and an image can be defined as

$$y_{ij} = w_i \oplus x_j \tag{1}$$

where $\oplus$ denotes the 2D convolution, $y_{ij} \in R^{r_x \times c_x}$ is the convolution result and the padding is set to keep the input and output dimensions consistent. $w_i \in R^{r_w \times c_w}$ is the ith kernel, and $x_j \in R^{r_x \times c_x}$ denotes the jth training image.

Then, transform the convolution based on circulant matrix in linear system. A circulant matrix is a special kind of Toeplitz matrix where each row vector is rotated one element to the right relative to the preceding row vector. An $n \times n$ circulant matrix C takes the form in

$$C = \begin{bmatrix} c_0 & c_{n-1} & \cdots & c_2 & c_1 \\ c_1 & c_0 & & c_3 & c_2 \\ \vdots & & \ddots & \vdots & \\ c_{n-2} & c_{n-3} & & c_0 & c_{n-1} \\ c_{n-1} & c_{n-2} & \cdots & c_1 & c_0 \end{bmatrix} \tag{2}$$

Let $f_j, h_i \in R^{r_e \times c_e}$ be the extension of x_j, w_i, where $r_e = r_x + r_w - 1, c_e = c_x + c_w - 1$. And the method is as follows (3) and (4), where O is zero matrix:

$$f_j = \begin{bmatrix} x_j & O \\ O & O \end{bmatrix} \tag{3}$$

$$h_i = \begin{bmatrix} w_i & O \\ O & O \end{bmatrix} \tag{4}$$

Let $f_j^v \in R^{r_e c_e \times 1}$ be f_j in vectored form, row_a be a row of h_i, and $row_a = [n_0, n_1, n_2, \cdots, n_{c_e-1}]$. To build circulant matrices $H_0, H_1, H_2, \cdots, H_{c_e-1}$ by row_a and by these circulant matrices, a block circulant matrix is defined as shown in formula (5).

$$H = \begin{bmatrix} H_0 & H_{c_e-1} & \cdots & H_2 & H_1 \\ H_1 & H_0 & & H_3 & H_2 \\ \vdots & & \ddots & \vdots & \\ H_{c_e-2} & H_{c_e-3} & & H_0 & H_{c_e-1} \\ H_{c_e-1} & H_{c_e-2} & \cdots & H_1 & H_0 \end{bmatrix} \tag{5}$$

Here, we can transform the convolution into (6).

$$Q = Hf_j^v \tag{6}$$

Q is the vector form of result of convolution calculation and then to reshape Q into $Q' \in R^{r_e \times c_e}$. In this convolution process, the padding is dealing with filling 0, but in actual implementation of our approach, we keep the convolutional input and output dimensions consistent. So we need to prune the extra values to keep the input and output dimensions consistent. So we intercept the matrix Q', then $y_{ij} = Q'[r_w - 1 : r_e - r_w + 1, c_w - 1 : c_e - c_w + 1]$.

To simplify the calculation, we extract the effective rows of H according to the effective row indexes which indicates the position of the elements of y_{ij} in Q and denote these rows by $W_i \in R^{(r_e-2r_w+2)(c_e-2c_w+2) \times 2c_e}$.

Now, $Y_{ij} \in R^{(r_e-2r_w+2)(c_e-2c_w+2) \times 1}$ is vector form of y_{ij}, so the convolution can be rewritten as

$$Y_{ij} = W_i f_j^v \tag{7}$$

There are J training vehicle images and K kernels. Let $X = [f_1^v, f_2^v, \ldots, f_J^v]$, and $W = [W_1; W_2; \ldots; W_K]$.

The convolution can be calculated as

$$Y = \mathrm{W}X \tag{8}$$

And the deconvolution can be calculated

$$X' = W^T Y \tag{9}$$

So X' is the X reconstruction. Then loss function based on Euclidean paradigm is defined in formula (10), being

$$\text{loss}(W, X) = \left\| X - X' \right\|_2 = \sqrt{\sum_{j=1}^{J} \left(X_j - X_j' \right)^2} \tag{10}$$

Then, we used adam optimizer, which is an algorithm for first-order gradient-based optimization of stochastic objective functions based on adaptive estimates of lower-order moments, to solve the minimum optimization problem in formula (10).

After the greedy layer-wise unsupervised pretraining, we initiate the parameters in every convolutional layer with the pretrained values and run the supervised training for classification according to the method in previous subsection.

4. Experiments and Discussions

We evaluated the presented algorithm on our data and compared it with other four state-of-the-art methods.

4.1. Datasets and Experiment Environment. The vehicles images are taken by the static cameras in different refuel stations; after being compressed, they are sent to servers. The quality of images on servers is lower than that selected by random and classified into four categories of motorcycle, transporter, passenger, and others by hand. We got 498 motorcycle images, 1109 transporter images, 1238 passenger images, and 328 other images. Due to the time-consuming and labor-intensive of manual labeling, there is a shortage of labeled images. Image augmentation has been used to enrich the data. Keras, the excellent high level neural network API, provides the ImageDataGenerator for image data preparation and augmentation. Shear range is set to -0.2 to 0.2, zoom range is set to -0.2 to 0.2, rotation range is set to -7 to 7, size is set to 256×256, and the points outside the boundaries are

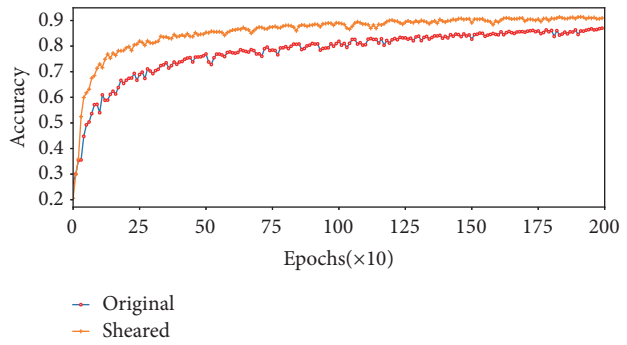


Figure 5: Comparison of training process between original and sheared.

filled according to the nearest mode. After configuration and taking into account the balance of data, we fitted it on our data and got 1400 samples for every categories on the training set and 600 samples for every categories on the testing set to assess our classification model.

For vehicle classification, the CNNs are under Tensorflow framework, the SIFT is under OpenCV (https://opencv.org/), and other feature embedding methods are under scikit-image (http://scikit-image.org/). All the experiments were conducted on a regular notebook PC (2.5-GHz 8-core CPU, 12-G RAM, and Ubuntu 64-bit OS).

4.2. Vehicle Detection Experiment with YOLOv2. For the experiment using original images, the original training set and test set are used for training and testing, for the experiment using sheared images, we used the approach based on trained YOLOv2 to detect the original training set and test set to get the sheared training set and sheared testing set for training and testing.

To verify the importance of vehicle detection for vehicle classification, we designed two groups of vehicle classification experiments, one using original images and the other one using sheared images after vehicle detection, then, the C4M3F2 model is used for vehicle classification experiments.

We initialized the C4M3F2 model by truncated normal distribution, fitted the model on original training set and sheared training set for 2000 epochs, respectively, and recorded the accuracy of our C4M3F2 model on different testing set; the results are shown in Figure 5. As we expected, the sheared images of vehicles more accurately represent the characteristics of vehicles, while pruning more useless information and facilitating the feature extraction and vehicle classification. As we can see in Figure 5, the accuracy of C4M3F2 model using sheared data set is much better than the one using original data set; in the previous training, the characteristics of vehicles were extracted more accurately, so that, the model quickly achieved a better classification results and a stable status.

Finally, the accuracy of C4M3F2 using sheared data set is 91.42%, which is 4.53% higher than the 86.89% of C4M3F2 using original data set. It can be concluded that the results of vehicle classification using sheared data set after vehicle detection based on YOLOv2 can be improved effectively.

Table 2: Accuracy and FPS of different methods.

Method	Accuracy	FPS
HOG+SVM	60.12%	4
DAISY+SVM	69.04%	2
ORB+BoW+SVM	64.07%	7
SIFT+BoW+SVM	74.49%	5
DeCAF[1]	66.20%	13
CNNs	91.42%	800

4.3. Compare Our Approach with Others. There are many other image classification methods. To assess our classification model, we compared our approach with other five methods.

The five methods are based on the image features defined by the scholars in computer image processing. Considering the comprehensive factors, four kinds of image features and a convolutional method are selected, they are histograms of oriented gradient (HOG) [42], DAISY [43], oriented FAST, and rotated BRIEF (ORB)[44], scale-invariant feature transform (SIFT) [45], and DeCAF[1] respectively. These methods are excellent in target object detection in [1, 42–45]. HOG is based on computing and counting the gradient direction histogram of local regions. DAISY is a fast computing local image feature descriptor for dense feature extraction, and it is based on gradient orientation histograms similar to the SIFT descriptor. ORB uses an oriented FAST detection method and the rotated BRIEF descriptors; unlike BRIEF, ORB is comparatively scale and rotation invariant while still employing the very efficient Hamming distance metric for matching. SIFT is the most widely used algorithm of key point detection and description. It takes full advantage of image local information. SIFT feature has a good effect in rotation, scale, and translation, and it is robust to changes in angle of view and illumination; these features are beneficial to the effective expression of targets information. For HOG and DAISY, image features regions are designed; features are computed and sent into SVM classifier to be classified. For ORB and SIFT, they do not have acquisition features regions and specified number of features; we get the image features based on Bag-of-Words (BoW) model by treating image features as words. In the first instance, all features points of all training build the visual vocabulary; in the next place, a feature vector of occurrence counts of the vocabulary is constructed from an image; in the end, the feature vector is sent into SVM classifier to be classified. DeCAF uses five convolutional layers and two fully connected layers to extract features and a SVM to classify the image into the right group [1].

Here, we performed vehicle classification experiments on sheared data set. Table 2 shows the accuracy and FPS of CNNs and other state-of-the-art methods in test; other methods are very slow because they take a lot of time to extract features. It can be observed that the results show the effectiveness of CNNs on vehicle classification problem.

From another point of view, we demonstrated the classification ability of each method by confusion matrix of the

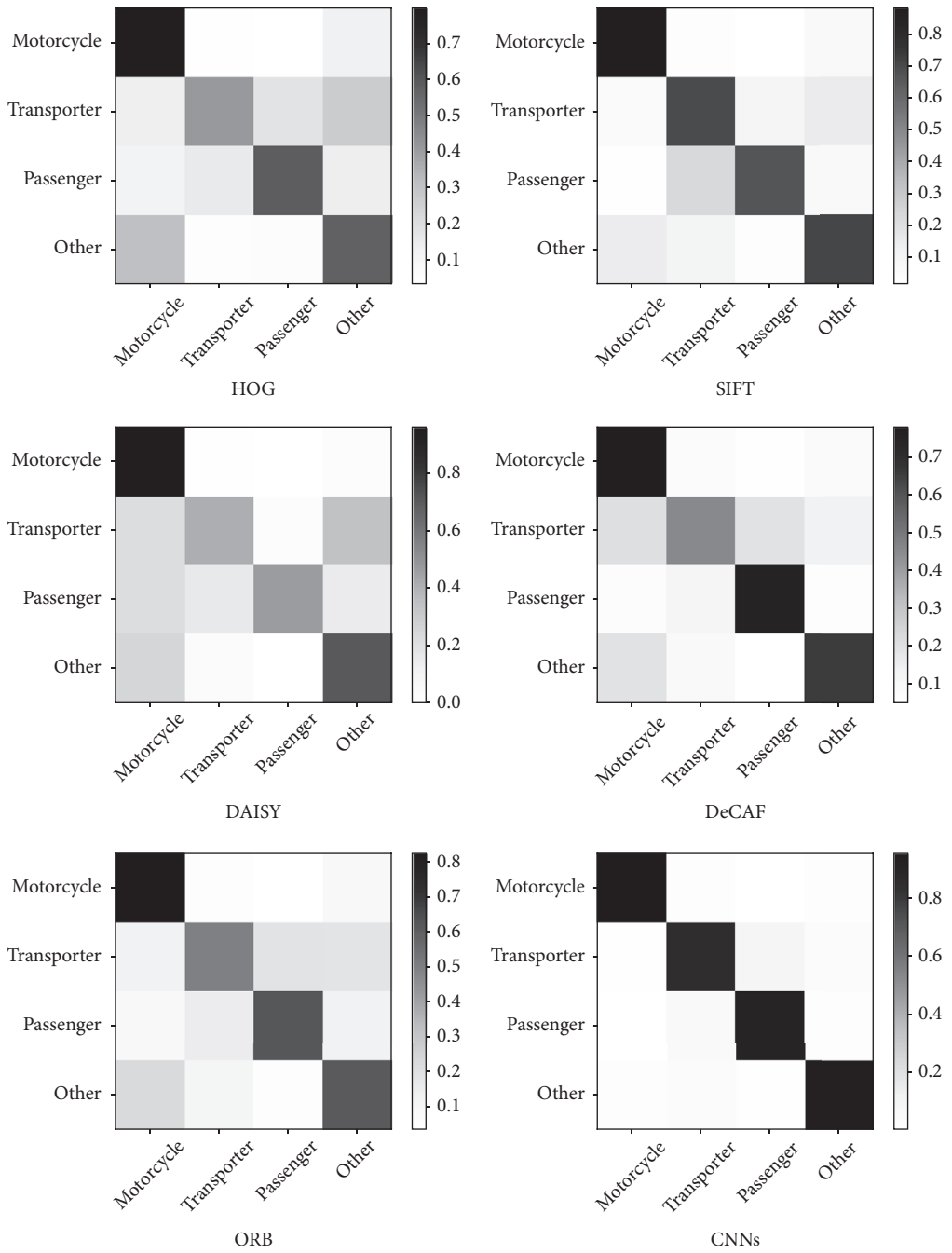


Figure 6: Confusion matrix of classification of different methods.

classification process from the five methods in Figure 6. The main diagonal displays the high recognition accuracy. As shown in Figure 6, the top five comparative methods are better in recognition of motorcycle than other categories. Generally speaking, ORB or SIFT combined with BoW and SVM method is a little better than the other two methods. All taken into account, our CNNs method is the best. But, the performance of CNNs turns out not so satisfactory in view than the confusion of transporter and passenger.

Next, we will focus on the reason of confusion in CNNs. According to the precision, recall, and f1-score of classification in Table 3, it shows that the identification of motorcycle is very good enough, and the identification of transporter and passenger is relatively poor.

As shown in the examples in Figure 7, it can be seen that the wrongly recognized transporter and passenger images include vehicle face information mainly and very little vehicle body information, as far as the main vehicle face is concerned;

Figure 7: Examples of wrongly recognized vehicles images. First row: wrongly recognized as passenger which should be transporter. Second row: wrongly recognized as transporter which should be passenger.

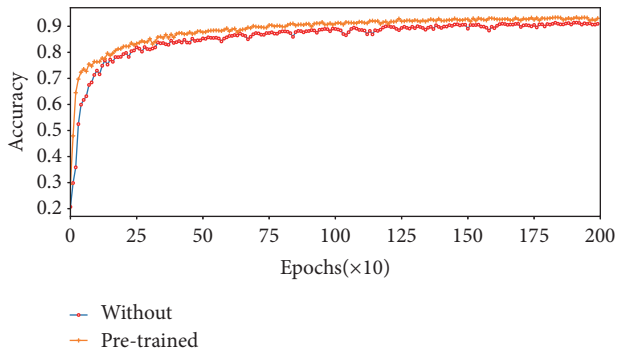


Figure 8: Comparison of training process between pretrained and without pretraining.

Table 3: Classification precision, recall, and f1-score of CNNs.

Type	Precision	Recall	F1-score
Motorcycle	0.97	0.95	0.96
Transporter	0.87	0.85	0.86
Passenger	0.90	0.91	0.90
Other	0.93	0.93	0.93

these vehicles images are so similar in profile that it is still a challenge to recognize same images in Figure 7 manually.

4.4. Pretraining Approach Experiment. We are eager for better performance of our classification model C4M3F2. Here, unsupervised pretraining has been used for optimization of our classification model. 17395 sheared vehicles images are obtained by shearing the unlabeled vehicles images.

We pretrained the parameters of every convolutional layer for 2000 epochs and then supervised training the model on our sheared training set and tested it on our sheared testing set; the results is shown in Figure 8. In the process of training, the conclusion is that the effect is more obvious in the previous epochs, and the overall training process is stable relatively. Ultimately, the accuracy of pretrained CNNs is 93.50%, which is 2.08% higher than the 91.42% of CNNs without pretraining.

Table 4: Classification precision, recall, and f1-score of pretrained CNNs based on sheared dataset.

Type	Precision	Recall	F1-score
Motorcycle	0.99	0.99	0.99
Transporter	0.90	0.99	0.99
Passenger	0.91	0.92	0.92
Other	0.95	0.96	0.95

Table 5: Classification precision, recall, and f1-score of pretrained CNNs based on original dataset.

Type	Precision	Recall	F1-score
Motorcycle	0.99	0.99	0.99
Transporter	0.79	0.82	0.81
Passenger	0.84	0.79	0.81
Other	0.90	0.94	0.92

Table 6: The classification accuracy under different strategies.

	Original	Sheared
CNNs Without pre-training	86.89%	91.42%
CNNs with pre-training	88.29%	93.5%

By analyzing the classification performance of pretrained CNNs, shown in Table 4, we can draw a conclusion that its performance is better than the one of CNNs without pretraining shown in Table 3, especially for the classification of transporter and passenger. In summary, pretrained CNN is more effective in recognizing the vehicles categories, and it is a state-of-the-art approach for vehicle classification.

In the end, to verify the effect of detection in the entire system, we conducted ablation study by pretraining and testing our model on the original dataset which has not been sheared by YOLOv2 and contains a large quantity of irrelevant background. Ultimately, the accuracy of pretrained CNNs on original dataset is 88.29%, which is 5.21% lower than the 93.5% of pretrained CNNs on sheared dataset and even lower than the 91.42% of CNNs without pretraining on sheared dataset; the classification performance is shown in Table 5. And according the classification accuracy in Table 6, we can conclude that this ablation study confirms the essentiality of detection virtually in the whole vehicle classification system.

5. Conclusions

A classification method based on CNNs has been detailed in this paper. To improve the accuracy, we used vehicle detection to removing the unrelated background for facilitating the feature extraction and vehicle classification. Then, an autoencoder-based layer-wise unsupervised pretraining is introduced to improve the CNNs model by enhancing the classification performance. Several state-of-the-art methods have been evaluated on our labeled data set containing four categories of motorcycle, transporter, passenger, and others.

Experimental results have demonstrated that the pretrained CNNs method based on vehicle detection is the most effective for vehicle classification.

In addition, the success of our vehicle classification makes a vehicle color or logo recognition system possible in our refueling behavior analysis; meanwhile, it is a great help to urban computing, intelligent transportation system, etc.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research is supported by Youth Innovation Promotion Association CAS (2015355). The authors gratefully acknowledge the invaluable contribution of Yupeng Ma and the members of his laboratory during this collaboration.

References

[1] D. He, C. Lang, S. Feng, X. Du, and C. Zhang, "Vehicle detection and classification based on convolutional neural network," in *Proceedings of the 7th International Conference on Internet Multimedia Computing and Service*, 2015.

[2] J. F. Forren and D. Jaarsma, "Traffic monitoring by tire noise," *Computer Standards & Interfaces*, vol. 20, pp. 466-467, 1999.

[3] J. George, A. Cyril, B. I. Koshy, and L. Mary, "Exploring Sound Signature for Vehicle Detection and Classification Using ANN," *International Journal on Soft Computing*, vol. 4, no. 2, pp. 29–36, 2013.

[4] J. George, L. Mary, and K. S. Riyas, "Vehicle detection and classification from acoustic signal using ANN and KNN," in *Proceedings of the 2013 International Conference on Control Communication and Computing, (ICCC '13)*, pp. 436–439, Thiruvananthapuram, India, 2013.

[5] K. Wang, R. Wang, Y. Feng et al., "Vehicle recognition in acoustic sensor networks via sparse representation," in *Proceedings of the 2014 IEEE International Conference on Multimedia and Expo Workshops, (ICMEW '14)*, pp. 1–4, Chengdu, China, 2014.

[6] A. Duzdar and G. Kompa, "Applications using a low-cost baseband pulsed microwave radar sensor," in *Proceedings of the 18th IEEE Instrumentation and Measurement Technology Conference, (IMTC '01)*, vol. 1 of *Rediscovering Measurement in the Age of Informatics*, pp. 239–243, IEEE, Budapest, Hungary, 2001.

[7] H.-T. Kim and B. Song, "Vehicle recognition based on radar and vision sensor fusion for automatic emergency braking," in *Proceedings of the 13th International Conference on Control, Automation and Systems, (ICCAS '13)*, pp. 1342–1346, Gwangju, Republic of Korea, 2013.

[8] Y. Jo and I. Jung, "Analysis of vehicle detection with wsn-based ultrasonic sensors," *Sensors*, vol. 14, no. 8, pp. 14050–14069, 2014.

[9] Y. Iwasaki, M. Misumi, and T. Nakamiya, "Robust vehicle detection under various environments to realize road traffic flow surveillance using an infrared thermal camera," *The Scientific World Journal*, vol. 2015, Article ID 947272, 2015.

[10] J. Lan, Y. Xiang, L. Wang, and Y. Shi, "Vehicle detection and classification by measuring and processing magnetic signal," *Measurement*, vol. 44, no. 1, pp. 174–180, 2011.

[11] B. Li, T. Zhang, and T. Xia, "Vehicle Detection from 3D Lidar Using Fully Convolutional Network," https://arxiv.org/abs/1608.07916, 2016.

[12] V. Kastrinaki, M. Zervakis, and K. Kalaitzakis, "A survey of video processing techniques for traffic applications," *Image and Vision Computing*, vol. 21, no. 4, pp. 359–381, 2003.

[13] F. M. Kazemi, S. Samadi, H. R. Poorreza, and M.-R. Akbarzadeh-T, "Vehicle recognition based on fourier, wavelet and curvelet transforms - A comparative study," in *Proceedings of the 4th International Conference on Information Technology-New Generations, ITNG '07*, pp. 939-940, Las Vegas, Nev, USA, 2007.

[14] J. Y. Ng and Y. H. Tay, "Image-based Vehicle Classification System," https://arxiv.org/abs/1204.2114, 2012.

[15] R. A. Hadi, G. Sulong, and L. E. George, "Vehicle Detection and Tracking Techniques: A Concise Review," *Signal & Image Processing: An International Journal*, vol. 5, no. 1, pp. 1–12, 2014.

[16] J. Arróspide and L. Salgado, "A study of feature combination for vehicle detection based on image processing," *The Scientific World Journal*, vol. 2014, Article ID 196251, 13 pages, 2014.

[17] P. Piyush, R. Rajan, L. Mary, and B. I. Koshy, "Vehicle detection and classification using audio-visual cues," in *Proceedings of the 3rd International Conference on Signal Processing and Integrated Networks, (SPIN '16)*, pp. 726–730, Noida, India, 2016.

[18] Z. Chen, T. Ellis, and S. A. Velastin, "Vehicle detection, tracking and classification in urban traffic," in *Proceedings of the 15th International IEEE Conference on Intelligent Transportation Systems, ITSC '12*, pp. 951–956, Anchorage, Alaska, USA, 2012.

[19] X. Wen, L. Shao, Y. Xue, and W. Fang, "A rapid learning algorithm for vehicle classification," *Information Sciences*, vol. 295, pp. 395–406, 2015.

[20] P. Mishra and B. Banerjee, "Multiple Kernel based KNN Classifiers for Vehicle Classification," *International Journal of Computer Applications*, vol. 71, no. 6, pp. 1–7, 2013.

[21] A. Tourani and A. Shahbahrami, "Vehicle counting method based on digital image processing algorithms," in *Proceedings of the 2nd International Conference on Pattern Recognition and Image Analysis, IPRIA '15*, pp. 1–6, Rasht, Iran, 2015.

[22] H. Wang, Y. Cai, and L. Chen, "A vehicle detection algorithm based on deep belief network," *The Scientific World Journal*, vol. 2014, Article ID 647380, 7 pages, 2014.

[23] M. Yi, F. Yang, E. Blashch et al., "Vehicle Classification in WAMI Imagery using Deep Network," in *Proceedings of the SPIE 9838: Sensors and Systems for Space Applicatioins IX*, 2016.

[24] B. Li, T. Zhang, and T. Xia, "Vehicle detection from 3D lidar using fully convolutional network," in *Proceedings of the Robotics: Science and Systems*, 2016.

[25] Y. Lecun, B. Boser, J. S. Denker et al., "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989.

[26] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2323, 1998.

[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Neural Information Processing Systems*, pp. 1097–1105, 2012.

[28] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. Lecun, "OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks," https://arxiv.org/abs/1312.6229, 2013.

[29] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '14)*, pp. 580–587, Columbus, Ohio, USA, 2014.

[30] R. Girshick, "Fast R-CNN," in *Proceedings of the 15th IEEE International Conference on Computer Vision (ICCV '15)*, pp. 1440–1448, Santiago, Chile, 2015.

[31] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, (CVPR '16)*, pp. 779–788, Las Vegas, Nev, USA, 2016.

[32] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[33] W. Liu, D. Anguelov, D. Erhan et al., "SSD: single shot multibox detector," in *Proceedings of the Computer Vision – ECCV 2016*, vol. 9905 of *Lecture Notes in Computer Science*, pp. 21–37, 2016.

[34] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object Detection via Region-based Fully Convolutional Networks," https://arxiv.org/abs/1605.06409, 2016.

[35] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition, (CVPR '17)*, pp. 6517–6525, Honolulu, Hawaii, USA, 2017.

[36] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio, "Why does unsupervised pre-training help deep learning?" *Journal of Machine Learning Research*, vol. 11, pp. 625–660, 2010.

[37] G. E. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[38] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," *Neural Information Processing Systems*, pp. 153–160, 2007.

[39] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, "Stacked Convolutional Auto-Encoders for Hierarchical Feature Extraction," in *Proceedings of the Artificial Neural Networks and Machine Learning - ICANN 2011*, vol. 6791 of *Lecture Notes in Computer Science*, pp. 52–59, Springer, Berlin, Heidelberg, Germany, 2011.

[40] G. E. Hinton and R. S. Zemel, "Autoencoders, minimum description length and helmholtz free energy," *Neural Information Processing Systems*, pp. 3–10, 1994.

[41] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, "Deconvolutional networks," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR' 10)*, pp. 2528–2535, San Francisco, Calif, USA, 2010.

[42] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, vol. 1, pp. 886–893, 2005.

[43] E. Tola, V. Lepetit, and P. Fua, "DAISY: an efficient dense descriptor applied to wide-baseline stereo," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 5, pp. 815–830, 2010.

[44] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: an efficient alternative to SIFT or SURF," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '11)*, pp. 2564–2571, Barcelona, Spain, 2011.

[45] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the 7th IEEE International Conference on Computer Vision (ICCV '99)*, vol. 2, pp. 1150–1157, Kerkyra, Greece, 1999.

Hindawi
Advances in Multimedia
Volume 2018, Article ID 7357284, 10 pages
https://doi.org/10.1155/2018/7357284

Research Article

Robust Visual Tracking with Discrimination Dictionary Learning

Yuanyun Wang,[1,2] Chengzhi Deng,[1] Jun Wang,[1,2] Wei Tian,[1] and Shengqian Wang[1]

[1]*Jiangxi Province Key Laboratory of Water Information Cooperative Sensing and Intelligent Processing, Nanchang Institute of Technology, Nanchang 330099, China*
[2]*School of Information Engineering, Nanchang Institute of Technology, Nanchang 330099, China*

Correspondence should be addressed to Chengzhi Deng; dengchengzhi@126.com

Received 3 June 2018; Revised 3 August 2018; Accepted 10 August 2018; Published 2 September 2018

Academic Editor: Shih-Chia Huang

It is a challenging issue to deal with kinds of appearance variations in visual tracking. Existing tracking algorithms build appearance models upon target templates. Those models are not robust to significant appearance variations due to factors such as illumination variations, partial occlusions, and scale variation. In this paper, we propose a robust tracking algorithm with a learnt dictionary to represent target candidates. With the learnt dictionary, a target candidate is represented with a linear combination of dictionary atoms. The discriminative information in learning samples is exploited. In the meantime, the learning processing of dictionaries can learn appearance variations. Based on the learnt dictionary, we can get a more stable representation for target candidates. Additionally, the observation likelihood is evaluated based on both the reconstruct error and dictionary coefficients with ℓ_1 constraint. Comprehensive experiments demonstrate the superiority of the proposed tracking algorithm to some state-of-the-art tracking algorithms.

1. Introduction

Visual tracking is a fundamental task in computer vision, which is applied in a wide range of applications, such as intelligent transportation, video surveillance, human-computer interaction, and video editing. The goal of visual tracking is to estimate target states of a tracked target in each frame. Although many tracking algorithms are proposed in recent decades [1], designing a robust tracking algorithm remains a challenging issue due to factors such as fast motion, out-of-rotation, nonrigid deformation, and background clutters.

Based on the types of target observations, visual tracking algorithms can be classified as either generative [2–6] or discriminative [7–14]. Generative tracking algorithms search for an image patch that has the most similarity to the tracked target model as the tracking result in the current frame. For a generative algorithm, the prime problem is to build an effective appearance model that is robust to complicated appearance variations.

The discriminative tracking algorithms consider visual tracking as a binary classification problem. The tracked target is distinguished from the surround background by learnt classifiers. The classifiers compute the confidence value for target candidates and distinguish each as a foreground target or a background block. In this work, we will propose a generative algorithm. Next, we briefly review some related works to our tracking algorithm and some recent tracking algorithms.

Kwon et al. [3] decompose the observation model and motion models into multiple basic observation models and multiple basic motion models, respectively. Each basic observation model covers a target appearance variation. Each basic motion model covers a special motion model. A basic observation model and a basic motion model are combined into a basic tracker. The tracking algorithm is robust to drastic appearance changes. He et al. [4] propose an appearance model based on locality sensitive histograms at each pixel location. The proposed observation model is robust to drastic illumination variations. In [2], a target candidate is represented by a set of intensity histograms of multiple image patches, which has a vote value on the corresponding position. A target candidate is represented

by fixed target templates, which is not robust to drastic appearance variations. Wang et al. [6] represent a target candidate based on target templates with affine constraint. The observation likelihood is computed based on a learnt distance metric.

The representation technique with sparse constraint is applied into visual tracking [15–19]. The target representations with sparsity are robust to outliers and occlusions. In [15], Mei et al. use a set of target templates to represent target candidates and represent partial occlusions with trivial templates. The algorithm in [15] is robust to partial occlusion. While severe occlusions occur, the algorithm is not effective. Zhong et al. [18] propose a collaborative model with sparsity constraint. In order to improve the tracking performance, the tracking algorithm combines the generative model and discriminative model. Zhang et al. [16] exploit the spatial layout structure of a target candidate and represent target appearance based on local information and spatial structure. In [19], a target candidate is represented by underlying low-rank with sparse constraints, in which the temporal consistency is used.

Recently, correlation filter [21–24] and deep network [25–28] techniques are applied into visual tracking. In [23], the tracking algorithm takes different features to learn correlation filter. The proposed appearance model is robust to large-scale variations and maintain multiple modes in a particle filter tracking framework. Liu et al. [22] exploit the part-based structure information for correlation filter learning. The learnt filters can accurately distinguish foreground parts from the background. In [28], Ma et al. exploit object features from deep convolutional neural networks. The output of the convolutional layers includes semantic information and hierarchies, which is robust to appearance variations. Huang et al. [27] propose deep feature cascades based tracking algorithm, which considers the visual tracking as a decision-making process.

Motivated by the above-mentioned work, we propose a learnt dictionary based appearance model. A target candidate is represented by a linear combination of the learnt dictionary atoms. The dictionary learning process can learn appearance variations. The dictionary atoms cover recent appearance variations and a stable target representation is obtained. The observation likelihood is evaluated based on the reconstruction error with sparse constrain on dictionary coefficients. Extensive experimental results on some challenging video sequences show the robustness and effectiveness of the proposed appearance model and the tracking algorithm.

The remainder of this paper is organized as follows. Section 2 proposes the novel tracking algorithm, which includes the appearance model, the dictionary learning, the observation likelihood evaluation, and the dictionary update. Section 3 compares the tracking performance of the proposed tracking algorithm with some state-of-the-art algorithms. Section 4 concludes this work.

2. Proposed Tracking Algorithm

In this section, we detail the proposed tracking algorithm including an appearance model based on a learnt dictionary, discrimination dictionary learning for target representation, and a novel likelihood function. In this work, we propose the tracking algorithm in a particle filter tracking framework [29]. The particle filter framework is widely used in visual tracking due to its effectiveness and simplification.

In our tracking algorithm, the target state in the first frame is given as $\mathbf{s}_1$. $\mathbf{y}_1$ denotes the corresponding target observation of $\mathbf{s}_1$. In the first frame, a set of particles (i.e., target candidates) are extracted and denoted as $\mathbf{X}_1 = \{\mathbf{x}_1^1, \mathbf{x}_1^2, \ldots, \mathbf{x}_1^m\}$. These particles are collected by cropping out an image regions surrounding the location of $\mathbf{s}_1$. These particles have same sizes as $\mathbf{s}_1$ and they have same important weights as $w_1^i = 1/m$, $i = 1, 2, \ldots, m$. The particles in frame t are denoted as $\mathbf{X}_t = \{\mathbf{x}_t^1, \mathbf{x}_t^2, \ldots, \mathbf{x}_t^m\}$. The states and the corresponding observation of particle $\mathbf{x}_t^i$ are denoted as $\mathbf{s}_t^i$ and $\mathbf{y}_t^i$ with important weights w_t^i, respectively.

The particles $\mathbf{X}_t = \{\mathbf{x}_t^1, \mathbf{x}_t^2, \ldots, \mathbf{x}_t^m\}$ in frame t are propagated from frame $t-1$ according the state transition model $p(\mathbf{x}_t^i \mid \mathbf{x}_{t-1}^i)$. $p(\mathbf{x}_t^i \mid \mathbf{x}_{t-1}^i)$ is assumed to be a Gaussian distribution:

$$p\left(\mathbf{x}_t^i \mid \mathbf{x}_{t-1}^i\right) \sim \mathbf{G}\left(\mathbf{x}_t^i \mid \mathbf{x}_{t-1}^i, \Sigma\right), \tag{1}$$

where the covariance Σ is a diagonal matrix, in which the diagonal entries denote the variances of the 2D position and the scale of a target candidate.

The target state and the corresponding observation in the t-th frame are denoted as $\mathbf{s}_t$ and $\mathbf{y}_t$, respectively. In the particle filter framework, the $\mathbf{s}_t$ in the frame t is approximately estimated by $\mathbf{x}_t^i$ as

$$\mathbf{s}_t = \sum_{i=1}^{m} w_t^i \mathbf{x}_t^i, \tag{2}$$

where w_t^i is the weight of the particle $\mathbf{x}_t^i$. In the tracking, the particle weights are dynamically updated according to the likelihood of the particle $\mathbf{x}_t^i$ as

$$w_t^i = w_{t-1}^i p\left(\mathbf{y}_t^i \mid \mathbf{x}_t^i\right), \tag{3}$$

where $p(\mathbf{y}_t^i \mid \mathbf{x}_t^i)$ is the likelihood function of particle $\mathbf{x}_t^i$, which is introduced in (15).

2.1. Target Representations. In existing algorithm, a target candidate is represented by a linear combination of a set of target templates. These templates are usually generated from tracking results in previous frames. There are some noises and uncertain information in these templates due to complicated appearance variations. These tracking algorithms are not robust to drastic variations. Thus, in our tracking algorithm, a target candidate is approximately represented by the atoms of a learnt dictionary.

Based on a learnt dictionary $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_n]$, a target candidate $\mathbf{y}$ is approximately represented as

$$\mathbf{y} = \mathbf{d}_1\alpha_1 + \mathbf{d}_2\alpha_2 + \cdots + \mathbf{d}_n\alpha_n, \tag{4}$$

where $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_n]$ is a learnt dictionary. $\mathbf{d}_i$ is an atom of the learnt dictionary $\mathbf{D}$. n is the number of the atoms. $\boldsymbol{\alpha} =$

$[\alpha_1, \alpha_2, \ldots, \alpha_n]^{\mathrm{T}} \in \mathbb{R}^n$ is the coefficient of the dictionary $\mathbf{D}$. The dictionary coefficient $\boldsymbol{\alpha}$ is evaluated by solving

$$\widehat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}} \left\| \mathbf{y} - \mathbf{D}\boldsymbol{\alpha} \right\|_2^2, \tag{5}$$

where $\boldsymbol{\alpha}$ are the coefficient vector for the target candidate $\mathbf{y}$ associated with a learnt dictionary $\mathbf{D}$.

2.2. Dictionary Learning. In existing tracking algorithms, target candidates are represented by target templates, which are some tracking results from previous frames. To improve the tracking performance, we use a learnt dictionary to approximately represent target candidates.

Denote by $\mathbf{T} = [\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_n]$ the set of training samples. Denote by $\mathbf{V}$ the coding vector matrix of training samples $\mathbf{T}$ over $\mathbf{D}$, i.e., $\mathbf{T} = \mathbf{DV}$. The learnt dictionary should have discriminative capability and can adapt to learn appearance variations like partial occlusions, nonrigid deformation, illumination variations, and so on. Based on the learnt dictionary, a stable target representation model is obtained. Motivated by a dictionary learning method [30], we use a discriminative dictionary learning model as

$$J_{D,V} = \arg\min_{\mathbf{D},\mathbf{V}} \left\{ \Phi(\mathbf{T}, \mathbf{D}, \mathbf{V}) + \gamma_1 \|\mathbf{V}\|_1 + \gamma_2 f(\mathbf{V}) \right\}, \tag{6}$$

where $\Phi(\mathbf{T}, \mathbf{D}, \mathbf{V})$ is the discriminative fidelity term; $\|\mathbf{V}\|$ is the sparsity constraint on the coefficient matrix $\mathbf{V}$; $f(\mathbf{V})$ is a discriminate constraint on the coding coefficient matrix $\mathbf{V}$; γ_1 and γ_2 are parameters for balancing this constraint terms. In [30], the dictionary $\mathbf{D}$ includes a set of subdictionaries for all classes. Different from the dictionary learning in [30], in our tracking algorithm, $\mathbf{D}$ is a one-class dictionary and it is learnt from only a set of positive training samples.

In the dictionary learning process, based on the reconstruction error between the training samples $\mathbf{T}$ and the dictionary $\mathbf{D}$, the discriminative fidelity term is defined as

$$\Phi(\mathbf{T}, \mathbf{D}, \mathbf{V}) = \|\mathbf{T} - \mathbf{DV}\|_F^2. \tag{7}$$

To improve the discriminative performance of the learnt dictionary, we add the Fisher discriminative criterion to minimize the within-class scatter of the coefficient matrix $\mathbf{V}$. Denote by $S(V)$ the with-class scatter, which is defined as

$$S(V) = \Sigma_{v_i \in \mathbf{V}} (v_i - m)(v_i - m)^T, \tag{8}$$

where v_i is a vector of the coefficient matrix $\mathbf{V}$, m is the mean vector of the coefficient vector $\mathbf{V}$. In the learning process, we use the trace of $S(V)$ as constraint term $f(\mathbf{V})$ in (6). To prevent some coefficients that are too large in constraint term, a regularized term $\|\mathbf{V}\|_F^2$ is added to $f(\mathbf{V})$

$$f(\mathbf{V}) = \mathrm{tr}(S(\mathbf{V})) + \mu \|\mathbf{V}\|_F^2, \tag{9}$$

where μ is a balancing parameter.

Based on (7), (8), and (9), the dictionary learning model can be rewritten as

$$J_{(D,V)} = \arg\min_{(\mathbf{D},\mathbf{V})} \left\{ \|\mathbf{T} - \mathbf{DV}\|_F^2 + \gamma_1 \|\mathbf{V}\|_1 + \gamma_2 \mathrm{tr}(S(\mathbf{V})) + \gamma_3 \|\mathbf{V}\|_F^2 \right\}, \tag{10}$$

where γ_1, γ_2, and γ_3 are positive scalar parameters.

In (10), we update the dictionary $\mathbf{D}$ and the corresponding coefficient $\mathbf{V}$, iteratively. In the updating processing, one is updated when the other is fixed. When the dictionary $\mathbf{D}$ is fixed, the optimization function is reduced to the following:

$$J_{(V)} = \arg\min_{(\mathbf{D},\mathbf{V})} \left\{ \|\mathbf{T} - \mathbf{DV}\|_F^2 + \gamma_1 \|\mathbf{V}\|_1 + \gamma_4 \|\mathbf{V}\|_F^2 \right\}, \tag{11}$$

and

$$f(\mathbf{V}) = (v_i - m)^T (v_i - m) + \gamma_3 \|\mathbf{V}\|_F^2, \tag{12}$$

where v_i is a vector of the coefficient matrix $\mathbf{V}$ and m is the mean vector of the coefficient vector $\mathbf{V}$.

When $\mathbf{V}$ is fixed, the dictionary $\mathbf{D}$ in (10) is updated as

$$J_{(D)} = \arg\min_{(\mathbf{D})} \|\mathbf{T} - \mathbf{DV}\|_F^2. \tag{13}$$

In the learning process, the training samples are primely important, which should reflect the recent variations of the tracked target and keep diversity to adapt to target appearance variations. In the first frame, a set of training samples are collected. Firstly, the initialized target is selected as training samples. In the meantime, the other training samples are selected by perturbing a few pixels surrounding the center location of the tracked target.

In order to keep the diversity of the learnt dictionary to appearance variations, we set the size of the training samples to 25. In the subsequent frames, we should update the training samples and relearn a dictionary to adapt to target appearance variations. At the current frame, when the tracked target state is computed and located, we crop the corresponding image and extracted the feature vector as a new training sample. Then, the new training sample is added to the set of current training samples and the oldest training sample is swapped.

2.3. Likelihood Evaluation. The similarity metric of a target candidate and the corresponding candidate is an important issue in visual tracking. In this work, the similarity is measured as

$$d(\mathbf{y}, \mathbf{D}\widehat{\boldsymbol{\alpha}}) = (\mathbf{y} - \mathbf{D}\widehat{\boldsymbol{\alpha}})^T (\mathbf{y} - \mathbf{D}\widehat{\boldsymbol{\alpha}}), \tag{14}$$

where $\mathbf{D}$ is the learnt dictionary and $\widehat{\boldsymbol{\alpha}}$ is the coefficient vector of the learnt dictionary computed in (6).

Based on the distance between a target candidate and the corresponding template dictionary, the target observation likelihood is computed as

$$p(\mathbf{y} \mid \mathbf{x}) \propto \exp\left\{ -\psi \left(d(\mathbf{y}, \mathbf{D}\widehat{\boldsymbol{\alpha}}) \right) - \zeta \|\widehat{\boldsymbol{\alpha}}\|_1 \right\}, \tag{15}$$

where $d(\mathbf{y}, \mathbf{D}\widehat{\boldsymbol{\alpha}})$ is the distance between a target candidate $\mathbf{y}$ and the corresponding dictionary $\mathbf{D}$, ψ is the standard deviation of the Gaussian, and ζ is a positive parameter.

2.4. Visual Tracking with Dictionary Based Representation. By integrating the proposed target representation and the online dictionary learning and updating and the observation evaluation, the proposed visual tracking algorithm is outlined in Algorithm 1. The particle filter framework is used for all

(1) In the first frame, manually select the tracked target state $\mathbf{s}_1$; collect n training samples $\mathbf{T} = [\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_n]$; learn a dictionary $\mathbf{D}_1 = [\mathbf{d}_1, \ldots, \mathbf{d}_n]$ according to the learning scheme in Section 2.2; sample m particles $\{\mathbf{x}_1^i\}_{i=1}^m$ with equal weights as $1/m$.
Input: t-th video frame.
(2) Resample m particles $\{\mathbf{x}_t^i\}_{i=1}^m$ according to motion model $p(\mathbf{x}_t^i \mid \mathbf{x}_{t-1}^i)$.
(3) Extract feature vectors $\{\mathbf{y}_t^i\}_{i=1}^m$ according to $\{\mathbf{x}_t^i\}_{i=1}^m$.
(4) **for** $i = 1$ to m **do**
(5) Compute observation likelihoods $p(\mathbf{y}_t^i \mid \mathbf{x}_t^i)$ via Eqn. (15).
(6) Update particle weight w_t^i via Eqn. (3).
(7) **end**
(8) Compute target state $\widehat{\mathbf{s}}_t$ with Eqn. (2).
(9) Extract feature vector $\mathbf{y}_t$ according to $\widehat{\mathbf{s}}_t$.
(10) Update the training samples with $\mathbf{y}_t$.
(11) Learn dictionary $\mathbf{D}_t$ according to Eqns. (11) and (13) in Section 2.2.
(12) Obtain dictionary $\mathbf{D}_t$.
(13) Return $\widehat{\mathbf{s}}_t$.

Algorithm 1: Proposed tracking algorithm.

Table 1: Average frames per second (FPS).

Sequence	Coupon	Fish	Football	Football1	Man	Singer2	Sylv	Walking
Frames	327	476	362	74	134	366	1345	412
Total times	269	430	196	39	67	437	724	253
FPS	1.21	1.11	1.84	1.90	1.99	0.84	1.86	1.63

video sequences. For a video sequence, the tracked target is manually selected by a bounding box in the first frame. A set of particles (i.e., target candidates) are selected with same weights in the particle framework. The training samples are collected and a dictionary is learnt in the first frame. In the subsequent tracking processing, when the current target states are evaluated, the current tracking result is added to the training samples. The dictionary is relearnt according to the updated training samples.

3. Experiments

We conduct comprehensive experiments on some challenging video sequences and compare the proposed tracking algorithm against some state-of-the-art tracking algorithms. These tracking algorithms include Struck [12], SCM [18], VTD [3], Frag [2], L1 [15], LSHT [4], LRT [19], and TGPR [20]. For fairness, we use the source codes or binary codes provided by the authors and initialize all the evaluated algorithms with default parameters in all experiments. 8 challenging video sequences from a recent benchmark [1] are used to evaluate the tracking performance. Table 2 shows the main challenging attributes in these test sequences.

The proposed tracking algorithm is implemented in MATLAB. All experimental results are conducted on a PC with Intel(R) Core(TM) i5-2400 3.10GHZ and 4 GB memory. The number of particles is set to 300. The target features are described by the histograms of sparse coding (HSC) [31]. The value of ψ in (15) is set to 20. In the proposed tracking algorithm, the number of atoms of a learnt dictionary is set to 25.

The proposed tracking average processing time of the proposed tracking algorithm is 1.55 frames per second (FPS). We show the average tracking speed for each sequence in Table 1. Compared with some state-of-the-art tracking algorithms [1], the proposed tracking algorithm is superior to SCM. However, it is slow to some tracking algorithms, e.g., Struck, VTD, and LSHT. This is due to online dictionary learning spending some time in optimizing the target representation. We can learn the dictionary every five frames. But this may influence the dictionary adaption to complicated tracking surrounding. In our tracking algorithm, the dictionary is learnt in each frame.

3.1. Quantitative Evaluation. We use four evaluation measures in the experiments including average center location error, success rate, overlap rate, and precision. These measures are adopted in recent tracking benchmark [1].

We show the precision plots for these tracking algorithms in Figure 1 for 9 tracking algorithms. The average center location errors (CLE) are shown in Table 3. From Figure 1 and Table 3, we can see that the proposed tracking algorithm obtains the best two results in six of eight sequences. The proposed tracking algorithm achieves the smallest CLE over all the 8 sequences. TGPR achieves robust tracking results in the *Football*1, *Sylv*, and *Singer*2 video sequences. LRT obtains the best tracking results in the *Coupon*, *Walking*, and *Football* video sequences. Struck tracks the *Fish* and *Man* video sequences well and achieves the best tracking results in CLE.

Table 4 presents success rates for 9 tracking algorithms on the 8 sequences. Figure 2 also shows the success rate

Table 2: The main attributes of the 8 video sequences. Target size: the initial target size in the first frame; OPR: out-of-plane rotation; IPR: in-plane rotation; BC: background clutter; IV: illumination variation; Occ: occlusion; Def: deformation; SV: scale variation.

Sequence	Frames	Image size	Target size	OPR	IPR	BC	IV	Occ	Def	SV
Coupon	327	320×240	62×98			√		√		
Fish	476	320×240	60×88				√			
Football	362	624×352	39×50	√	√	√		√		
Football1	74	352×288	26×43	√	√	√				
Man	134	241×193	26×40				√			
Singer2	366	624×352	67×122	√	√	√	√		√	
Sylv	1345	320×240	51×61	√	√		√			
Walking	412	768×576	24×79					√	√	√

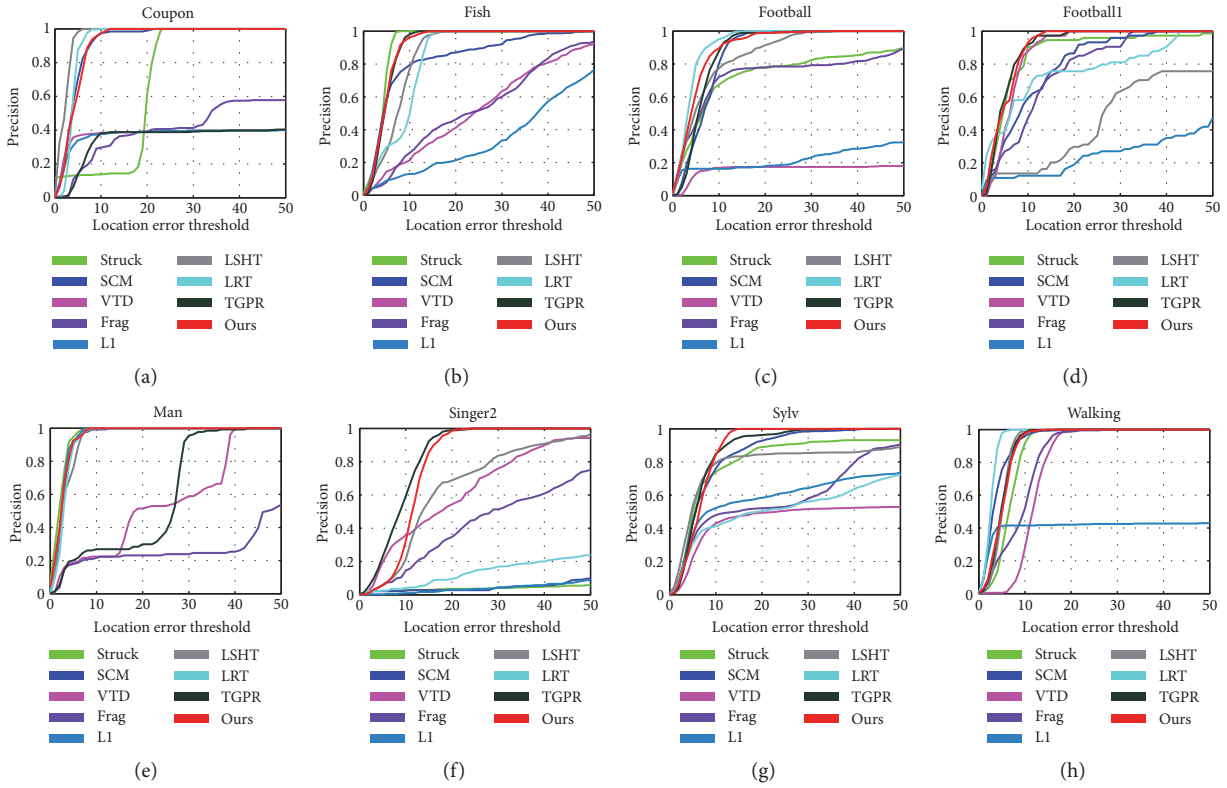


Figure 1: Precision plots in terms of location error threshold (in pixels).

plots for the evaluated tracking algorithms. From Table 4 and Figure 2, it can be seen that the proposed tracking tracks well in most of these video sequences. The proposed tracking algorithm obtains the best tracking results in most of these video sequences. Additionally, TGPR achieves accurate tracking results in the *Fish* and *Singer*2 video sequences. LRT achieves the best tracking results in the *Coupon*, *Walking*, and *Football* video sequences. LSHT obtains the best tracking results in the *Fish*, *Man*, and *Coupon* video sequences.

Table 5 presents the average overlap rates for the 9 tracking algorithms. It can be seen that the proposed tracking algorithm achieves the best or the second best tracking results in most video sequences. Struck, LSHT, LRT, and TGPR also achieve robust tracking results in some video sequences.

3.2. Qualitative Evaluation. Next, we will analyze the tracking performance of these tracking algorithms on the 8 video sequences.

In the *Coupon* sequence shown in Figure 3(a), the tracked target is a coupon book with cluttered background. When the target is occluded by himself, VTD and L1 drift away from the target. Frag, TGPR, VTD, and L1 lose the target and track the other similar object until the end of whole sequence. Struck, SCM, LSHT, LRT, TGPR, and the proposed tracking algorithm can accurately track the target throughout the video sequence.

Figure 3(b) presents some tracking results for the 9 tracking algorithms on the *Fish* sequence. Struck, LSHT, and TGPR achieve accurate tracking results. The proposed

Table 3: Average center location errors (in pixels). The best two results are shown in italic and bold colors, respectively.

Sequence	Struck [12]	SCM [18]	VTD [3]	Frag [2]	L1 [15]	LSHT [4]	LRT [19]	TGPR [20]	Ours
Coupon	15.0	6.0	65.2	56.2	66.3	**4.3**	*3.4*	65.7	**4.3**
Fish	*3.9*	8.3	24.7	24.7	36.4	7.3	8.5	4.8	**4.7**
Football	15.3	6.9	218.3	**4.6**	68.4	7.2	*3.8*	6.2	4.9
Football1	7.0	10.4	6.4	11.9	59.3	30.8	12.1	**5.0**	*4.9*
Man	*2.3*	2.9	22.8	44.6	**2.6**	3.1	3.0	20.8	**2.6**
Singer2	174.7	172.2	20.2	35.9	145.8	18.4	126.6	*8.8*	**11.4**
Sylv	11.7	7.9	58.4	22.7	31.0	13.6	28.9	**6.8**	*6.5*
Walking	6.5	**3.9**	11.8	8.9	125.3	5.0	*2.6*	5.0	5.0
Average	29.5	27.3	53.5	26.2	66.9	*11.2*	23.6	15.4	*5.5*

Table 4: Success rates (%). The best two results are shown in italic and bold colors, respectively.

Sequence	Struck [12]	SCM [18]	VTD [3]	Frag [2]	L1 [15]	LSHT [4]	LRT [19]	TGPR [20]	Ours
Coupon	*100*	*100*	39.4	40.9	39.4	*100*	*100*	39.4	*100*
Fish	*100*	86.6	39.7	47.3	20.2	*100*	*100*	*100*	*100*
Football	69.3	88.7	16.9	72.9	16.3	79.6	*96.7*	**94.8**	92.5
Football1	**89.2**	39.2	81.1	43.2	12.2	14.9	58.1	85.1	*90.5*
Man	99.3	98.5	22.4	21.0	98.5	*100*	99.3	26.1	*100*
Singer2	3.6	3.0	33.3	45.9	4.1	67.5	9.8	*100*	*100*
Sylv	80.3	86.6	48.4	50.3	55.5	83.3	48.1	**94.3**	*100*
Walking	54.9	**79.1**	16.3	50.2	41.5	54.6	*97.3*	55.1	*77.7*
Average	74.6	72.7	37.2	46.5	36.0	75.0	**76.2**	74.3	*95.1*

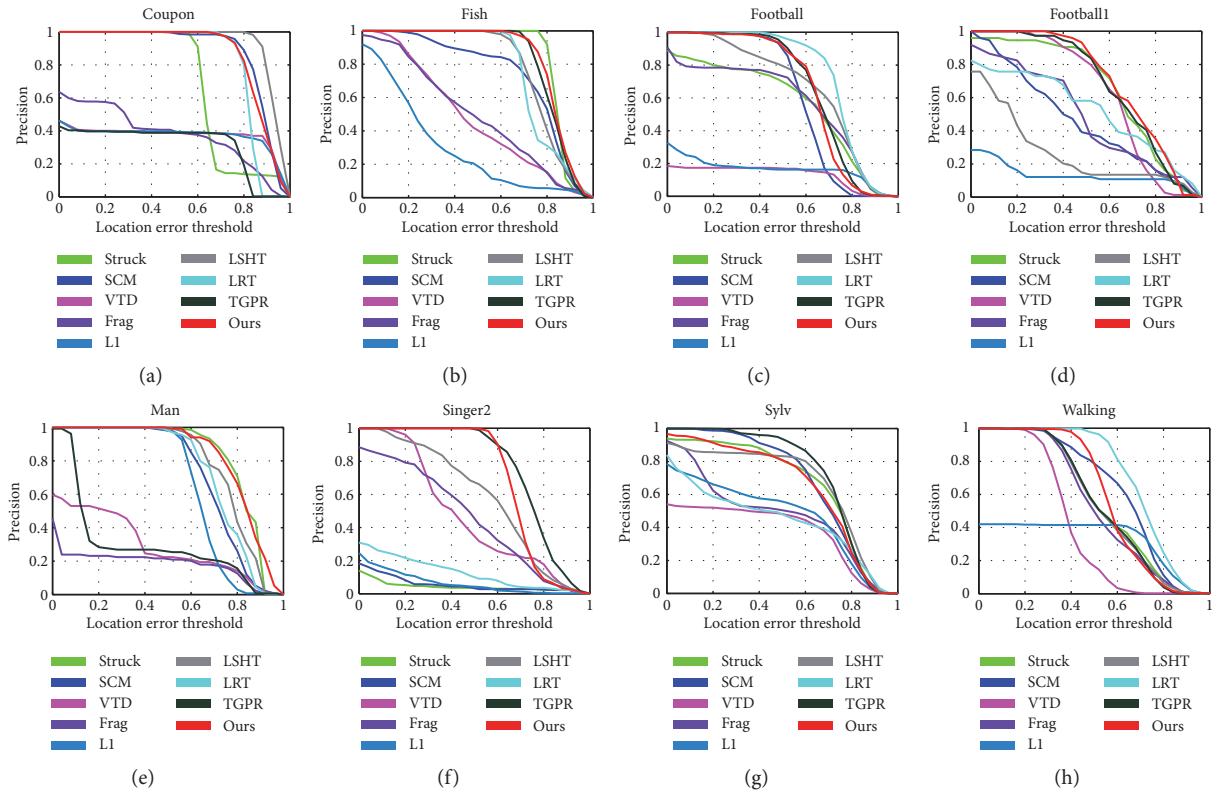
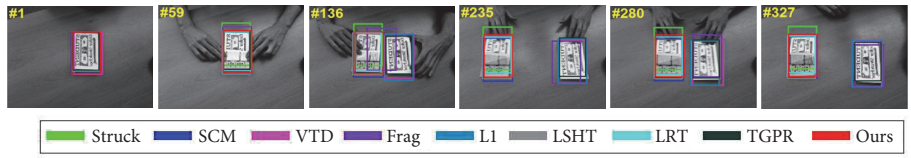
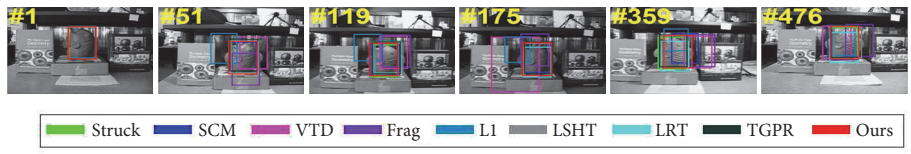


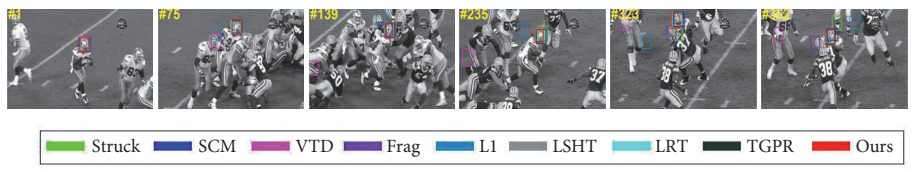
Figure 2: Success plots in terms of overlap threshold.



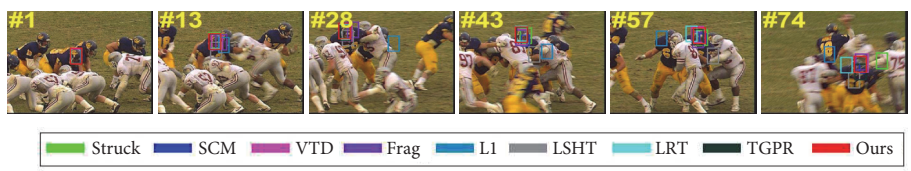
(a) *Coupon*



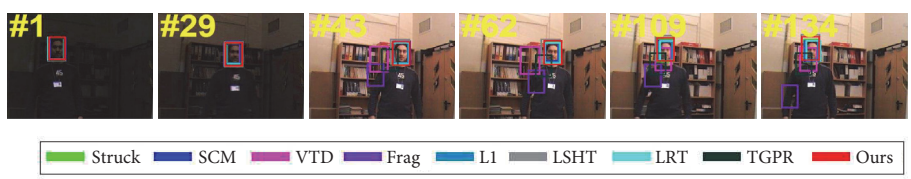
(b) *Fish*



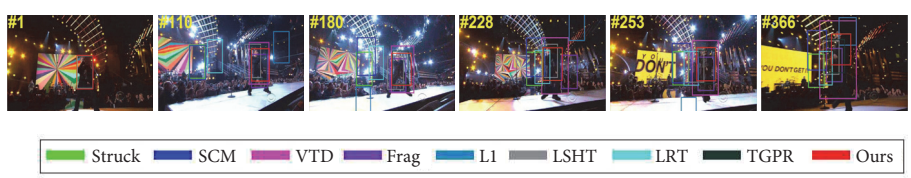
(c) *Football*



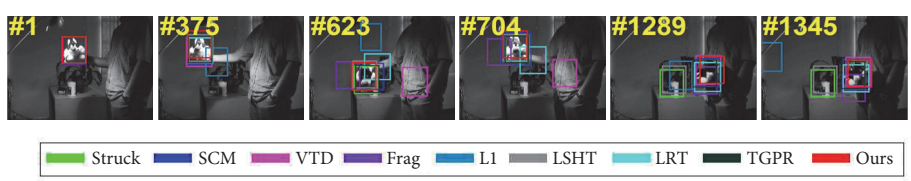
(d) *Football1*



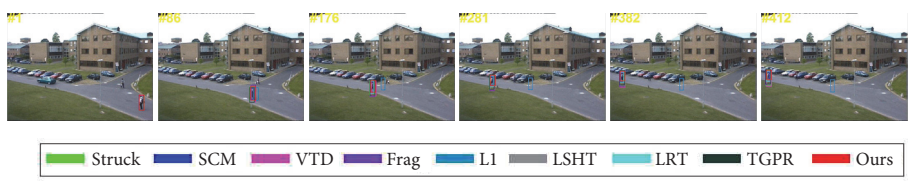
(e) *Man*



(f) *Singer2*



(g) *Sylv*



(h) *Walking*

Figure 3: The tracking results on the 8 sequences.

Table 5: Average overlap rates (%). The best two results are shown in italic and bold colors, respectively.

Sequence	Struck [12]	SCM [18]	VTD [3]	Frag [2]	L1 [15]	LSHT [4]	LRT [19]	TGPR [20]	Ours
Coupon	70.2	82.3	36.2	37.1	35.2	*89.2*	80.7	32.8	**84.1**
Fish	*84.3*	74.0	47.3	48.9	28.6	78.1	76.5	82.4	**83.7**
Football	55.7	60.3	13.0	56.2	16.2	66.0	*75.0*	**66.9**	65.5
Football1	66.0	45.4	63.1	48.4	13.1	26.0	52.2	**67.9**	*69.6*
Man	**81.9**	71.9	28.8	17.5	65.3	77.8	74.1	29.4	*82.9*
Singer2	4.2	5.3	46.9	44.9	6.0	58.1	12.3	*75.4*	**69.5**
Sylv	66.0	67.8	37.4	46.4	46.4	65.8	43.4	*72.5*	**71.0**
Walking	55.9	**63.4**	39.2	53.4	32.8	55.0	*71.6*	55.1	58.9
Average	60.5	58.8	39.0	44.1	30.4	**64.5**	60.7	60.3	*73.2*

tracking algorithm can learn the appearance variations in the dictionary learning processing. It can accurately track the target throughout the video sequence. Struck, LSHT, LRT, and TGPR also track the target until the end of the video sequence.

In the *Football* sequence shown in Figure 3(c), a football match is going on. The tracked target is a player, which is similar to others in color and shape. The target undergoes partial occlusion, background clutter, and in-plane and out-of-plane rotations. Due to the influence of background clutter, VTD and L1 lose the target. When some similar objects appear surrounding the target, Struck, SCM, Frag, and LSHT track the other similar distracters and lose the target. Compared with these algorithms, LRT, TGPR, and the proposed tracking algorithm can track the target successfully.

As shown in Figure 3(d), the tracked target is influenced by out-of-plane and in-plane rotations. And there are some other objects that are similar to the tracked target. Frag loses the target when other distracters appear surrounding the target, which is very similar to the tracked target. L1 and Frag achieve inaccurate tracking results when the target rotates in-plane and out-of-plane. Struck, TGPR and the proposed tracking algorithm can track the target throughout the sequence. In the three algorithms, the proposed algorithm achieves the most accurate tracking results in average center location error, success, and overlap rates.

Figure 3(e) shows some tracking results in the *Man* sequence. The tracked target is a moving face in an indoor room. Influenced by drastic illumination variations, VTD, Frag, and TGPR lose the target after the 40th frame until the end of the sequence. Struck, SCM, LSHT, LRT, and the proposed tracking algorithm can successfully track the whole sequence. The proposed tracking algorithm obtains the most robust tracking result.

The *Singer*2 video sequence shown in Figure 3(f) is captured on an indoor stage with drastic illumination variations. The target is also affected by nonrigid deformation, background clutters, and in-plane and out-of-plane rotations. Struck, SCM, L1, and LRT only track the target before the first 47th frame due to the appearance variations. VTD obtains inaccurate scale evaluation when the target undergoes nonrigid deformation. The proposed tracking algorithm can successfully track the target in the whole sequence. The learnt dictionary covers the target variations, so the linear combination of the atoms in the dictionary can represent these variational appearance.

In the *Sylv* video sequence shown in Figure 3(g), the target is a moved toy, which undergoes illumination variations and in-plane and out-of-plane rotations. L1 loses the target due to the influence of illumination variations and rotation from up to down. It locates the target again after the 495th frame when the tracked target rotates from down to up. When the target is affected by illumination variation, LRT drift away from the tracked target until the end of the video sequence. Frag uses fixed target templates, so it cannot adapt to the appearance variations. It achieves inaccurate tracking results. Compared with these algorithms, the proposed tracking algorithm obtains more tracking results. This is attributed to the fact that the proposed target representation can learn the appearance variations.

As shown in Figure 3(h), the tracked target is a walking man in an outdoor scene. Due to the influence of partial occlusion, nonrigid deformation, and scale variation, L1 loses the target until the end of this sequence. VTD, TGPR, and LSHT can not accurately evaluate the scale variation. SCM, Struck, LRT, and the proposed tracking algorithm obtain more accurate tracking results.

From the above analysis, we can see that the proposed appearance model is effective and efficient. The proposed tracking algorithm is robust to significant appearance variations, e.g., drastic illumination variations, partial occlusion, and out-of-plane rotation.

4. Conclusion

We have presented an effective tracking algorithm based on learnt discrimination dictionary. Different from exist tracking algorithms, a target candidate is represented with a linear combination of dictionary atoms. The dictionaries are leant and updated in the tracking processing, which can learn the target appearance variation and exploit the discriminative information in the learning samples. The learning samples are collected from previous tracking results. The proposed tracking algorithm is robust to drastic illumination variations, nonrigid deformation, and rotation. Conducted experiments on some challenging video sequences demonstrate the robustness in comparison with state-of-the-art tracking algorithms.

Data Availability

(1) The video sequences data used to support the findings of this study have been deposited in http://cvlab.hanyang.ac.kr/tracker_benchmark/datasets.html or http://www.visual-tracking.net. (2) The measure metrics and the challenging attributes in videos used to support the finding of this study are included within the articles: Y. Wu, J. Lim, and M. Yang, Online Object Tracking: A Benchmark, IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 2411-2418.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Jiangxi Science and Technology Research Project of Education Department of China (Nos: GJJ151135 and GJJ170992), the National Natural Science Foundation of China (No: 61661033), the Jiangxi Natural Science Foundation of China (Nos: 20161BAB202040 and 20161BAB202041), and the Open Research Fund of Jiangxi Province Key Laboratory of Water Information Cooperative Sensing and Intelligent Processing (No: 2016WICSIP020).

References

[1] Y. Wu, J. Lim, and M.-H. Yang, "Object tracking benchmark," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1834–1848, 2015.

[2] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '06)*, pp. 798–805, June 2006.

[3] J. Kwon and K. M. Lee, "Visual tracking decomposition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '10)*, pp. 1269–1276, June 2010.

[4] S. He, Q. Yang, R. W. H. Lau, J. Wang, and M.-H. Yang, "Visual tracking via locality sensitive histograms," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '13)*, pp. 2427–2434, June 2013.

[5] J. Wang, H. Z. Wang, and Y. Yan, "Robust visual tracking by metric learning with weighted histogram representations," *Neurocomputing*, vol. 153, pp. 77–88, 2015.

[6] J. Wang, Y. Wang, and H. Wang, "Adaptive Appearance Modeling with Point-to-Set Metric Learning for Visual Tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 9, pp. 1987–2000, 2017.

[7] H. Grabner, M. Grabner, and H. Bischof, "Real-time tracking via on-line boosting," in *Proceedings of the British Machine Vision Conference (BMVC '06)*, pp. 47–56, September 2006.

[8] B. Babenko, M. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1619–1632, 2010.

[9] K. H. Zhang, L. Zhang, and M. H. Yang, "Fast compressive tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 10, pp. 2002–2015, 2014.

[10] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1409–1422, 2012.

[11] J. Zhang, S. Ma, and S. Sclaroff, "MEEM: Robust Tracking via Multiple Experts Using Entropy Minimization," in *Computer Vision – ECCV 2014*, vol. 8694 of *Lecture Notes in Computer Science*, pp. 188–203, Springer International Publishing, Cham, 2014.

[12] S. Hare, A. Saffari, and P. H. S. Torr, "Struck: structured output tracking with kernels," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '11)*, pp. 263–270, IEEE, Barcelona, Spain, November 2011.

[13] L. Ma, X. Zhang, W. Hu, J. Xing, J. Lu, and J. Zhou, "Local Subspace Collaborative Tracking," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 4301–4309, Santiago, Chile, December 2015.

[14] Y. Sui, Y. Tang, and L. Zhang, "Discriminative low-rank tracking," in *Proceedings of the 15th IEEE International Conference on Computer Vision, ICCV 2015*, pp. 3002–3010, Chile, December 2015.

[15] X. Mei and H. Ling, "Robust visual tracking and vehicle classification via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 11, pp. 2259–2272, 2011.

[16] T. Zhang, S. Liu, C. Xu, S. Yan, and B. Ghanem, "Structure sparse tracking, IEEE Conference on Computer Vision and Pattern Recognition," in *Proceedings of the Structure sparse tracking, IEEE Conference on Computer Vision and Pattern Recognition*, pp. 150–158, 2015.

[17] L. Zhang, H. Lu, D. Du, and L. Liu, "Sparse hashing tracking," *IEEE Transactions on Image Processing*, vol. 25, no. 2, pp. 840–849, 2016.

[18] W. Zhong, H. Lu, and M.-H. Yang, "Robust object tracking via sparse collaborative appearance model," *IEEE Transactions on Image Processing*, vol. 23, no. 5, pp. 2356–2368, 2014.

[19] T. Zhang, S. Liu, N. Ahuja, M.-H. Yang, and B. Ghanem, "Robust visual tracking via consistent low-rank sparse learning," *International Journal of Computer Vision*, pp. 1–20, 2014.

[20] J. Gao, H. Ling, W. Hu, and J. Xing, "Transfer learning based visual tracking with gaussian processes regression," in *Computer Vision—ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., vol. 8691 of *Lecture Notes in Computer Science*, pp. 188–203, 2014.

[21] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *Proceedings of the 15th IEEE International Conference on Computer Vision (ICCV '15)*, pp. 4310–4318, Santiago, Chile, December 2015.

[22] T. Zhang, S. Liu, C. Xu, B. Liu, and M.-H. Yang, "Correlation particle filter for visual tracking," *IEEE Transactions on Image Processing*, vol. 27, no. 6, pp. 2676–2687, 2018.

[23] T. Zhang, C. Xu, and M.-H. Yang, "Multi-task correlation particle filter for robust object tracking," in *Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, pp. 4819–4827, USA, July 2017.

[24] M. Mueller, N. Smith, and B. Ghanem, "Context-aware correlation filter tracking," in *Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, pp. 1387–1395, USA, July 2017.

[25] Z. Teng, J. Xing, Q. Wang, C. Lang, S. Feng, and Y. Jin, "Robust Object Tracking Based on Temporal and Spatial Deep Networks," in *Proceedings of the 16th IEEE International Conference*

on Computer Vision, ICCV 2017, pp. 1153–1162, Italy, October 2017.

[26] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR '16)*, pp. 4293–4302, July 2016.

[27] C. Huang, S. Lucey, and D. Ramanan, "Learning Policies for Adaptive Tracking with Deep Feature Cascades," in *Proceedings of the 16th IEEE International Conference on Computer Vision, ICCV 2017*, pp. 105–114, Italy, October 2017.

[28] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, "Hierarchical convolutional features for visual tracking," in *Proceedings of the 15th IEEE International Conference on Computer Vision, ICCV 2015*, pp. 3074–3082, Chile, December 2015.

[29] M. Isard and A. Blake, "Condensation-conditional density propagation for visual tracking," *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5–28, 1998.

[30] M. Yang, L. Zhang, X. C. Feng, and D. Zhang, "Fisher discrimination dictionary learning for sparse representation," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '11)*, pp. 543–550, Barcelona, Spain, November 2011.

[31] X. Ren and D. Ramanan, "Histograms of sparse codes for object detection," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2013*, pp. 3246–3253, USA, June 2013.

Hindawi
Advances in Multimedia
Volume 2018, Article ID 8320207, 13 pages
https://doi.org/10.1155/2018/8320207

Research Article

Lane Detection Based on Connection of Various Feature Extraction Methods

Mingfa Li, Yuanyuan Li, and Min Jiang

Department of Electronic and Electrical Engineering, Shanghai University of Engineering Science, Shanghai, China

Correspondence should be addressed to Yuanyuan Li; liyuanyuanedu@163.com

Received 31 May 2018; Accepted 31 July 2018; Published 7 August 2018

Academic Editor: Jenq-Neng Hwang

Lane detection is a challenging problem. It has attracted the attention of the computer vision community for several decades. Essentially, lane detection is a multifeature detection problem that has become a real challenge for computer vision and machine learning techniques. Although many machine learning methods are used for lane detection, they are mainly used for classification rather than feature design. But modern machine learning methods can be used to identify the features that are rich in recognition and have achieved success in feature detection tests. However, these methods have not been fully implemented in the efficiency and accuracy of lane detection. In this paper, we propose a new method to solve it. We introduce a new method of preprocessing and ROI selection. The main goal is to use the HSV colour transformation to extract the white features and add preliminary edge feature detection in the preprocessing stage and then select ROI on the basis of the proposed preprocessing. This new preprocessing method is used to detect the lane. By using the standard KITTI road database to evaluate the proposed method, the results obtained are superior to the existing preprocessing and ROI selection techniques.

1. Introduction

With the rapid development of society, automobiles have become one of the transportation tools for people to travel. In the narrow road, there are more and more vehicles of all kinds [1]. As more and more vehicles are driving on the road, the number of victims of car accidents is increasing every year [2]. How to drive safely under the condition of numerous vehicles and narrow roads has become the focus of attention. Advanced driver assistance systems which include lane departure warning (LDW) [3], Lane Keeping Assist, and Adaptive Cruise Control (ACC) [4] can help people analyse the current driving environment and provide appropriate feedback for safe driving or alert the driver in dangerous circumstances. This kind of auxiliary driving system is expected to become more and more perfect [5]. However, the bottleneck of the development of this system is that the road traffic environment is difficult to predict [6]. After investigation, in the complex traffic environment where vehicles are numerous and speed is too fast, the probability of accidents is much greater than usual. In such a complex traffic situation, road colour extraction and texture detection as well as road boundary and lane marking are the main perceptual clues of human driving [7].

Lane detection is a hot topic in the field of machine learning and computer vision and has been applied in intelligent vehicle systems [8]. The lane detection system comes from lane markers in a complex environment and is used to estimate the vehicle's position and trajectory relative to the lane reliably [9]. At the same time, lane detection plays an important role in the lane departure warning system. The lane detection task is mainly divided into two steps: edge detection and line detection.

Qing et al. [10] proposed the extended edge linking algorithm with directional edge gap closing. The new edge could be obtained with the proposed method. Mu and Ma proposed Sobel edge operator which can be applied to adaptive area of interest (ROI) [11]. However, there are still some false edges after edge detection. These errors will affect the subsequent lane detection. Wang et al. proposed a Canny edge detection algorithm for feature extraction [12]. The algorithm provides an accurate fit to lane lines and could be adaptive to complicated road environment. In 2014, Srivastava et al. proposed that the improvements to the Canny

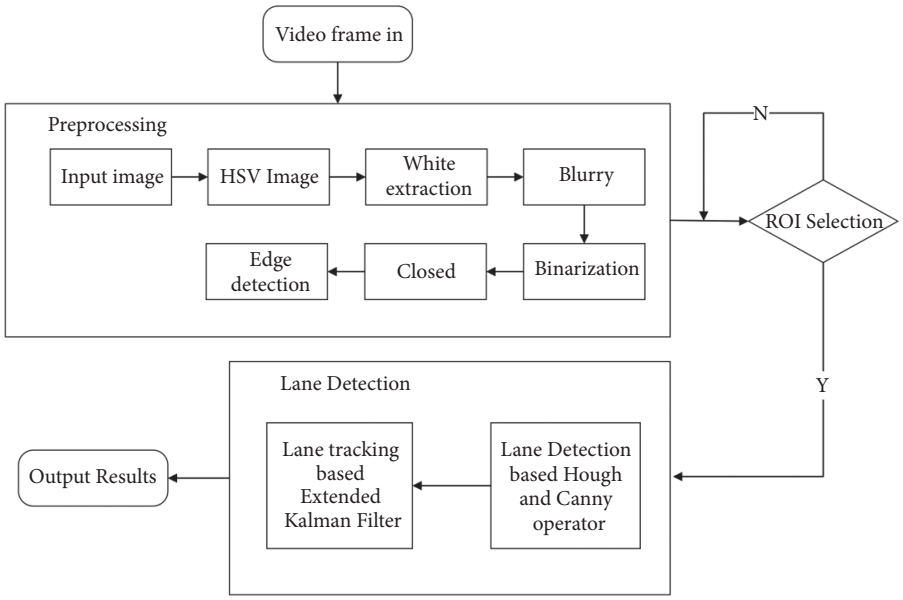


Figure 1: Block diagram of proposed methods.

edge detection can effectively deal with various noises in the road environment [13]. Sobel and Canny edge operator are the most commonly used and effective methods for edge detection.

Line detection is as important as edge detection in lane detection. With regard to line detection, we usually have two methods which include feather-based method and model-based methods. Niu et al. used a modified Hough transform to extract segments of the lane profile and used DBSCAN (density based spatial application noise clustering) clustering algorithm for clustering [14]. In 2016, Mammeri et al. used progressive probabilistic Hough transform combined with maximum stable extreme area (MSER) technology to identify and detect lane lines and utilized Kalman filter to achieve continuous tracking [15]. However, the algorithm does not work well at night.

In this paper, we propose a lane detection method that is suitable for all kinds of complex traffic situations, especially as driving speed in roads is too fast. First, we preprocessed each frame image and then selected the area of interest (ROI) of the processed images. Finally, we only needed edge detection vehicle and line detection for the ROI area. In this study, we introduced a new preprocessing method and ROI selection method. First, in the preprocessing stage, we converted the RGB colour model to the HSV colour space model and extracted white features on the HSV model. At the same time, the preliminary edge feature detection is added in the preprocessing stage, and then the part below the image is selected as the ROI area based on the proposed preprocessing. Compared with the existing methods, the existing preprocessing methods only perform operations such as graying, blurring, X-gradient, Y-gradient, global gradient, thresh, and morphological closure. And the ways to select the ROI area are also very different. Some of them are based on the edge feature of the lane to select the ROI area, and some are based on the colour feature of the lane to select the ROI area. These existing methods do not provide accurate and fast lane information, which increases the difficulty of lane detection. In this paper, experiments show that the proposed method is significantly better than the existing preprocessing method and ROI selection method in lane detection.

2. Overview of the Proposed System

This paper presents an advanced lane detection technology to improve the efficiency and accuracy of real-time lane detection [16]. The lane detection module is usually divided into two steps: (1) image preprocessing and (2) the establishment and matching of line lane detection model.

Figure 1 shows the overall diagram of our proposed system where lane detection blocks are the main contributions of this paper. The first step is to read the frames in the video stream. The second step is to enter the image preprocessing module. What is different from others is that in the preprocessing stage we not only process the image itself but also do colour feature extraction and edge feature extraction [17]. In order to reduce the influence of noise in the process of motion and tracking, after extracting the colour features of the image, we need to use Gaussian filter to smooth the image. Then, the image is obtained by binary threshold processing and morphological closure. These are the preprocessing methods mentioned in this paper.

Next, we select the adaptive area of interest (ROI) in the preprocessed image. The last step is lane detection. Firstly, Canny operator is used to detect the edge of lane line; then Hough transform is used to detect line lane. Finally, we use Extended Kalman Filter (EKF) to detect and track lane line in real time.

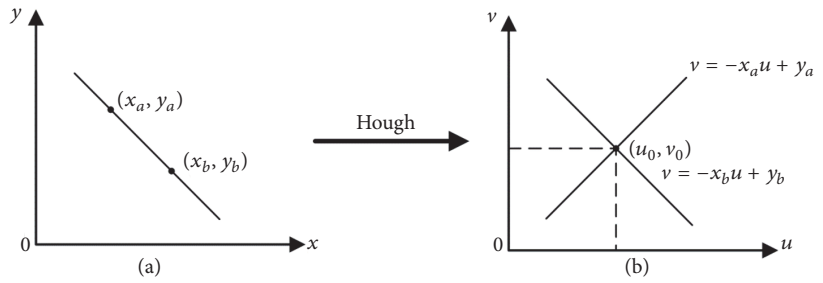


Figure 2: Hough transform. (a) A line in a Cartesian coordinate system and (b) spatial parameters after Hough transformation.

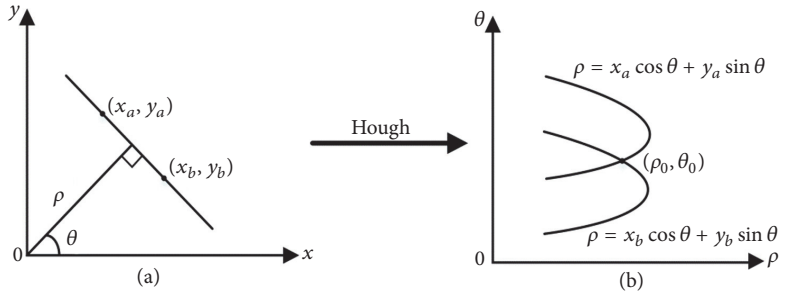


Figure 3: Hough transform. (a) Cartesian coordinate system parameter and (b) polar coordinate system parameter.

3. Proposed Methods

In this paper, based on the previous preprocessing, we firstly extract the colour features based on the white colour and then extract the edge features based on the straight lane. Because the high-speed section is the traffic accident-prone section, the high-speed road section mostly is the straight line lane [18]. Therefore, in order to obtain a very high recognition rate, we successively carry on colour detection and edge detection to the lane. This paper combines colour features extraction and edge features extraction, and the experiment proves that the recognition rate and accuracy of lane detection are greatly improved.

Our main contribution in this paper is to do a lot of work in the preprocessing stage. We proposed to perform colour transform of HSV in the preprocessing stage, then extract white, and then perform conventional preprocessing operations in sequence. Moreover, we selected an improved method proposed in the area of interest (ROI). In this paper, based on the proposed preprocessing method (after HSV colour transform, white feature extraction, and basic preprocessing), one-half part of the processed image is selected as the area of interest (ROI). In addition, we performed twice edge detection. The first is in the preprocessing stage, and the second is in the lane detection stage after the ROI is selected. The purpose of performing twice edge detection is to enhance the lane recognition rate.

In this paper, Hough transform is used for the straight line detection. Figure 2 shows the basic principles of the Hough transform. In Figure 2(a), each point on the straight line crossing the point (x_a, y_a) and the point (x_b, y_b) corresponds to a straight line $v = -x_a u + y_a$ and a straight line $v = -x_b u + y_b$ on the parameter space map in Figure 2(a) after Hough transformation; two lines intersect at the point (u_0, v_0), where u_0 and v_0 are the parameters of the line determined by the point (x_a, y_a) and point (x_b, y_b) in Figure 2(a) [19]. On the contrary, the straight line $v = -x_a u + y_a$ and the straight line $v = -x_b u + y_b$ where the parameter space in Figure 2(b) intersects at the same point and the collinear points in Figure 2(a) are correspondence [20]. According to this characteristic, given some specific points in Figure 2(a), the line equations connecting these points in Figure 2(b) can be calculated by Hough transform.

The Hough transform is implemented in polar form as [21]

$$\rho = x \cos(\theta) + y \sin(\theta), \tag{1}$$

where (x, y) are coordinates of nonzero pixels in binary image.

ρ is the distance between the x-axis and fitted line.

θ is the angle between x-axis and normal line. The value range of θ is $\pm 90^\circ$.

As shown in Figure 3, the Hough transform transforms the points of the image in Figure 3(a) into the polar coordinate parameter space of Figure 3(b). We can see that the collinear point (x_a, y_a) and point (x_b, y_b) in Figure 3(a) intersect at the same point (ρ_0, θ_0) in Figure 3(b). Here, ρ and θ are the polar parameters of the desired straight line [21].

Different from Figure 2(b), when Figure 3(b) is expressed in polar coordinates, the collinear point (x_a, y_a) and point (ρ_0, θ_0) mapped to the parameter space in the original image intersect at the point (ρ_0, θ_0) [21].

The Kalman filtering algorithm is used to track lane lines in real time. In this paper, we use Extended Kalman Filter (EKF) to track the lane in real time [22]. After the

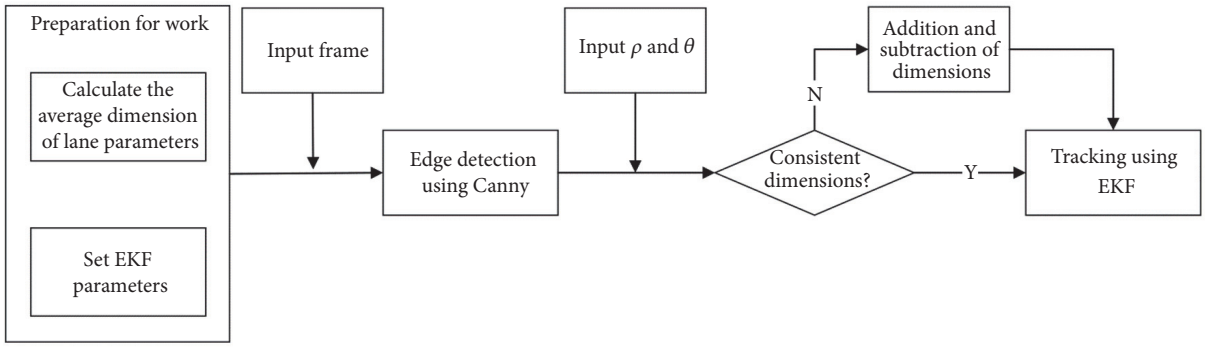


Figure 4: EKF Algorithm.

Table 1: Extended Kalman Filter algorithm module.

Initialization:	
	$\hat{x}_0 = E[x_0]$
	$P_0 = E[(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T]$
Prediction:	
(I) State Prediction:	$\hat{x}_{k\|k-1} = f(\hat{x}_{k-1})$
(II) State Prediction Error Covariance Matrix: $P_{k\|k-1} = F_{k-1}P_{k-1}F_{k-1}^T + Q_{k-1}$	
Where,	$F_{k-1} = \left.\frac{\partial f(x_{k-1})}{\partial x}\right\|_{x=\hat{x}_{k-1}}$
Error Correction:	
(I) Kalman Gain:	$G_x = P_{k\|k-1}H_k^T(H_kP_{k\|k-1}H_k^T + R_k)^{-1}$
Where,	$H_k = \left.\frac{\partial h(x_k)}{\partial x}\right\|_{x=\hat{x}_{k\|k-1}}$
(II) State Estimation:	$\hat{x}_k = \hat{x}_{k\|k-1} + G_k(z_k - h(\hat{x}_{k\|k-1}))$
(III) State Estimation Error Covariance Matrix: $P_k = (I - G_KH_k)P_{k\|k-1}$	

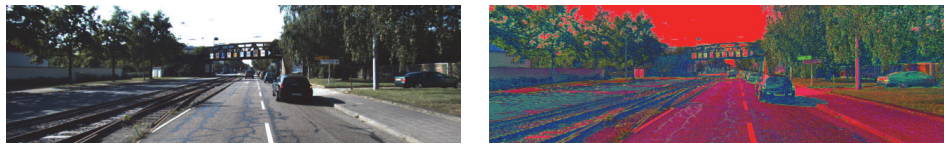
parameters ρ and θ of the lane based on the straight line model are obtained from the Hough transforms of Figures 2 and 3, the lane line can be tracked using the EKF. The EKF tracking algorithm is described in Table 1, the initial value of the parameter $\hat{x}_0$ and the initial value of the covariance P_0 are set as the unit matrix, and the predicted value of the current state is the tracking result of the previous state [21]. The real value of the current state is the sequence frame of the current reading; thus the tracking value of the current state (the optimal estimation result) can be obtained. This value is also used as the prediction value of the next state to realize the cyclic estimation of the lane parameters, that is, the tracking [23]. Table 1 shows Extended Kalman Filter algorithm module.

As shown in Figure 4, before inputting the image frame, we made preparations such as calculating the average value of vehicle parameter dimensions and setting EKF parameters. The input image frame is detected by the Canny edge operator and the resulting edge image is obtained. Then, we add the parameters ρ and θ of the lane line based on the straight line model that have been obtained by the Hough transform and determine whether the lane parameters and dimensions detected by the Hough transform are the same for all input frame images. If they are equal, use EKF for lane tracking, or enter the dimension addition and subtraction module to adjust the parameter dimension.

4. Experiments

4.1. Preprocessing. Preprocessing is an important part of image processing and an important part of lane detection. Preprocessing can help reduce the complexity of the algorithm, thereby reducing subsequent program processing time. The video input is a RGB-based colour image sequence obtained from the camera. In order to improve the accuracy of lane detection, many researchers employ different image preprocessing techniques.

Smoothing and filtering graphics is a common image preprocessing technique. The main purpose of filtering is to eliminate image noise and enhance the effect of the image. Low-pass or high-pass filtering operation can be performed for 2D images, low-pass filtering (LPF) is advantageous for denoising, and image blurring and high-pass filtering (HPF) are used to find image boundaries [24–26]. In order to



(a) RGB (b) HSV

Figure 5: Two images of different colour spaces. (a) RGB and (b) HSV colour transform.

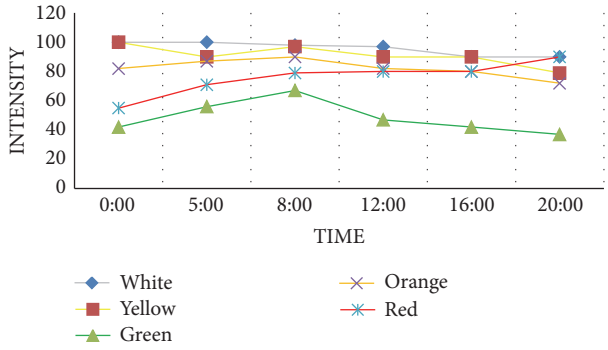


Figure 6: V-component values of representative colours under various illumination.

perform the smoothing operation, an average, median [8], or Gaussian [23] filter could be used. In [24], in order to preserve detail and remove unwanted noise, Xu and Li firstly use a median filter to filter the image and then use an image histogram in order to enhance the grayscale image [23].

4.2. Colour Transform. Colour model transform is an important part of machine vision, and it is also an indispensable part of lane detection in this paper. The actual road traffic environment and light intensity all produce noise that interferes with the identification of colour. We cannot detect the separation of white lines, yellow lines, and vehicles from the background. The RGB colour space used in the video stream is extremely sensitive to light intensity, and the effect of processing light at different times is not ideal. In this paper, the RGB sequence frames in the video stream are colour-converted into HSV colour space images. Figures 5(a) and 5(b) are images of RGB colour space and HSV colour space, respectively. HSV represents hue, saturation, and value [6]. As can be seen in Figure 6, the values of white and yellow colours are very bright in the V-component compared to other colours and are easily extracted, providing a good basis for the next colour extraction. Experiments show that the colour processing performed in the HSV space is more robust to detecting specific targets.

4.3. Basic Preprocessing. As shown in Figure 7, a large number of frames in the video will be preprocessed. The images are individually gray scaled, blurred, X-gradient calculated, Y-gradient calculated, global gradient calculated, thresh of frame, and morphological closure [25]. In order to cater for different lighting conditions, an adaptive threshold is implemented during the preprocessing phase. Then, we remove the spots in the image obtained from the binary conversion and perform the morphological closing operation. As can be seen from Figure 7, the basic preprocessed frames cannot be very good at removing noise. It can be seen from the results after the morphological closure that although preliminary lane information can be obtained, there is still a large amount of noise.

4.4. Adding Colour Extraction in Preprocessing. In order to improve the accuracy of lane detection, we add a feature extraction module in the preprocessing stage. The purpose of feature extraction is to keep any features that may be lane and remove features that may be nonlane. This paper mainly carries on the feature extraction to the colour. After the graying of the image and colour model conversion, we add the white feature extraction and then carry out the conventional preprocessing operation in turn. The process of the colour extraction proposed in this paper is shown in Figure 8.

4.5. Adding Edge Detection in Preprocessing. This paper has carried out edge detection two times successively; the first time is to perform a wide range of edge detection extraction in the entire frame image. In the second, the edge detection is performed again after the lane detection after ROI selection. This detection further improves the accuracy of lane detection [22]. This section mainly performs the overall edge detection on the frame image, using the improved Canny edge detection algorithm. The concrete steps of Canny operator edge detection are as follows: First, we use a Gaussian filter to smooth the image (preprocessed image), and then we use the Sobel operator to calculate the gradient magnitude and direction. Next step is to suppress the nonmaximal value of the gradient amplitude. Finally, we need to use a double-threshold algorithm to detect and connect edges. Figure 9 shows the image after extraction with Canny edge detection.

4.6. ROI Selection. After edge detection by Canny edge detection, we can see that the obtained edge not only includes the required lane line edges, but also includes other unnecessary lanes and the edges of the surrounding fences. The way to remove these extra edges is to determine the visual area of a polygon and only leave the edge information of the visible area. The basis is that the camera is fixed relative to the car, and the relative position of the car with respect to the lane is also fixed, so that the lane is basically kept in a fixed area in the camera.

In order to lower image redundancy and reduce algorithm complexity, we can set an adaptive area of interest

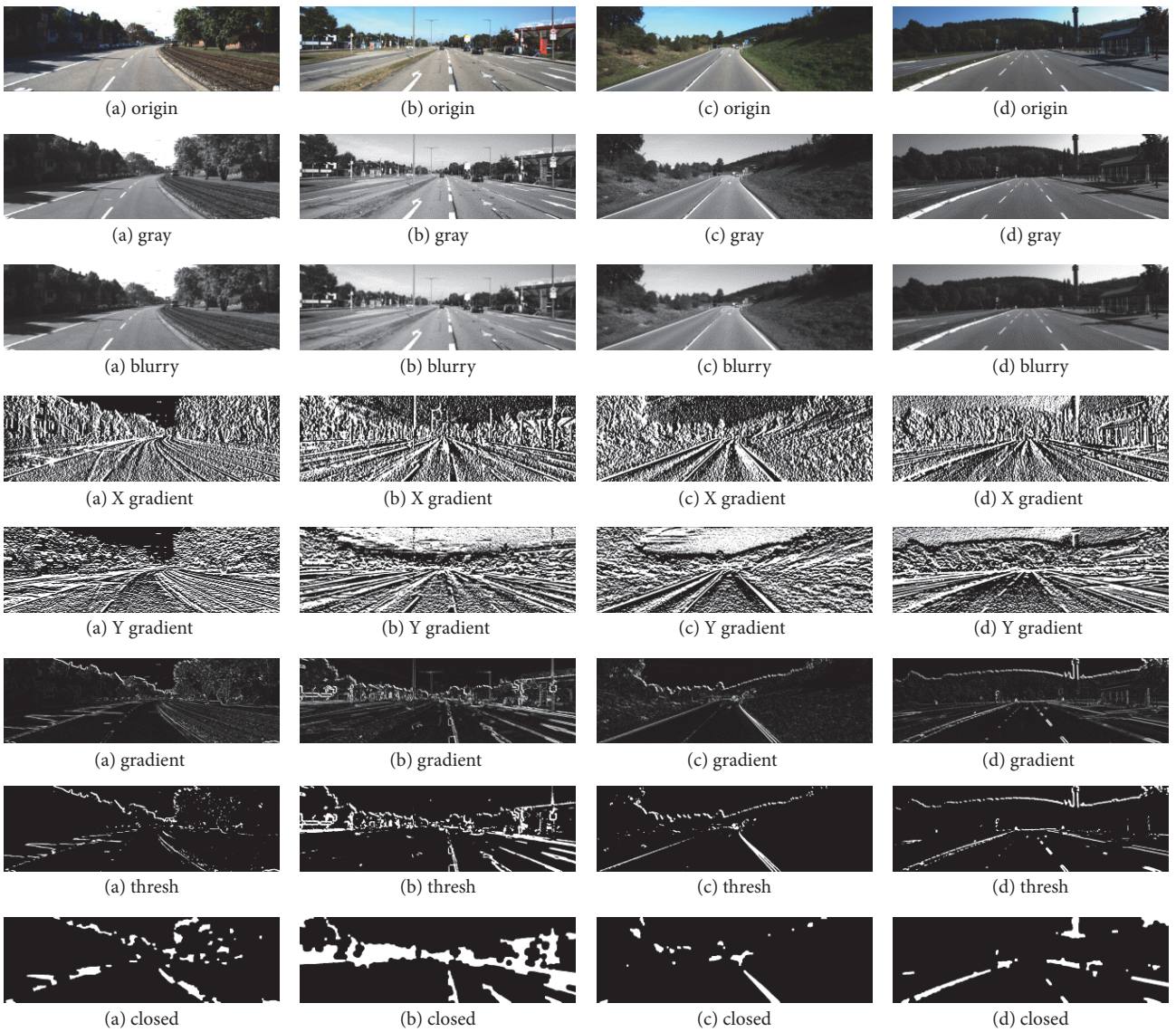


Figure 7: Basic preprocessing of sample frames (a), (b), (c), and (d).

(ROI) on the image [19]. We only set the input image on the ROI area and this method can increase the speed and accuracy of the system. In this paper, we use the standard KITTI road database [26]. We divide the image of each frame in the running video of the vehicle into two parts, and one-half of the lower part of the image frame serves as the ROI area. Figure 10 shows the ROI selection of sample frames (a), (b), (c), and (d) which are processed by the proposed preprocessing. The images of the four different sample frames have been able to substantially display the lane information after being processed by the proposed preprocessing method, but not only the lane information but also a lot of nonlane noise is present in the upper half of the image. So we cut out the lower half of the image (one-half) as the ROI area.

4.7. Lane Detection. The lane detection module is mainly divided into lane edge detection and linear lane detection. This section implements the basic functions of lane detection and performs lane detection based on improved preprocessing and the proposed ROI selection.

4.8. Edge Detection. Feature extraction is very important for lane detection. There are many common methods used for edge detection, such as Canny transform, Sobel transform, and Laplacian transform [18, 24]. We have selected Canny transform which is better. As shown in Figure 11, we performed Canny edge detection after the proposed ROI selection.

4.9. Lane Detection. The methods of lane detection include feature based methods and model-based methods. The method based feature is used in this paper to detect the colour and edge features of lanes in order to improve the accuracy and efficiency of lane detection.

There are two methods to achieve straight lane detection. One is to use the Hough line detected function encapsulated

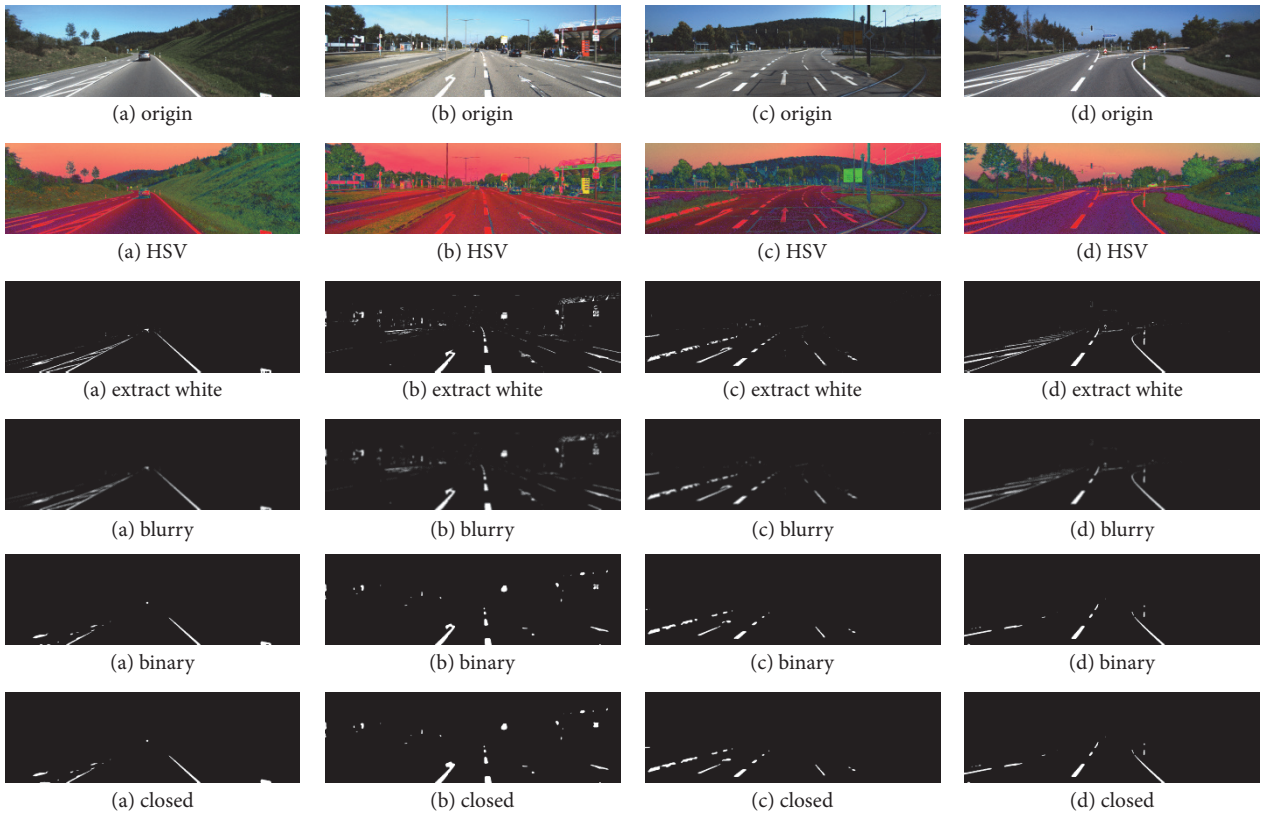


Figure 8: Adding white extraction in preprocessing of sample frames (a), (b), (c), and (d).

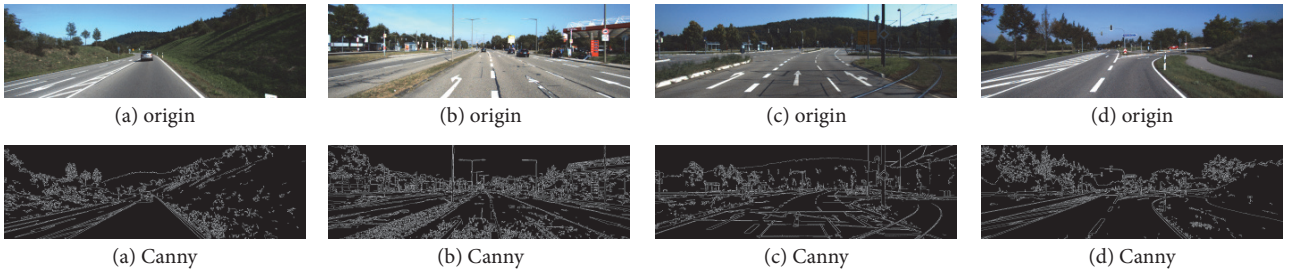


Figure 9: Canny edge detection of sample frames (a), (b), (c), and (d).

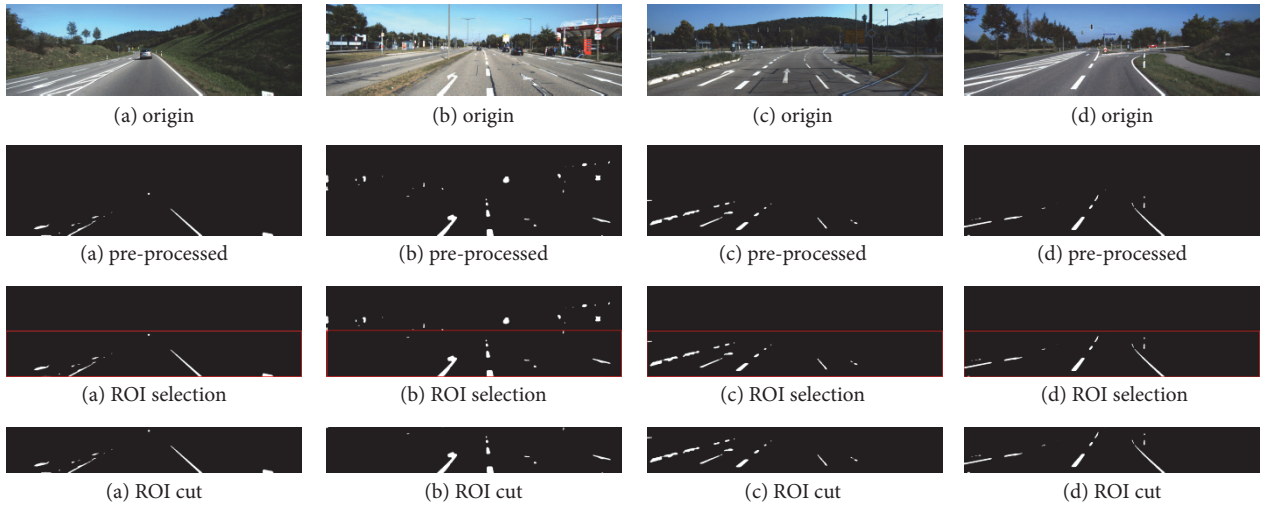


Figure 10: ROI selection of sample frames (a), (b), (c), and (d) based on the proposed preprocessing.

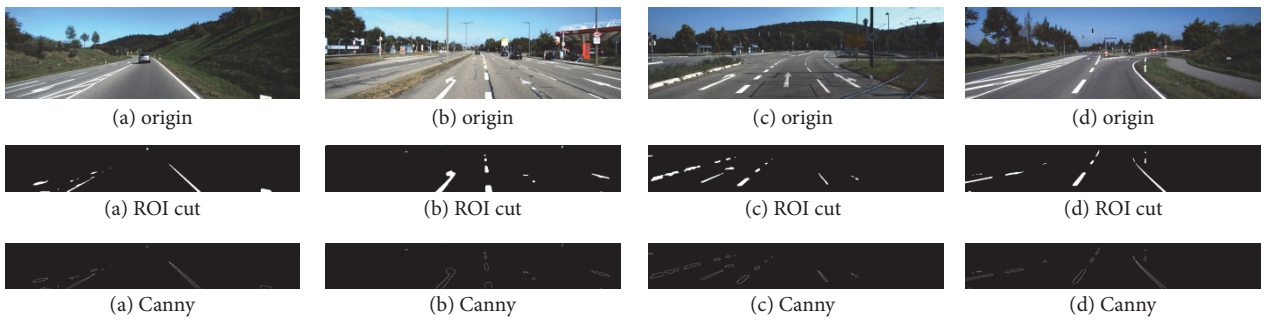


Figure 11: Canny edge detection of sample frames (a), (b), (c), and (d) after the proposed preprocessing.

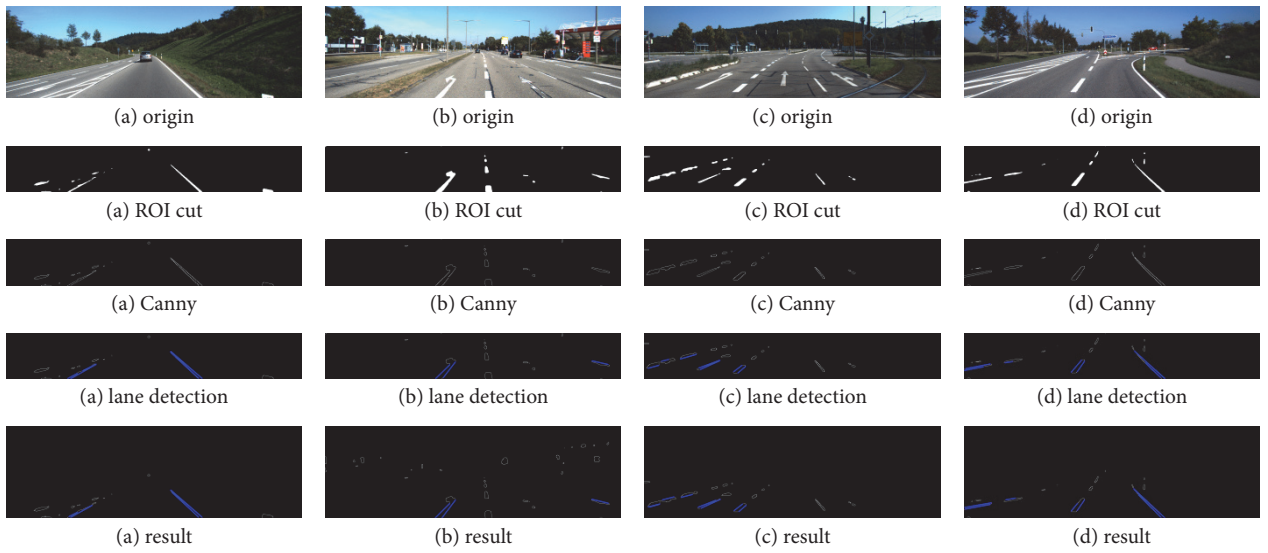


Figure 12: Lane detection using Hough of sample frames (a), (b), (c), and (d).

by the OpenCV library commonly used for image processing, and draw lane lines in the corresponding area of the original image. The other is self-programming. In the header file, the ROI area is traversed to perform line detection for a specific range of angles [27].

Both methods can be reflected in the video, and the first method runs faster. Since this article focuses on the accuracy and efficiency of lane detection, we chose the first method (Hough line function in the OpenCV library) to run faster for linear detection. Moreover, because the Hough transform is insensitive to noise and can process straight lines well, Hough transform is used to extract lane line parameters in each frame of the image sequence for lane detection.

In image processing, the Hough transform is used to detect any shape that can be expressed in a mathematical formula, even if the shape is broken or somewhat distorted. Compared with other methods, the Hough transform can find noise reduction better than other methods. The classic Hough transform is often used to detect lines, circles, ellipses, etc. As shown in Figure 12, lane detection uses Hough of sample frames (a), (b), (c), and (d).

4.10. Lane Tracking Using Extended Kalman Filter. After completing the lane detection, the next step is to track the lane, which is also a key technology for smart and automated vehicle (SAV) [24].

Image edge detection technology and linear lane detection are technologies used to detect lane; then EKF is used to track these parameters one by one [22]. In this way, the tracking of lane lines is converted into the tracking of lane line parameters, which not only improves the tracking speed, but also introduces the method of Kalman tracking to improve the tracking accuracy.

The experimental results are shown in Figures 13 and 14. The real-time tracking lane line is detected in the video stream. Figure 13 shows different results of lane detection at different times (i), (ii), (iii), and (iv) in one video. Figure 14 shows different results of lane detection at different times (i), (ii), (iii), and (iv) in another video.

5. Results and Discussion

Figure 15 shows the preprocessing of four frames of images. Frame (a.i) and frame (b.i) are processed by basic preprocessing (without white feature extraction), and frame (a.ii) and frame (b.ii) are processed by the proposed preprocessing

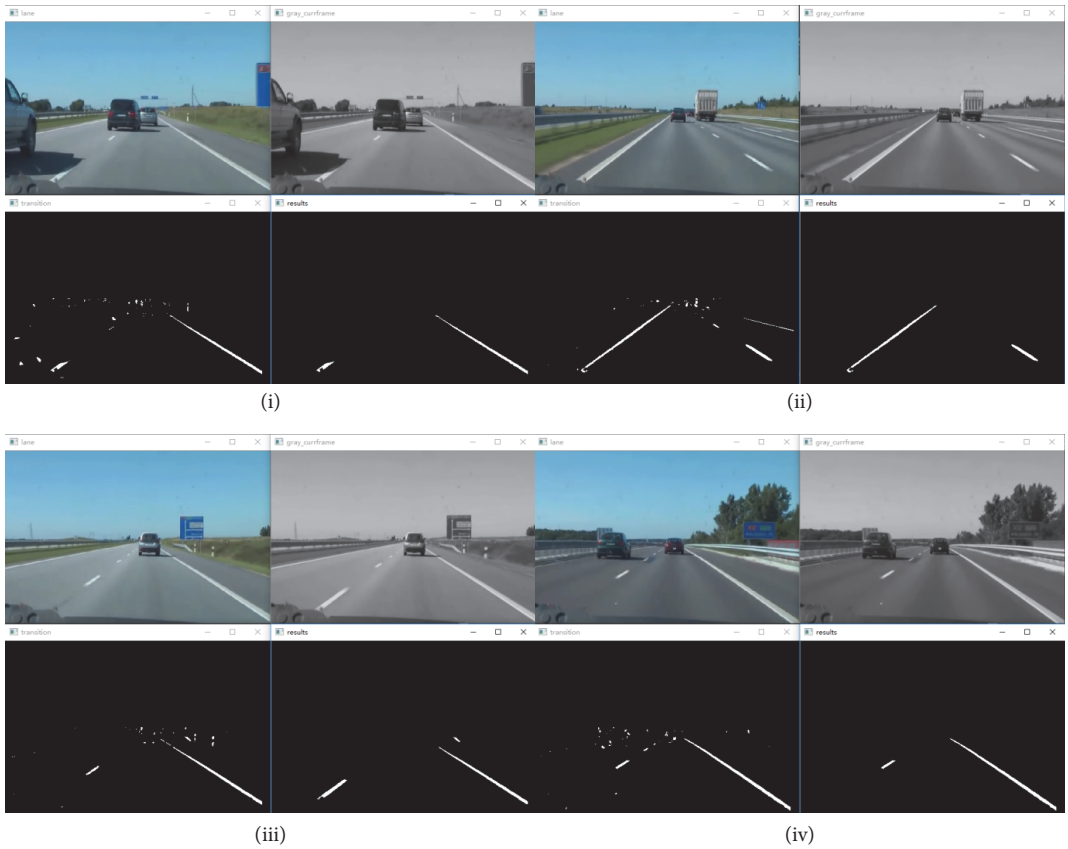


Figure 13: Different moments (i), (ii), (iii), and (iv) in one video.

(with the white feature extraction). From the Figure 15 we can see that frame (a.ii) and frame (b.ii) which are processed by the proposed preprocessing can display the lane line. But there is a large amount of white residue in frame (a.i) and frame (b.i), and it is difficult to detect lane lines. Therefore, the basic preprocessing of the frame does not work well for lane detection. In view of these, we propose to add HSV colour conversion in the preprocessing stage and then extract the white features of the frame before the blurry ones, so as to achieve a better detection effect and improve the detection accuracy.

As shown in Figure 16, frame (a) and frame (b) are extracted white features, respectively.

Most research scholars directly perform ROI selection on the original image. In this paper, a new ROI selection method is proposed. Experiments show that the proposed ROI selection can improve the accuracy and efficiency of lane detection.

Figures 17 and 18 show the ROI selection of white feature. It can be seen from the figures that ROI selection of white feature cannot accurately detect the area of lane line, which will eventually produce a great error.

Half of the input frames are proposed as ROI selection. As shown in Figure 19, the ROI selection implemented on the original image is followed by edge detection and lane detection on the selected ROI area. Compared with Figure 12, the result of the final lane detection contains many nonlane areas, and the effect of the lane detection is poor. The more the lane parameters are marked, the less efficient the calculation is. Therefore, the proposed method in this paper can lower the number of lane parameters, thereby reducing the calculation time and improving the detection efficiency.

To quantify the accuracy of lane detection, we used the correct detection rate to evaluate the performance of our proposed method for lane detection under the data set used. For better results of the proposed method, we first set the size of the image in the data set to the same size and randomly take 300, 500, 800, 1000, and 1500 images as a test set in training sets. In order to verify the excellence of our proposed method, as shown in Figure 20, we compared the detection efficiency of the basic preprocessing method for lane detection with the detection efficiency of the proposed preprocessing method. Moreover, as shown in Figure 21, we also compare the lane detection efficiency of the lane detection method that selects the ROI area only based on the lane colour with the lane detection efficiency of the proposed ROI selection method. From Figures 20 and 21, we can see that the results of the proposed method achieve the highest correct detection rate to prove the effectiveness of our proposed method.

6. Conclusions

In this paper, we proposed a new lane detection preprocessing and ROI selection methods to design a lane detection

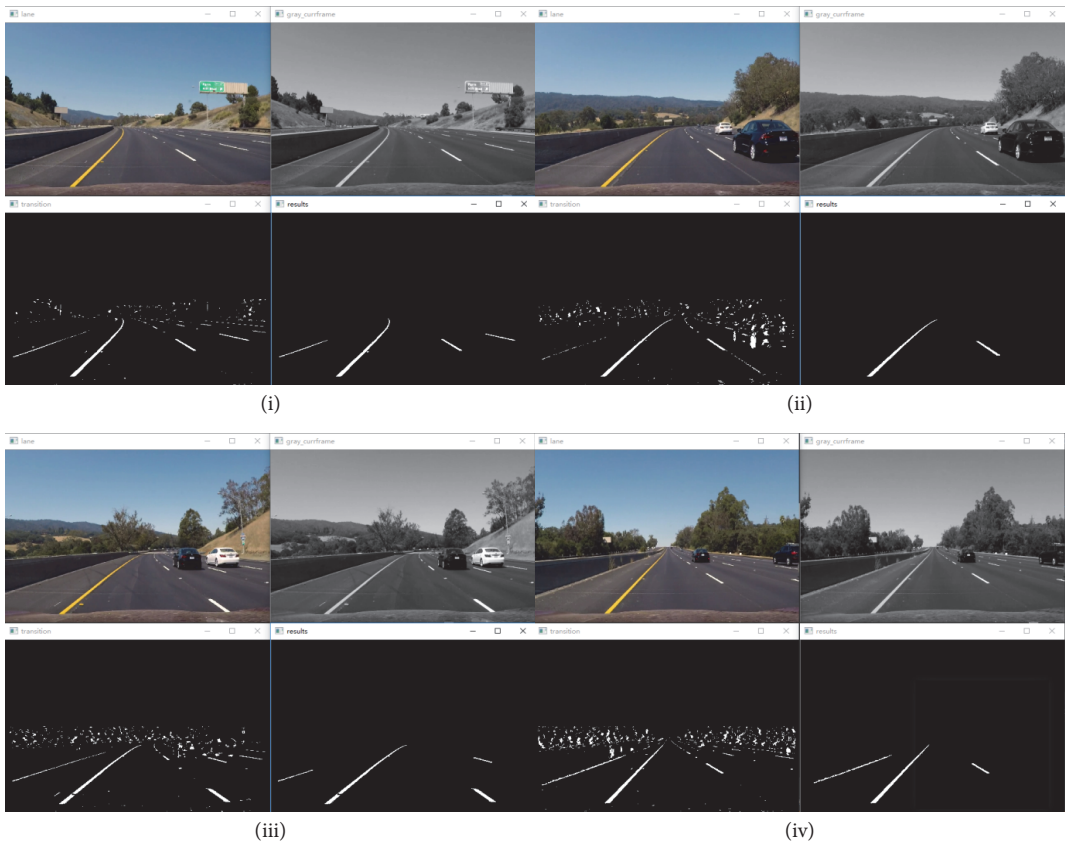


Figure 14: Different moments (i), (ii), (iii), and (iv) in another video.

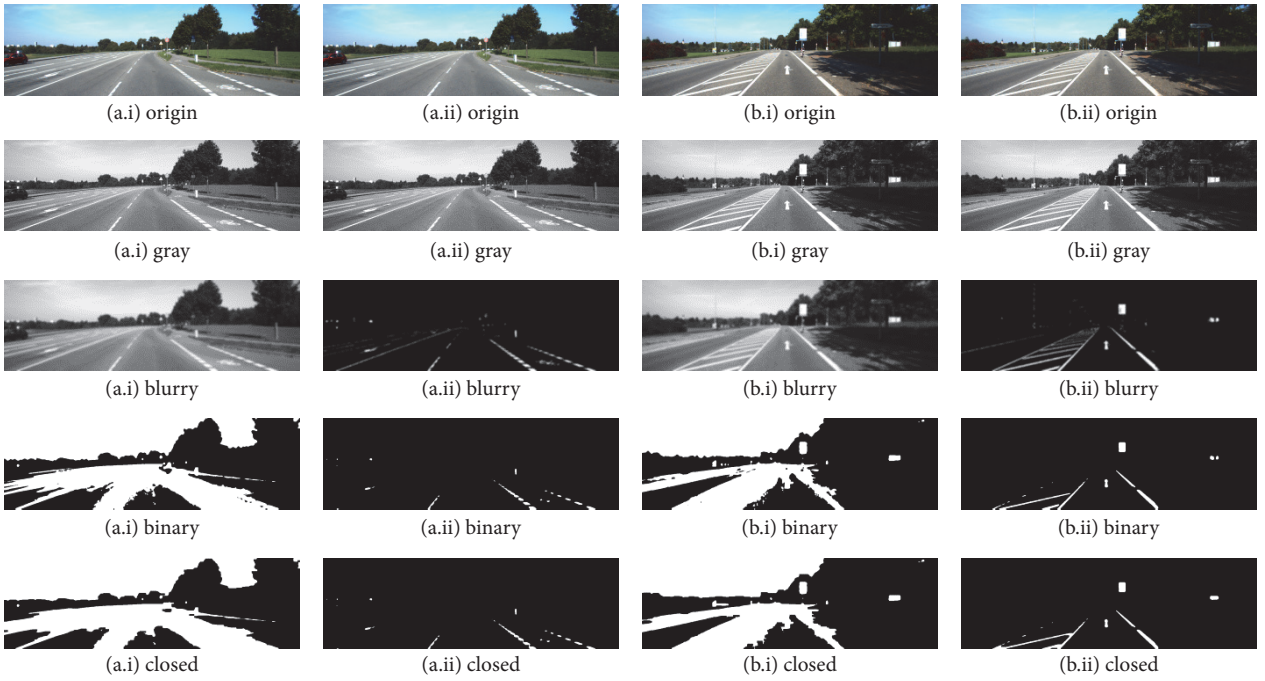


Figure 15: Comparison between the basic preprocessing method and the proposed preprocessing method. ((a.i) and (b.i)) Without extracting white before blurry one (the basic preprocessing method) and (a.ii) and (b.ii) with extracting white before blurry one (the proposed preprocessing method).

(a.i) extract white (a.ii) extract white (b.i) extract white (b.ii) extract white

Figure 16: White extraction of frames (a) and (b).

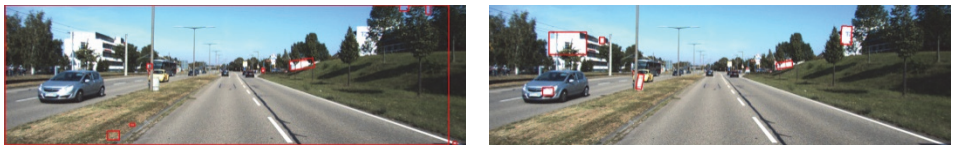
(a) origin (b) origin

(a) draw ROI (b) draw ROI

(a) ROI (b) ROI

Figure 17: ROI selection of white of the sample frames (a) and (b).

(c) origin (d) origin



(c) result (d) result

Figure 18: ROI selection of white of the sample frames (c) and (d).

system. The main idea is to add white extraction before the conventional basic preprocessing. Edge extraction has also been added during the preprocessing stage to improve lane detection accuracy. We also placed the ROI selection after the proposed preprocessing. Compared with selecting the ROI in the original image, it reduced the nonlane parameters and improved the accuracy of lane detection. Currently, we only use the Hough transform to detect straight lane and EKF to track lane and do not develop advanced lane detection methods. In the future, we will exploit a more advanced lane detection approach to improve the performance.

Data Availability

The proposed lane detection data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

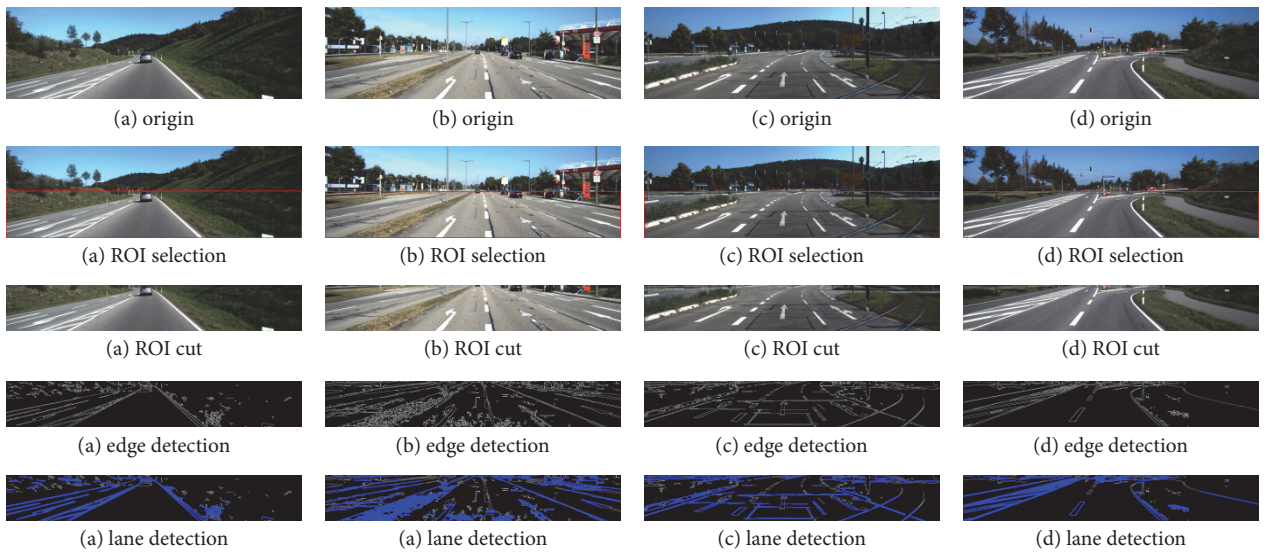


Figure 19: ROI selection and lane detection of sample frames (a), (b), (c), and (d).

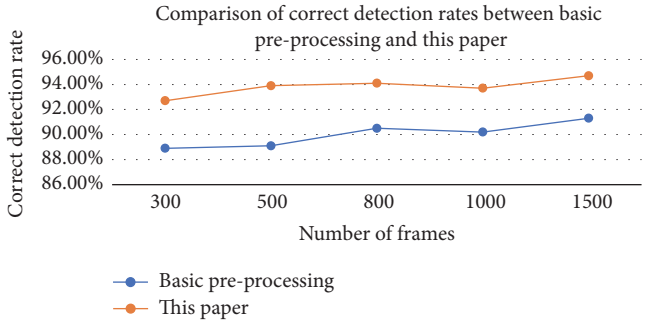


Figure 20: Comparison of correct detection rates between basic preprocessing and this paper.

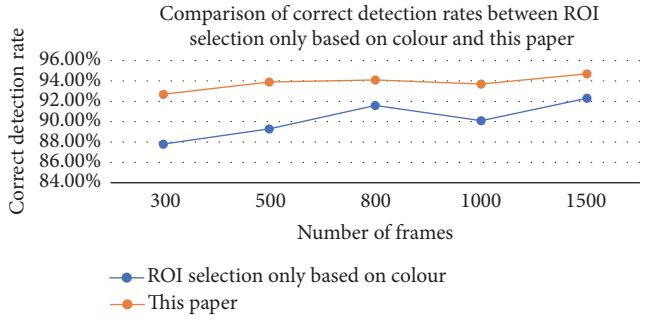


Figure 21: Comparison of correct detection rates between ROI selection only based on colour and this paper.

Acknowledgments

This paper was also supported by the project of Local Colleges and Universities Capacity Construction of Science and Technology Commission in Shanghai (no. 15590501300).

References

[1] D. Pomerleau, "RALPH: rapidly adapting lateral position handler," in *Proceedings of the Intelligent Vehicles '95. Symposium*, pp. 506–511, Detroit, MI, USA, 2003.

[2] J. Navarro, J. Deniel, E. Yousfi, C. Jallais, M. Bueno, and A. Fort, "Influence of lane departure warnings onset and reliability on car drivers' behaviors," *Applied Ergonomics*, vol. 59, pp. 123–131, 2017.

[3] P. N. Bhujbal and S. P. Narote, "Lane departure warning system based on Hough transform and Euclidean distance," in *Proceedings of the 3rd International Conference on Image Information Processing, ICIIP 2015*, pp. 370–373, India, December 2015.

[4] V. Gaikwad and S. Lokhande, "Lane Departure Identification for Advanced Driver Assistance," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 910–918, 2015.

[5] H. Zhu, K.-V. Yuen, L. Mihaylova, and H. Leung, "Overview of Environment Perception for Intelligent Vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 10, pp. 2584–2601, 2017.

[6] F. Yuan, Z. Fang, S. Wu, Y. Yang, and Y. Fang, "Real-time image smoke detection using staircase searching-based dual threshold AdaBoost and dynamic analysis," *IET Image Processing*, vol. 9, no. 10, pp. 849–856, 2015.

[7] P.-C. Wu, C.-Y. Chang, and C. H. Lin, "Lane-mark extraction for automobiles under complex conditions," *Pattern Recognition*, vol. 47, no. 8, pp. 2756–2767, 2014.

[8] M.-C. Chuang, J.-N. Hwang, and K. Williams, "A feature learning and object recognition framework for underwater fish images," *IEEE Transactions on Image Processing*, vol. 25, no. 4, pp. 1862–1872, 2016.

[9] Y. Saito, M. Itoh, and T. Inagaki, "Driver Assistance System with a Dual Control Scheme: Effectiveness of Identifying Driver Drowsiness and Preventing Lane Departure Accidents," *IEEE Transactions on Human-Machine Systems*, vol. 46, no. 5, pp. 660–671, 2016.

[10] Q. Lin, Y. Han, and H. Hahn, "Real-Time Lane Departure Detection Based on Extended Edge-Linking Algorithm," in

Proceedings of the 2010 Second International Conference on Computer Research and Development, pp. 725–730, Kuala Lumpur, Malaysia, May 2010.

[11] C. Mu and X. Ma, "Lane detection based on object segmentation and piecewise fitting," *TELKOMNIKA Indonesian Journal of Electrical Engineering*, vol. 12, no. 5, pp. 3491–3500, 2014.

[12] J.-G. Wang, C.-J. Lin, and S.-M. Chen, "Applying fuzzy method to vision-based lane detection and departure warning system," *Expert Systems with Applications*, vol. 37, no. 1, pp. 113–126, 2010.

[13] S. Srivastava, M. Lumb, and R. Singal, "Improved lane detection using hybrid median filter and modified hough transform," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 4, no. 1, pp. 30–37, 2014.

[14] J. Piao and H. Shin, "Robust hypothesis generation method using binary blob analysis for multi-lane detection," *IET Image Processing*, vol. 11, no. 12, pp. 1210–1218, 2017.

[15] J. Niu, J. Lu, M. Xu, P. Lv, and X. Zhao, "Robust Lane Detection using Two-stage Feature Extraction with Curve Fitting," *Pattern Recognition*, vol. 59, pp. 225–233, 2015.

[16] J. Son, H. Yoo, S. Kim, and K. Sohn, "Real-time illumination invariant lane detection for lane departure warning system," *Expert Systems with Applications*, vol. 42, no. 4, pp. 1816–1824, 2015.

[17] A. Mammeri, A. Boukerche, and Z. Tang, "A real-time lane marking localization, tracking and communication system," *Computer Communications*, vol. 73, pp. 132–143, 2016.

[18] C. J. Chen, B. Wu, W. H. Lin, C. C. Kao, and Y. H. Chen, "Mobile lane departure warning system in," in *Proceedings of the 2009 IEEE 13th International Symposium on Consumer Electronics*, pp. 1–5, 2009.

[19] J. W. Lee, C. D. Kee, and U. K. Yi, "A new approach for lane departure identification," in *Proceedings of the IEEE IV2003 Intelligent Vehicles Symposium*, pp. 100–105, 2003.

[20] J. W. Lee and U. K. Yi, "A lane-departure identification based on LBPE, Hough transform, and linear regression," *Computer Vision and Image Understanding*, vol. 99, no. 3, pp. 359–383, 2005.

[21] H. Xu and H. Li, "Study on a robust approach of lane departure warning algorithm," in *Proceedings of the EEE International Conference on Signal Processing System (ICSPS)*, pp. 201–204, 2010.

[22] A. Borkar, M. Hayes, and M. T. Smith, "Robust lane detection and tracking with Ransac and Kalman filter," in *Proceedings of the 2009 IEEE International Conference on Image Processing, ICIP 2009*, pp. 3261–3264, November 2009.

[23] H. Xu and H. Li, "Study on a robust approach of lane departure warning algorithm," in *Proceedings of the IEEE International Conference on Signal Processing System (ICSPS)*, pp. 201–204, 2010.

[24] H. Chen and Z. Jin, "Research on Real-Time Lane Line Detection Techonogy Based on Machine Vision," in *Proceedings of the 2010 International Symposium on Intelligence Information Processing and Trusted Computing (IPTC)*, pp. 528–531, Huanggang, China, October 2010.

[25] H. Aung and M. H. Zaw, "Video based lane departure warning system using hough transform," in *Proceedings of the International Conference on Advances in Engineering and Technology*, pp. 85–88, Singapore, 2010.

[26] J. Fritsch, T. Kuhnl, and A. Geiger, "A new performance measure and evaluation benchmark for road detection algorithms," in *Proceedings of the 16th International IEEE Conference on Intelligent Transportation Systems (ITSC '13)*, pp. 1693–1700, IEEE, The Hague, The Netherlands, October 2013.

[27] S.-C. Huang and B.-H. Chen, "Automatic moving object extraction through a real-world variable-bandwidth network for traffic monitoring systems," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 4, pp. 2099–2112, 2014.

Hindawi
Advances in Multimedia
Volume 2018, Article ID 8207201, 8 pages
https://doi.org/10.1155/2018/8207201

Research Article

Scene Understanding Based on High-Order Potentials and Generative Adversarial Networks

Xiaoli Zhao, Guozhong Wang, Jiaqi Zhang, and Xiang Zhang

School of Electronic and Electrical Engineering, Shanghai University of Engineering Science, Shanghai 201620, China

Correspondence should be addressed to Xiaoli Zhao; evawhy@163.com

Received 31 May 2018; Accepted 19 July 2018; Published 5 August 2018

Academic Editor: Shih-Chia Huang

Scene understanding is to predict a class label at each pixel of an image. In this study, we propose a semantic segmentation framework based on classic generative adversarial nets (GAN) to train a fully convolutional semantic segmentation model along with an adversarial network. To improve the consistency of the segmented image, the high-order potentials, instead of unary or pairwise potentials, are adopted. We realize the high-order potentials by substituting adversarial network for CRF model, which can continuously improve the consistency and details of the segmented semantic image until it cannot discriminate the segmented result from the ground truth. A number of experiments are conducted on PASCAL VOC 2012 and Cityscapes datasets, and the quantitative and qualitative assessments have shown the effectiveness of our proposed approach.

1. Introduction

Scene understanding, based on semantic segmentation, is a core problem in the field of computer vision, which has been applied to 2D image, video, and even volumetric data. Its goal is to assign each pixel a label and then provide complete understanding of a scene. Two examples of scene understanding are shown in Figure 1. The importance of scene understanding is highlighted by the fact that there are increasing applications, such as autonomous driving [1], human-computer interaction [2], robot technology, and augmented reality, to name a few.

The earliest scene parsing [3] is to classify 33 scenes for 2688 images on LMO dataset, which adopts label transfer technology to establish dense correspondences between the input image and each of the nearest neighbors using SIFT flow algorithm. State-of-the-art scene parsing frameworks are mostly based on fully convolutional network (FCN) [4]. FCN transforms the well-known networks-AlexNet, VGG, GooLeNet, and ResNet into fully convolutional ones by replacing the fully connected layers with convolutional ones. The key insight of FCN is to build the "fully convolutional" networks that take input of arbitrary size and produce corresponding-sized output with efficient inference and learning and realize end-to-end and image-to-image system of deep learning. For all these reasons and other contributions, FCN is considered as the milestone of deep learning. Although amounts of pooling operations enlarge the receptive fields of the convolution kernel of FCN, they lose the detailed location information, resulting in coarse segmentation result, which hinders its further application.

In order to refine the segmentation result, a postprocessing stage using conditional random field (CRF) is adopted after the output of system [5], which makes use of the fully connected pairwise CRF to capture the dependencies of pixels and achieve fine local details. Dilated convolution is a generalization of Kronecker-factored convolutional filters [6] which expand exponentially receptive fields without losing resolution by disposing of some pooling layers. The works [7] that make use of this technique allow dense feature extraction on any arbitrary resolution and then combine dilated convolutions of different scales to have wider receptive fields with no additional cost. Combined CRF with dilated convolution, Chen et al. [8] propose the "deeplab" system, which enlarges the receptive fields of filters at multiple scales and overcomes the disadvantage of location accuracy by using a fully connected CRF to response the final layer of network. In order to take the dense CRF with pairwise potentials as an integral part of the network, Zheng et al. [9] propose a model called CRFasRNN to refine the

(a) (b)

Figure 1: Examples of scene parsing: (a) image; (b) ground truth.

segmentation of FCN; they make it possible to fully integrate the CRF with a FCN and train the whole network end to end. Although CRF taking into account the correlation of pixels has improved the segmentation accuracy, it has also increased the computational complexity. To incorporate suitable global features, Zhao et al. [10] propose a pyramid scene parsing network (PSPNet), which extends the pixel-level feature to special designed pyramid pooling one in addition to traditional dilated convolution. This algorithm achieves the champion of ImageNet scene parsing challenge 2016.

In the above-mentioned algorithms, a common property is that all label variables are predicted either using unary potentials such as FCN or using pairwise potentials such as methods based on CRF. Despite the fact that pairwise potentials refine the accuracy of semantic segmentation, they only consider the correlation of two pixels. In an image, many pixels have the consistency across superpixels; high-order potentials should be effective in refining the segmentation accuracy. Arnab et al. [11] have integrated specific classes of high-order potentials in CNN-based segmentation models. This specific class may be object or superpixel and so on, for which we need to design different energy function to calculate high-order potentials, whose computation is complicated.

The generative adversarial nets (GAN) proposed by Goodfellow et al. [12] in 2014 can be characterized by training a pair of networks in competition with each other, in which an adversarial network can estimate the generative model without approximating many intractable probability computation. Because there is no need for any Markov chains or unrolled approximate inference network, GAN has drawn many researchers' attention in the domains of superresolution [13], image-to-image translation [14, 15], and image synthesis [16, 17], etc. We are interested in higher-order consistency without confining to a certain class. We also do not want to have complex probability or inference computation. Motivated by all kinds of GAN, we proposed a semantic segmentation framework based on GAN, which consists of two components: generative network and adversarial network. The former one generates the segmented image, and the latter one encourages the segmentation model to improve continuously the semantic segmentation result until it cannot be distinguished from the ground truth according to the value of loss function. Different from the classic GAN, we take the original image as the input of the generative network and and the output of generative network or corresponding ground truth as the input of the adversarial network; then adversarial network discriminates the similarity of two inputs. If the value of loss function of the framework is large, backpropagation is performed to adjust the parameters of the network; if the value of loss function satisfies the termination criterion, the output of the generative network is the final semantic segmentation result. The semantic segmentation framework based on GAN is shown in Figure 2. This approach takes into account the high-order potentials of an image because it differentiates the similarity between the segmented image and the corresponding ground truth in the whole image.

2. The Proposed Semantic Segmentation Approach

The aim of the proposed framework is to generate the semantic image $G(x_n)$ from an original image x_n. To achieve this goal, we design a generator network G and an adversarial network D. The generator is trained as a network parameterized by θ_G. These parameters denote the weights and are obtained by minimizing the loss function; then the output $G(x_n)$ of generator and the ground truth y_n are fed into the adversarial network parameterized by θ_D, in which the discriminator is trained to distinguish real or fake value. In order to achieve the desired result, it is important to design the architecture network and loss function.

2.1. The Architecture of Networks. Some works have shown that deeper network model can improve the performance of

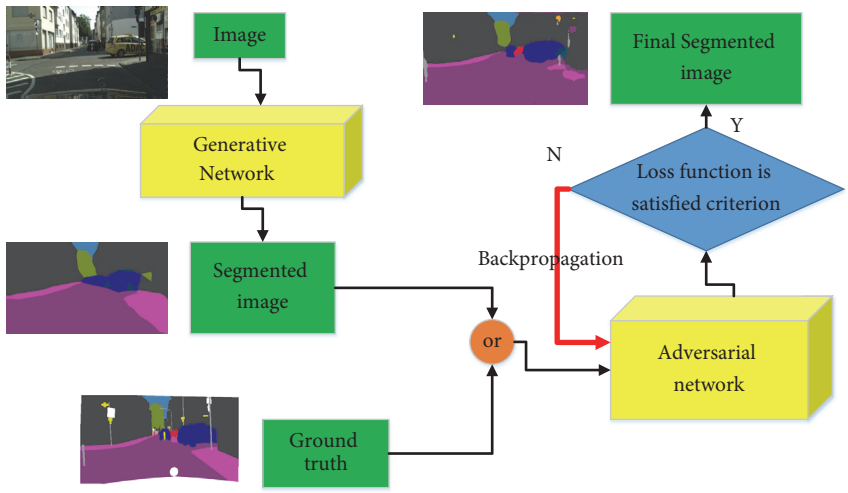


Figure 2: Overview of the proposed scene parsing framework.

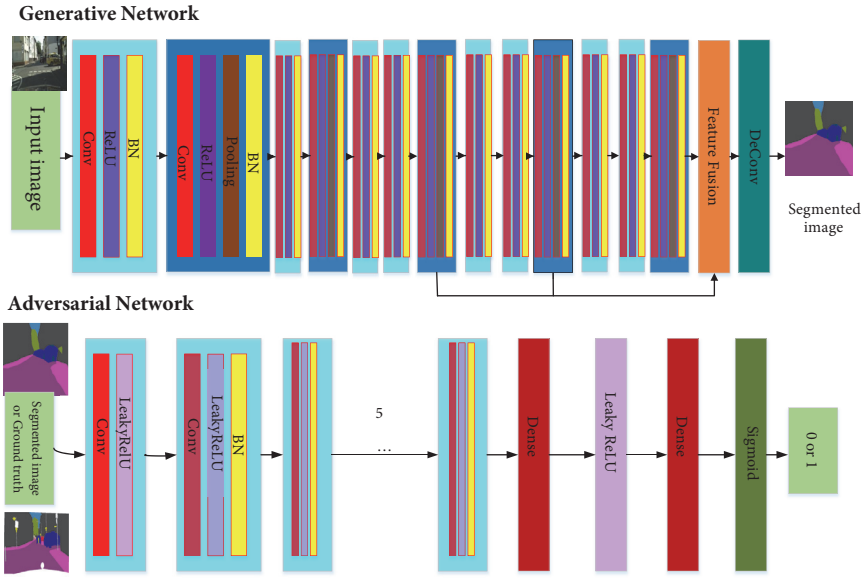


Figure 3: Architecture of generative and adversarial networks.

the segmentation and meanwhile make the architecture of the network complex, resulting in difficult training [18]. We make a compromise between the depth of the network and the performance of the algorithm.

In the generative network, which is shown in the first row of Figure 3, there are two modules of convolution and deconvolution. The role of convolution module is to extract the feature maps of an image, which consists of 10 layers. Each layer is composed of convolution, activation function, and batch normalization. The convolution is performed with 3×3 kernels and 64 feature maps followed by ReLU layer as the activation function, whose role is to conduct the nonlinear operation. Batch normalization is performed to avoid the network overfitting in each layer. Although pooling operations enlarge the receptive field of the network, they also reduce the accuracy of the segmentation. To improve the fine details of feature maps, the last three pooling outputs are integrated into one, on which deconvolution is performed to achieve the same size output with the original image.

To discriminate the ground truth from the segmented image, we train a discriminator network, which is illustrated in the second row of Figure 3. This architecture follows literature [13] to solve (4) in an alternating manner along with the generator. It contains eight convolution layers and uses LeakyReLU as the activation function. The convolution is conducted by 3×3 kernels, resulting in final feature maps of size 512, which are followed by two dense layers and a final sigmoid activation function to achieve a probability for classification.

2.2. Loss Function. In terms of information theory, cross entropy denotes the similarity of two variables; the more

similar the distribution of two variables, the smaller the cross entropy, so we adopt the cross entropy as the loss function. The definition of cross entropy is shown in the following:

$$CE\left(p,\widehat{p}\right) = -\sum_{i} p_i \log \widehat{p}_i. \tag{1}$$

where p and $\widehat{p}$ are the real value and predicted value. Equation (1) is Shannon entropy when p and $\widehat{p}$ are equal. In the multiple classification task, we use one-hot encoding cross entropy. Equation (1) can be rewritten as follows:

$$CE\left(y,\widehat{p}\right) = -\sum_{i} y_i \log \widehat{p}_i = -\log \widehat{p}_i. \tag{2}$$

where y specifies one pixel of ground truth and y_i represents 0 or 1.

The loss function of the proposed networks is a weighted sum of two terms. The first is a multiclass cross entropy term of a generator that encourages the segmented output similar to the input. We use $G(x)$ to denote the class probability map over C classes of size $H \times W \times C$ that the segmentation model generates given an input image x of size $H \times W \times C$. This segmentation model predicts the right class label at each pixel independently, which is described in the following:

$$GL\left(\theta_G\right) = l_{mce}\left(y, G\left(x\right)\right) = -\sum_{i=1}^{H\times W} \log \widehat{p}_i. \tag{3}$$

where $l_{mce}(G(x), y)$ represents the cross entropy loss function of multiple classification on an image of size $H \times W$, in which the class probability of per-pixel is predicted as $\widehat{p}_i$.

The second loss term represents the loss of the adversarial network. If the adversarial network can distinguish the output of generator from the ground truth, the loss value is large; otherwise, the loss is small. Because the loss is calculated based on the whole image or a large portion of it, this high-order statistics dissimilarity can be penalized by the adversarial loss term. We take the output of the adversarial network as $D(\cdot) \in [0, 1]$. Training the adversarial model is equivalent to minimizing the following binary classification loss:

$$AL\left(\theta_D\right) = l_{bce}\left(D\left(y\right), 1\right) + l_{bce}\left(D\left(G\left(x\right)\right), 0\right). \tag{4}$$

where l_{bce} denotes the binary cross entropy loss and $D(y)$ and $D(G(x))$ represent the label maps of adversarial network when the network input is the ground truth y or the output of a generator $G(x)$.

Given a data set of N original images x_n and the corresponding ground truth y_n, we define the total loss functions of the proposed semantic segmentation networks based on GAN as in the following:

$$\begin{aligned} TL\left(\theta_G, \theta_D\right) &= GL\left(\theta_G\right) + \alpha \times AL\left(\theta_D\right) \\ &= \sum_{n=1}^{N} \left(l_{mce}\left(y_n, G\left(x_n\right)\right) + \alpha\right. \\ &\left. \times \left(l_{bce}\left(D\left(y_n\right), 1\right) + l_{bce}\left(D\left(G\left(x_n\right)\right), 0\right)\right)\right). \end{aligned} \tag{5}$$

where α denotes weight factor. In this paper, we set it as 0.01.

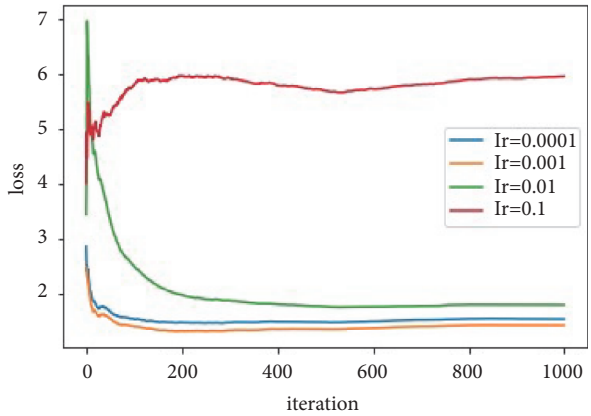


Figure 4: Divergence of different learning rate.

3. Experiments

To evaluate the proposed scene understanding algorithm based on GAN, we conduct some experiments on two widely used datasets, including PASCAL VOC 2012 [19] and urban scene understanding dataset Cityscapes [1]. We train networks on a NVIDIA Tesla K40 GPU and Intel Xeon E5 CPU using 2000 iterations and the batch size of size 16.

To quantitatively assess the accuracy of scene parsing, four performance indices are adopted: pixel accuracy (PA), mean pixel accuracy (MPA), mean intersection over union (MeanIoU), and frequency weighted intersection over union (FWIoU), whose formulations [20] are in (6)–(9). We assume a total of $k + 1$ classes, and p_{ij} is the amount of pixels of class i inferred to belong to class j. p_{ii} denotes the number of true positives, while p_{ij} and p_{ji} are usually represented as false positives and false negatives, respectively:

$$PA = \frac{\sum_{i=0}^{k} p_{ii}}{\sum_{i=0}^{k}\sum_{j=0}^{k} p_{ij}}. \tag{6}$$

$$MPA = \frac{1}{k+1}\sum_{i=0}^{k}\frac{p_{ii}}{\sum_{j=0}^{k} p_{ij}}. \tag{7}$$

$$MeanIoU = \frac{1}{k+1}\sum_{i=0}^{k}\frac{p_{ii}}{\sum_{j=0}^{k} p_{ij} + \sum_{j=0}^{k} p_{ji} - p_{ii}}. \tag{8}$$

$$FWIoU = \frac{1}{\sum_{i=0}^{k}\sum_{j=0}^{k}}\sum_{i=0}^{k}\frac{\sum_{j=0}^{k} p_{ij}p_{ii}}{\sum_{j=0}^{k} p_{ij} + \sum_{j=0}^{k} p_{ji} - p_{ii}}. \tag{9}$$

We use adaptive estimates of first-order moments (ADAM) [21] to optimize the algorithm because it requires little parameter-tuning, in which β_1 and β_2 are set to 0.9 and 0.999, respectively. We have also compared the divergence of different learning rate on the algorithm to select the optimal value, which is shown in Figure 4. According to this figure, we select 10^{-3} as the rate learning in these experiments.

3.1. Experiment 1: PASCAL VOC 2012. We carry out experiments on PASCAL VOC 2012 segmentation dataset, which

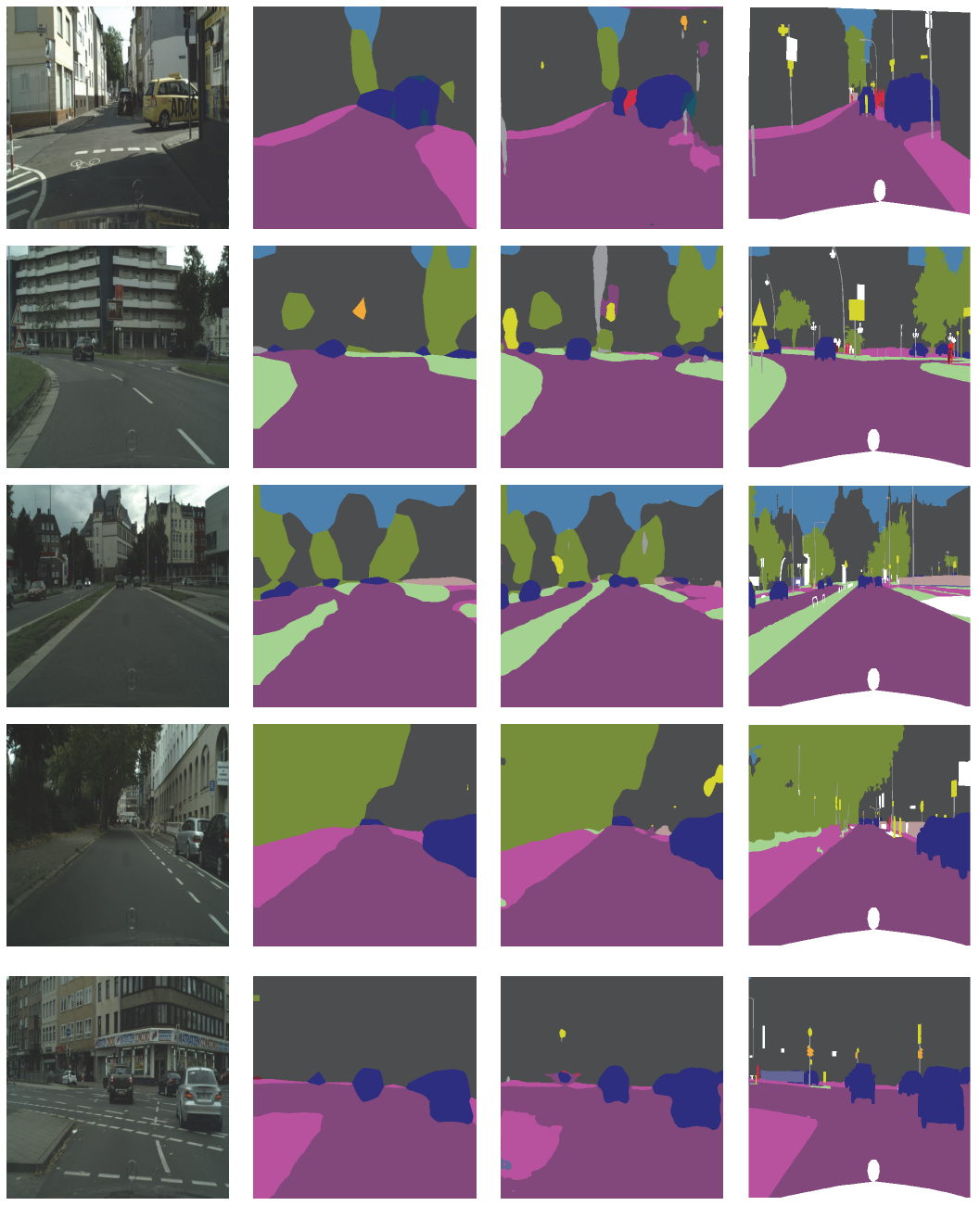
Figure 5: Comparison of different semantic segmentation algorithms on PASCAL VOC 2012: (a) original image; (b) FCN-8S; (c) DeepLab; (d) our method; (e) ground truth.

contains 20 object categories and 1 background class. Its augmented dataset [22] includes 10582, 1449, and 1456 images for training, validation, and testing. We have compared our method with the classic FCN [4] and popular DeepLab [5]: the accuracy of every class is shown in Table 1. Except for bicycle class, our approach achieves the highest accuracy on other 20 classes. Table 2 illustrates the four performance indices of different algorithms, PA, MPA, MeanIoU, and FWIoU. It is obvious that, from the left to right column, the accuracy of the algorithm gradually increases. The proposed approach gets the highest accuracy on these four performance indices.

To qualitatively validate the proposed method, several examples are exhibited in Figure 5. For "cat" in row one, our method gets the cat in accordance with the ground truth; however, FCN and DeepLab segment other noise regions. For "cow" and "child" in rows two and five, the details, such as leg, can be segmented in our method, while leg cannot be found in images using other two methods. In the fourth image, little cow and person are segmented in fine contour comparing with other two methods. In a word, the subjective quality of the segmented image using DeepLab is better than that using FCN; the segmented result using our method outperforms those using FCN and DeepLab.

3.2. Experiment 2: Cityscapes. Cityscapes [1] is a dataset for semantic urban scene understanding which was released in 2016. It contains 5000 high quality pixel-level finely annotated images collected from 50 cities in different seasons. The images, which consists of 2975, 500, and 1524 images for training, validation, and testing, are divided into 19 categories. Because this dataset is recently released, previous algorithms have not issued code for this dataset. We only do subjective assessment for Cityscapes using our method and FCN.

Several examples are shown in Figure 6. It is clear that our proposed method outperforms FCN and can achieve more details and distinguish road, building, cars, etc.



(a) (b) (c) (d)

Figure 6: Comparison of different semantic segmentation algorithms on Cityscapes: (a) original image; (b) FCN-8S; (c) our method; (d) ground truth.

4. Conclusion

In this paper, we propose a scene understanding framework based on generative adversarial networks, which trains the fully convolutional semantic segmentation network by adversarial network, and adopt high-order potentials to achieve the fine details and consistency of the segmented semantic image. We perform a number of experiments on two famous datasets, PASCAL VOC 2012 and Cityscapes. We analyze not only each class accuracy but also four accuracy indices by using different semantic segmentation algorithms. The quantitative and qualitative assessments have shown our proposed method achieves the best accuracy among all algorithms. In the future, we will do more experiments on Cityscapes dataset and address the misclassification caused by class imbalance.

Data Availability

The data used to support the findings of this study are included within the article.

Table 1: Each class accuracy.

Class Label	FCN-32s	FCN-16s	FCN-8s	DeepLab	Our Method
Background	92.8	92.8	91.2	92.6	93.8
aeroplane	75.4	76.2	76.8	83.5	86.1
bicycle	33.6	34.3	34.4	36.6	35.9
bird	67.7	68.2	68.9	82.5	87.7
boat	48.6	49.4	49.4	62.3	63.5
bottle	58.4	59.2	60.3	66.5	67.2
bus	73.4	74.6	75.3	85.4	87.1
car	74.2	73.2	74.4	78.5	82.3
cat	77.6	78.4	77.6	73.7	86.8
chair	21.8	22.5	21.4	30.4	32.3
cow	62.1	62.5	62.5	72.9	76.5
Dining-table	46.3	46.7	46.8	60.4	62.0
dog	68.4	69.8	71.8	78.5	81.1
horse	63.4	63.8	63.9	75.5	77.9
motorbike	76.2	76.4	76.5	82.1	84.3
person	72.3	72.4	73.9	79.7	82.4
Potted-plant	44.5	44.5	45.2	58.2	59.6
sheep	71.2	71.6	72.4	82.0	84.3
sofa	37.4	37.2	37.4	48.8	54.9
train	69.4	69.8	70.9	73.7	76.2
tv/monitor	54.3	54.5	55.1	63.3	64.2

Table 2: Four accuracy indices using different algorithms.

Accuracy	FCN-32s	FCN-16s	FCN-8s	DeepLab	Our Method
PA(%)	82.6	83.7	85.4	87.4	88.2
MPA(%)	61.3	61.8	62.1	69.8	72.6
Mean IOU(%)	63.5	64.5	67.2	70.3	73.9
FW IOU(%)	83.6	84.1	84.7	86.9	88.4

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by Shanghai Science and Technology Committee (no. 15590501300).

References

[1] M. Cordts, M. Omran, S. Ramos et al., "The Cityscapes dataset for semantic urban scene understanding," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, pp. 3213–3223, USA, July 2016.

[2] M. Oberweger, P. Wohlhart, and V. Lepetit, *Hands deep in deep learning for hand pose estimation*, Computer Science, 2015.

[3] C. Liu, J. Yuen, and A. Torralba, "Nonparametric Scene Parsing via Label Transfer," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 12, pp. 2368–2382, 2011.

[4] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015*, pp. 3431–3440, USA, June 2015.

[5] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected crfs," *Computer Science*, vol. 4, pp. 357–361, 2014.

[6] S. Zhou, J. N. Wu, Y. Wu, and X. Zhou, Exploiting local structures with the kronecker layer in convolutional networks, 2015.

[7] F. Yu and V. Koltun, *Multi-scale context aggregation by dilated convolutions*, arXiv preprint, 2015.

[8] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, 2018.

[9] S. Zheng, S. Jayasumana, B. Romera-Paredes et al., "Conditional random fields as recurrent neural networks," in *Proceedings of the 15th IEEE International Conference on Computer Vision, ICCV 2015*, pp. 1529–1537, Chile, December 2015.

[10] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid Scene Parsing Network," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6230–6239, Honolulu, HI, July 2017.

[11] A. Arnab, S. Jayasumana, S. Zheng, and P. H. Torr, "Higher Order Conditional Random Fields in Deep Neural Networks," in *Computer Vision – ECCV 2016*, vol. 9906 of *Lecture Notes in Computer Science*, pp. 524–540, Springer International Publishing, Cham, 2016.

[12] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza et al., "Generative adversarial nets," in *Proceedings of the 28th Annual Conference on Neural Information Processing Systems 2014, NIPS 2014*, pp. 2672–2680, Canada, December 2014.

[13] C. Ledig, L. Theis, F. Huszar et al., "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 105–114, Honolulu, HI, July 2017.

[14] P. Isola, J. Zhu, T. Zhou, and A. A. Efros, "Image-to-Image Translation with Conditional Adversarial Networks," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5967–5976, Honolulu, HI, July 2017.

[15] J. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks," in *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2242–2251, Venice, October 2017.

[16] X. Huang, Y. Li, O. Poursaeed, J. Hopcroft, and S. Belongie, "Stacked Generative Adversarial Networks," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1866–1875, Honolulu, HI, July 2017.

[17] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, *Generative adversarial text to image synthesis*, arXiv preprint, 2016.

[18] C. Szegedy, W. Liu, Y. Jia et al., "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '15)*, pp. 1–9, Boston, Mass, USA, June 2015.

[19] M. Everingham, L. van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

[20] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, P. Martinez-Gonzalez, and J. Garcia-Rodriguez, "A survey

on deep learning techniques for image and video semantic segmentation," *Applied Soft Computing*, vol. 70, pp. 41–65, 2018.

[21] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, arXiv preprint, 2014.

[22] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik, "Semantic contours from inverse detectors," in *Proceedings of the 2011 IEEE International Conference on Computer Vision, ICCV 2011*, pp. 991–998, Spain, November 2011.

Hindawi
Advances in Multimedia
Volume 2018, Article ID 8729645, 8 pages
https://doi.org/10.1155/2018/8729645

Research Article

A Power Control Algorithm Based on Outage Probability Awareness in Vehicular Ad Hoc Networks

Xintong Wu,[1] Shanlin Sun,[2] Yun Li,[2] Zhicheng Tan,[2] Wentao Huang,[2] and Xing Yao[2]

[1]*College of Information and Communication, Guilin University of Electronic Technology, Guilin, Guangxi 541004, China*
[2]*College of Electronic Information and Automation, Guilin University of Aerospace Technology, Guilin, Guangxi 541004, China*

Correspondence should be addressed to Yun Li; 44739235@qq.com

Received 30 March 2018; Revised 14 June 2018; Accepted 5 July 2018; Published 1 August 2018

Academic Editor: Shih-Chia Huang

This paper addresses the problem of adaptive power control based on outage probability minimization in Vehicular Ad Hoc Networks (VANETs), called a Power Control Algorithm Based on Outage Probability Awareness (PC-OPA). Unlike most of the existing works, our power control method aims at minimizing the outage probability and then is subject to the density of nodes in certain area. To fulfill power control, cumulative interference is assumed to be available at the transmitter of each terminal. The transmitters sent data by maximum power and then get the cumulative interference-aware outage probability. Furthermore, we build the interference model by stochastic geometric theory and then derive the expression between outage probability and cumulative interference. According to the expression, we adjust the transmitter power and optimize the outage probability. Simulation results are provided to demonstrate the effectiveness of the proposed power control strategies. It is shown that the PC-OPA can achieve a significant performance gain in terms of the outage probability and throughputs. Comparing MPC (Maximum Power Control algorithm) and WFPC (Water-Filled Power Control algorithm), the proposed PC-OPA decreased by 23% in terms of the outage probability and increased by 25% in terms of throughputs.

1. Introduction

Vehicular Ad Hoc Networks (VANETs) are a promising intelligent transportation system technology that offers many applications such as traffic and congestion control, safety assistance, and autodriving, all of which will drastically change and provide tremendous benefits to our lives [1–5]. The key technologies for VANETs, called Vehicle-to-Vehicle (V2V) communication, involve the networking of vehicles and other communication devices, e.g., roadside units (RSUs). Power control is the key to maintain the better connectivity of networks among devices, which is used for VANETs. However, unlike the current mobile ad hoc networks, VANETs have a lot of characteristics, such as broadcasting, random node mobility, time-space uncertainty transmission, and interference [6–8]; this makes VANETs more challenging. For example, when the transmitter with the maximum power control sends the data, in certain area big density of nodes brings more interference to the receiver, which results in high outage probability. Addressing this issue, a Power Control Algorithm Based on Outage Probability Awareness, simply named PC-OPA, is proposed.

In VANETs, traffic congestion is easy to happen [9–11]. When congestion happens, more density of nodes results in more interference, which leads to high outage probably. Furthermore, the retransmission results in more consumption, which leads to the poor connectivity in VANETs. If the high channel capacity is pursued, the probability of collision is greater. Therefore, compared to traditional power control algorithm, the PC-OPA aims at the optimal outage probability regardless of the optimal channel capacity. In [12], Power Control based on Broadcasting Messages (PCBM) algorithm is proposed, in which the transmission power is adjusted according to the distance of the nodes. Further, the broadcasting area of nodes is restricted, which reduces the interference among nodes. However, the constant position in nodes is hard to get due to the random mobility in nodes. Therefore, PCBM algorithm has rarely considered the random mobility in reality environment. In [13], in highway scene, Power Control based on Roadside Unit (PSRSU)

algorithm is proposed, in which the aim is to be sure of connectivity in nodes of one side. However, Roadside Unit (RSU) costs more. When the congestion happens, PCRSU algorithm is not good to solve the question of more interference because of the more density. In [14], Power Control based on Beacon (PCB) algorithm is proposed, in which action time of driver and access collision in nodes are considered. In long distance communication, the peak power control algorithm based on L beacon is used to obtain the SINR, whereas in short distance communication the minimum power control algorithm based on S beacon is used to satisfy the SINR. According to the communication distance, in PCB algorithm, difference beacon is selected to be adaptive to VANET. Therefore, PCB algorithm is widely used. However, when the speed of a vehicle is very fast, the power in transmitter is used more, which leads to more communication areas. Further, multiuser interference is serious due to more high density in nodes, which leads to high outage probability. At present for more interference of multiusers few powers control algorithm is considered.

In this paper, the performance of improvement of the proposed power control algorithm is achieved in terms of reducing cumulative interference of multiusers. Based on the stochastic-geometry theory in receiver the spatial user interference model is built. Further, the expression of outage probability is deduced. After the outage probability awareness, the transmitter adjusts the power. At last, PC-OPA is subject to obtaining the optimal outage probability and good throughput.

The rest of this paper is organized as follows. Section 2 discussed the related work on the system model, as well as its usage in the analysis of VANETs characteristics. Section 3 describes the mechanism of PC-OPA. Simulation results and the validation of the proposed matching mechanism are presented in Section 4. Finally, concluding remarks are given in Section 5.

2. System Model

VANETs have the obvious characteristics such as randomness and dynamics which makes interference of multiuser difficult to find. Therefore, multiuser's interference in power control of VANETs is rarely considered. Addressing this issue, the expression about interference is needed to describe the relationship between interference and outage probability, which is the theoretical support for power control algorithm. Therefore, according to the randomness, stochastic-geometry theory is applied to build the system model and then deduce the expression [15, 16]. In Figure 1, we present the model of urban road system.

Due to the fact that characteristics of VANETs are randomness and dynamic, multiple user interference model is established that node random arrived at some region, which can be regarded as stochastic point process. Using identical probability p ($0 \le p \le 1$), any nodes are joined by edges among N ($N \ge 1$) nodes. The total of edges is random variable and average value of edges is $pN(N-1)/2$. When $N \longrightarrow \infty$, we consider a set of transmitting nodes with locations specified by a homogeneous Poisson Point Process (PPP) [17], $\pi(\lambda) = \{x_i \in R^2, i \in Z\}$, of transmitting nodes i on the infinite two-dimensional plane. The nodes of random walk obey independent and uniform distribution and have the mobility and substitutable. Let h_i and h_j denote the random walk between two adjacent vehicles. Let V_i and V_j denote the speed of h_i and h_j. Therefore, the probability density of TX within communication coverage area is

$$f_{T_f}(\lambda) = \int_{-\infty}^{\infty} f_{T_f}(t)\, e^{-\lambda t} dt \tag{1}$$

Within communication coverage area of h_i, multiple user interference increased with density and mobility of nodes and then the information may not be decoded properly in target node, while outage probability increased significantly. We assume that network tends to be infinity, of Palm distribution [18] and Slivnyak theorem [19]; according to the requirement, the interference of receiver is analyzed by conditional distribution of TX and follows a homogeneous Poisson Point Process, where Poisson Point Process is moved. The signal-to-interference-and-noise radio seen at the RX_0 is

$$SINR = \frac{P_0 h_0 d_0^{-\alpha}}{\sum_{i=1}^{n} P_i h_i d_i^{-\alpha} + N_0} \tag{2}$$

where $I = \sum_{i=1}^{n} P_i h_i d_i^{-\alpha}$, denoted by multiple user interference; therefore,

$$SINR = \frac{P_0 h_0 d_0^{-\alpha}}{I + N_0} \tag{3}$$

where N_0 is background noise, P_0 is transmission power, P_i is transmission power of other users, h_0 is channel gain, and d is propagation distance. Therefore, the reference node has effects on background noise and on interference of other users. In Figure 2, we present the relationship between receiver and interference.

According to stochastic geometry, we consider a Vehicular Ad Hoc network that has the following key properties.

(1) Transmitter node locations are modeled by a homogeneous spatial Poisson Point Process. The number of random nodes in two-dimensional arbitrary finite area $A \in R^2$ is limited, which is called local finiteness of Poisson Point Process, and then any nodes' locations are nonoverlapping.

(2) Suppose that bounded A and B are disjoint areas, where $A, B \in R^2$, and $\Pi(A)$ and $\Pi(B)$ are independent random variables, where $\Pi(\cdot)$ denotes the set of Poisson Point Process in plane.

(3) The density of bounded disjoint area is superposition; in other words, aiming at characteristics of random mobility in VANETs, λ_1 and λ_1 random process is assumed to be a $\lambda_1 + \lambda_2$ homogeneous Poisson Point Process.

(4) According to theorem of Slivnyak, when moving and removing of nodes, the distribution of homogeneous Poisson Point Process will not be affected.

In short, we introduce theories and properties of random geometric, by space accumulated interference model building in VANETs; it is seen that accumulated interference and outage probability increased with density of nodes, which lead to the network throughput decreasing significantly.

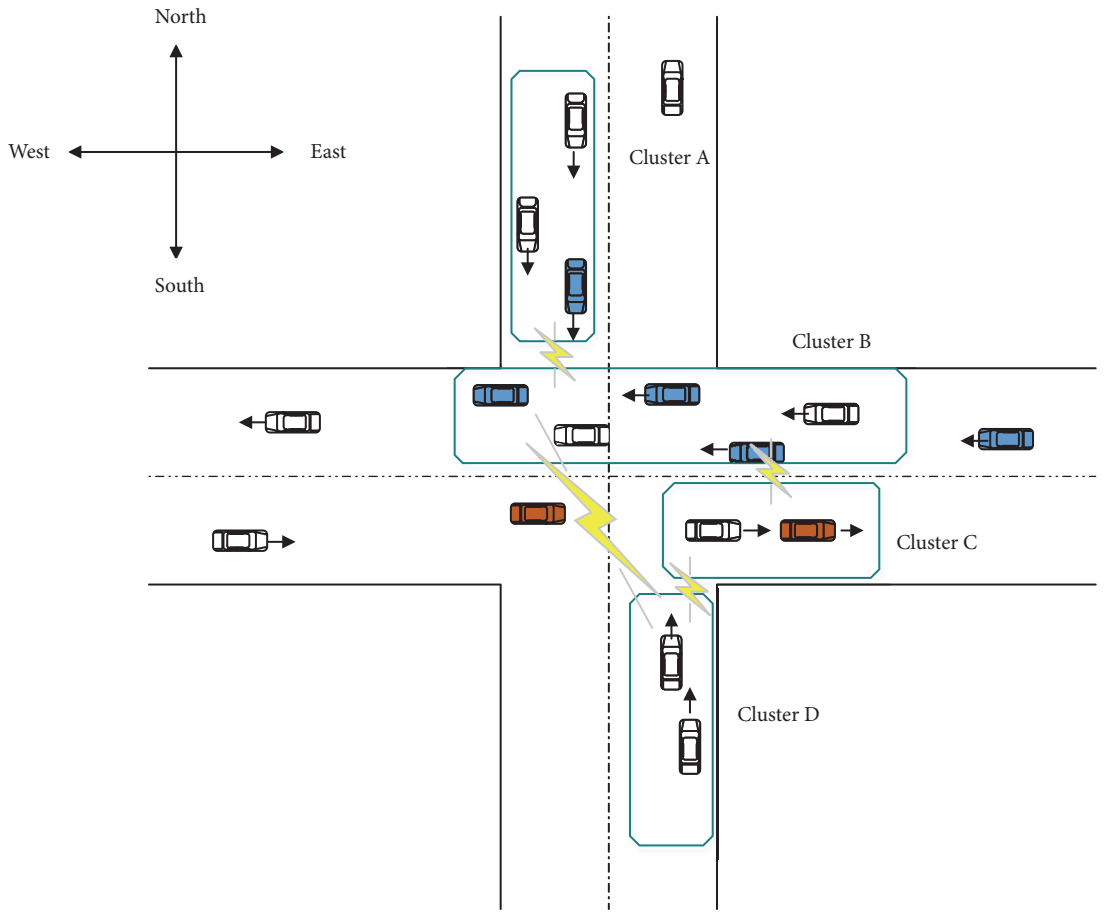


Figure 1: Urban road system model [3].

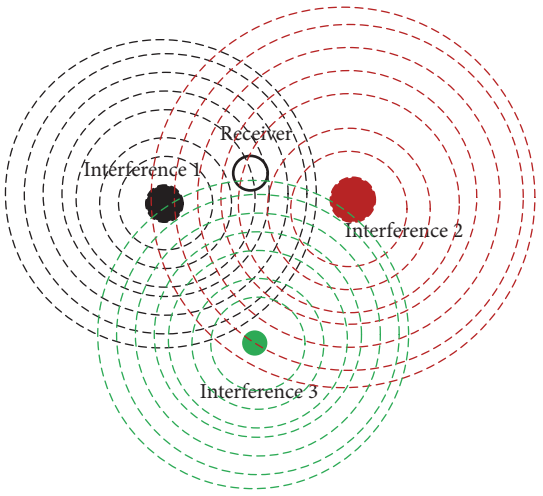


Figure 2: The multiple user interference [20].

3. The Mechanism of Power Control Algorithm Based on Outage Probability Awareness

In this section, we consider a power control algorithm that sends data with a maximum power to make a deduction of the formula of outage probability and then adjusts transmission power on the basis of outage probability information of awareness [21]. Finally, optimal outage probability and network throughput were obtained by PC-OPA.

3.1. Sending Data with a Maximum Power. SINR is shown as follows:

$$SINR = \frac{P_0 h_0 d_0^{-\alpha}}{\sum_{i=1}^{n} P_i h_i d_i^{-\alpha} + N_0} = \frac{S}{I + N_0} \tag{4}$$

where $S = P_0 h_0 d_0^{-\alpha}$, if $SINR < \beta$, the thesis holds that network transmission is interrupted. In accordance with statistical law, stochastic node sets distributed in space are called Poisson Point Processes. Suppose $\prod(x_n)$ satisfies $\prod_x = (x_n + x)$, $\prod$ and $\prod_x$ have the same distribution, and then $\prod$ is homogeneous Poisson Point Processes. Therefore, $\prod(B)$ obeys the Poisson distribution in a bounded domain of B, and the bounded function $\Lambda(B) = \lambda v_d(B)$ is a measurement.

$$\Pr\left(\prod(B) = \mathrm{k}\right) = \exp\left(-\lambda v_d(B)\right) \frac{\lambda v_d(B)}{\mathrm{k}!} \qquad \mathrm{k} = 0, 1 \cdots \tag{5}$$

where $v_d(B)$ is Lebesgue measure, namely, area of B. λ is intensity or average density of unit space. It is based on

such an assumption that location of interference sources obeys the Poisson Point distribution and interference power is function of power law decay of transmission distance. The accumulated interference signal in receiver constitutes the shot noise in two-dimensional space $I(x)$; we obtain that

$$I(x) = \sum_{x_i \in \Pi(\lambda)} h_i l\left(\left|x_i - x\right|\right) \tag{6}$$

where h_i is small scale power fading factor.

According to the above properties, when data is sent with a maximum power, outage probability is as follows:

$$\Pr_{\text{outage}}\left(SINR < \beta\right) = \Pr_{\text{outage}}\left(\frac{S}{(I + N_0)} < \beta\right) \tag{7}$$

where $I = \sum_{d_i \in \Pi(\lambda) \cap b(0,a)}^{n} h_i |d_i|^{-\alpha}$ denotes accumulated interference with area $b(0, a)$ of radius a; from the definition of (7), we obtain

$$\Pr\left(SINR < \beta\right) = 1 - \Pr\left(SINR > \beta\right) \tag{8}$$

$\Pr(SINR > \beta)$ is success probability:

$$\begin{aligned}\Pr\left(SINR > \beta\right) &= \exp\left(-\frac{\beta d^{\alpha} N}{p}\right) E\left(e^{-\beta d^{\alpha} I}\right) \\ &= \Pr_{s,n} \Pr_{s,I}\end{aligned} \tag{9}$$

and $\Pr_{s,n}$, $\Pr_{s,I}$ denote the success probabilities taking into account only noise and interference, respectively. Since $s = \beta d^{\alpha}$, the Laplace transform of the accumulated interference of $\Pr_{s,I}$ is

$$\Pr_{s,I} = \exp\left(-\lambda c_d d^{\alpha} \beta^{\delta} E\left[h^{\delta}\right] E\left[h^{-\delta}\right]\right) \tag{10}$$

Applying here with $c_d = 4\lambda\pi r^2$, $E[h^{\delta}] = \Gamma(1 + \delta)$:

$$\Pr_{s,I} = \exp\left(-\lambda c_d d^{\alpha} \beta^{\delta} \Gamma(1 + \delta) \Gamma(1 - \delta)\right) \tag{11}$$

Outage probability in a closed-form expression is as follows:

$$\begin{aligned}\Pr_{\text{outage}} &= 1 - \Pr_{s,n} \Pr_{s,I} \\ &= 1 - \Pr_{s,n} \exp\left(-\lambda c_d d^{\alpha} \beta^{\delta} \Gamma(1 + \delta) \Gamma(1 - \delta)\right)\end{aligned} \tag{12}$$

with

$$L = -\lambda c_d d^{\alpha} \beta^{\delta} \Gamma(1 + \delta) \Gamma(1 - \delta) \tag{13}$$

$$\Pr_{\text{outage}} = 1 - \Pr_{s,n} \exp\left(-\lambda c_d d^{\alpha} \beta^{\delta} L\right) \tag{14}$$

We define ε as outage probability. Now consider network throughput of success delivery with constrained outage probability.

$$C = \lambda^{\varepsilon} (1 - \varepsilon) B \log\left(1 + \frac{S}{N}\right) \tag{15}$$

where B is bandwidth.

3.2. Adjusting Transmission Power. Adjust transmission power $P = \text{ph}^{-w}$ based on channel state information, where w is chosen in $[0, 1]$. Clearly, if $w = 0$, $P = \text{p}$ implies maximum transmission power; whereas $w = 1$, $P = \text{p}$ is channel inversion.

From function (10), we have that

$$\begin{aligned}SINR &= \frac{P_0 h^{-\omega} h_0 {d_0}^{-\alpha}}{\sum_{i=1}^{n} P_i h^{-\omega} h_i {d_i}^{-\alpha} + N_0} \\ &= \frac{P_0 h^{1-\omega} {d_0}^{-\alpha}}{\sum_{i=1}^{n} P_i h^{1-\omega} {d_i}^{-\alpha} + N_0}\end{aligned} \tag{16}$$

Adjusting transmission power, we obtain

$$\begin{aligned}\Pr'_{\text{outage}} &= 1 - \Pr_{s,n} \Pr'_{s,I} = 1 \\ &\quad - \Pr_{s,n} \exp\left(-\lambda c_d d^{\alpha} \beta^{\delta} \Gamma(1 + \delta) \Gamma(1 - \omega\delta)\right. \\ &\quad \left. \cdot \Gamma(-(1 - \omega)\delta)\right)\end{aligned} \tag{17}$$

$$L' = \Gamma(1 + \delta) \Gamma(1 - \omega\delta) \Gamma(-(1 - \omega)\delta) \tag{18}$$

Then, the outage probability in a closed-form expression is as follows:

$$\Pr'_{\text{outage}} = 1 - \Pr_{s,n} \exp\left(-\lambda c_d d^{\alpha} \beta^{\delta} L'\right) \tag{19}$$

From (19), L' is the accumulated interference of channel fading. It can be verified that outage probability $\Pr'_{\text{outage}}$ decreased with power control exponent w since transmission power is adjusted. In order to improve the network throughput, after derivation calculus to (19), we can get optimal solution of power control exponent w.

3.3. Adjusting behind Transmission Power of Outage Probability. Adjust behind transmission power of outage probability $\Pr'_{\text{outage}}$ to judge whether there is maximum value. If $\Pr'_{\text{outage}}$ is not maximum value, outage probability information is obtained with feedback CSI in sender. If $\Pr'_{\text{outage}}$ is maximum value, outage probability minimum is obtained with adjusting behind transmission power. Since $L(h) = E(h^{-\omega})E(h^{-(1-\omega)})$, $L(h)$ is convex function:

$$E\left(h^{-\omega}\right) E\left(h^{-(1-\omega)}\right) = \Gamma(1 - \omega\delta) \Gamma(-(1 - \omega)\delta) \tag{20}$$

Taking logarithm on (20),

$$\log L(h) = \log\left(E\left[X^{-\omega}\right] E\left[X^{\omega-1}\right]\right) \tag{21}$$

By Holder's inequality,

$$E[XY] \le \left(E\left[X^{p}\right]\right)^{1/p} \left(E\left[Y^{q}\right]\right)^{1/q} \tag{22}$$

Applying here with $1/p + 1/q = 1$ and $p = 1/T$, $q = 1/(1 - T)$,

$$\begin{aligned}&L\left(Th_1 + (1 - T)h_2\right) \\ &= \log\left(E\left[h^{-(Tx_1 + (1-T)x_2)}\right] E\left[h^{(Tx_1 + (1-T)x_2) - 1}\right]\right) \\ &\le \log\left(E\left[h^{x_1}\right]^{T} E\left[h^{x_2}\right]^{1-T} E\left[h^{x_1 - 1}\right]^{T} E\left[h^{x_2 - 1}\right]^{1-T}\right) \\ &= Th(x_1) + (1 - T)h(x_2)\end{aligned} \tag{23}$$

Table 1: Simulation parameter setting.

Simulation Area	2000m∗2000m
Number of Vehicle	0-160
Transmission Distance	100m-300m
Channel Bandwidth	5-20MHz
Signal-to-Noise Ratio	15-30dB
Doppler Frequency Shift	100-300Hz

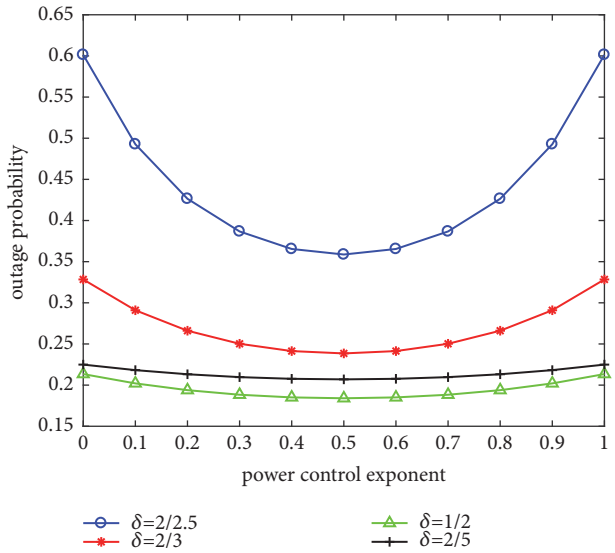


Figure 3: Power control exponents w versus outage probability.

Calculating the derivatives of (22),

$$\begin{aligned}&\left(L\left(Th_1+(1-T)h_2\right)\right)'\\&=E\left[h^{-T}\right]E\left[h^{T-1}\log h\right]-E\left[h^{T-1}\right]E\left[h^{-T}\log h\right]\end{aligned}\tag{24}$$

Function (20) is lowest when $T = 0.5$. The results show that transmission power is adjusted at $P = \mathrm{ph}^{-w} = \mathrm{ph}^{-0.5}$; the outage probability has minimum value.

4. Simulation and Results

Here, we present some numerical results to evaluate the performance our proposed PC-OPA strategies. We compared the outage performance of the proposed strategies with that of WFPC (Water-Filled Power Control Algorithm) and NPC (Non-Power Control algorithm). Assume that simulation area is 2000 m∗2000m; the numbers of nodes vary from 0 to 160. The simulation parameters are shown in Table 1.

In Figure 3 we present relationship between outage probability and power control exponent. Path loss exponents for different environments are shown in Table 2. Figure 3 is for the case of $2 < \alpha < 6$, where four different values of α, i.e., $\alpha = 2.5$, $\alpha = 3$, $\alpha = 4$, and $\alpha = 5$ are assumed. Different parameters represent the different environments for wireless channel. As is shown, the PC-OPA is more effective and achieves the minimization of the outage probability. $w = 0$ represents maximum transmission power, whereas $w = 1$ is channel inversion. Clearly, $w = 0.5$ achieves a significant performance gain in terms of the outage probability regardless of the radio environment, whereas $w = 0$ and $w = 1$ are seen to be essentially equivalent, which is high cumulative interference and outage probability in receiver. This simulation is provided to demonstrate the effectiveness of the proposed power control strategies.

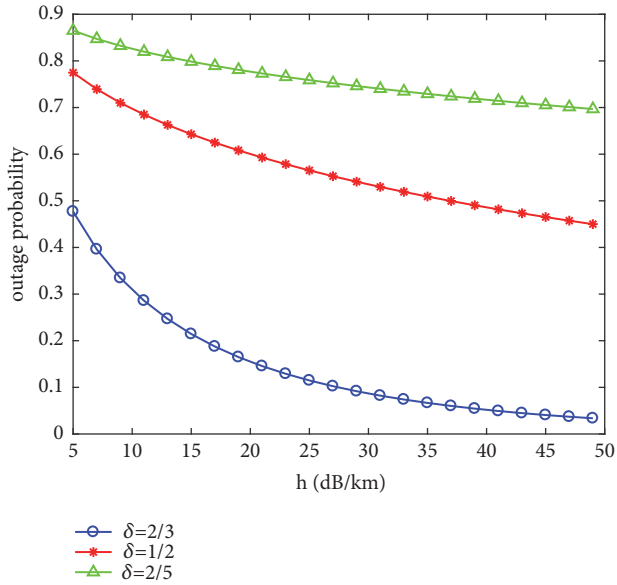


Figure 4: Outage probability versus h (dB/km) for absorption factor.

Table 2: Path loss exponents for different environments.

Environment	Path Loss Exponent, α
Free space	2
Urban area cellular radio	2.7 to 3.5
Shadowed urban cellular radio	3 to 5
In building line-of-sight	1.6 to 1.8
Obstructed in building	4 to 6
Obstructed in factories	2 to 3

Figure 4 is for the case of $2 < \alpha < 6$, where three different values of α, i.e., $\alpha = 3$, $\alpha = 4$, and $\alpha = 5$, are assumed. We plot the outage probability as absorption factor for the proposed PC-OPA strategy. Clearly, when the absorption factor varies from 5 to 50 dB/km, the outage probabilities reduce. The reason is that the accumulated interference declines as absorption factor h grows. Few accumulated interferences make it easy to be adaptive to the SINR of the receiver. Therefore, in different radio environment, the proposed PC-OPA is subjected to the minimization of outage probability according to the distribution of absorption factor.

Figure 5 shows that the optimized outage probability is a function of density of nodes for the proposed PC-OPA strategy. Clearly, as the density of nodes grows, the outage

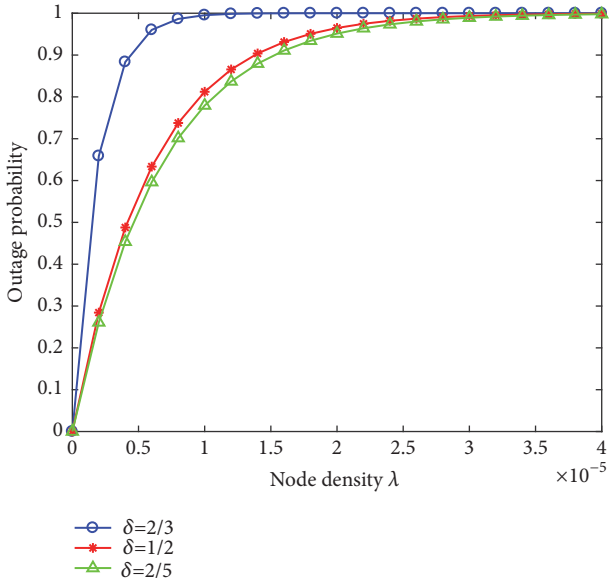


Figure 5: Outage probability versus λ density of nodes.

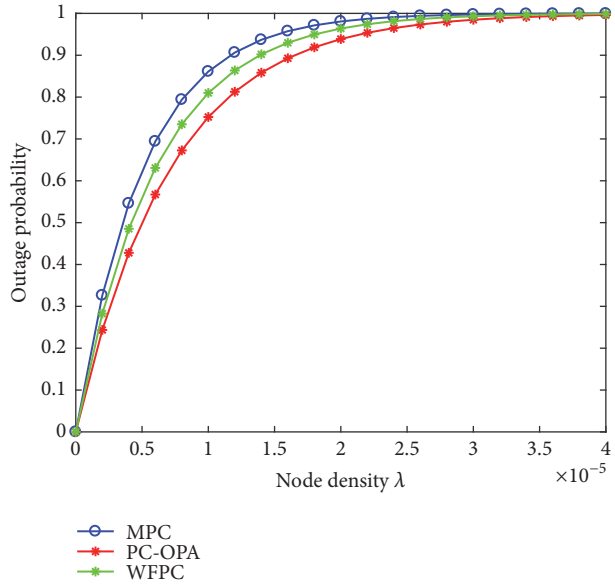


Figure 6: Outage probability for different algorithm versus λ density of nodes.

probability grows. The reason is that the more the number of the nodes is, the more accumulated the interference is. Then a lot of accumulated interference leads to more outages. Therefore, to reduce accumulated interference between multiusers, the density of nodes is limited in a certain area. According to the feedback of channel fading distribution, the transmitter adjusted the power to reduce the accumulated interference. The simulation results show that the proposed PC-OPA strategy achieves the optimum outage probability in different environment, in which the aim is to achieve the optimal outage probability by reducing accumulated interference.

In Figures 6, 7, and 8, we plot the outage probability as some parameters for the proposed the power control strategies, such as PPC (Peak Power Control), PC-OPA, and WFPC (Water-Filled Power Control). Considering above the parameters, we can see that outage probability increased with the density of nodes. As is shown, the PC-OPA strategy achieves the minimization of the outage probability. In the case of the same density, outage probabilities of PC-OPA, WFPC, and MPC are, respectively, 0.63, 0.75, and 0.86. The outage probability is significantly decreased by the PC-OPA compared with that by MPC, which is decreased by 23%. The MPC algorithm uses the maximum power to send the data. When the channel deteriorates beyond some point, transmissions are made in vain. The WFPC algorithm is greedy. However, the WFPC algorithm aims at achieving the optimal capacity regardless of the outage probability. More outage probability leads to deterioration of the network connectivity and brings more retransmission probability. Therefore, in this paper, the optimal outage probability algorithm is proposed. The PC-OPA achieves the optimal outage probability under multiusers.

In Figure 7, we plot the outage probability as a function of the distance from 0 to 250 m. The outage probability varies with the distances. It should be noted that the expression in

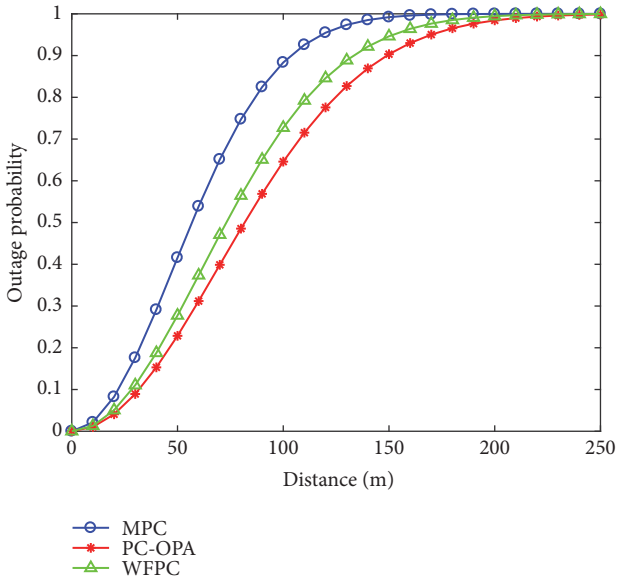


Figure 7: Outage probabilities versus distance.

(14) is for the case of channel fading. The method of PC-OPA provides channel fading variations for different distance and adaptively adjusts the transmission power according to the time varying characteristic of wireless channel; thus the outage probability of PC-OPA is lower compared to WFPC and MPC.

In Figure 8, we plot the outage probability as a function of the density of nodes for the three power control algorithms. As is shown in reality environment, there is serious Doppler frequency. When the density and the Doppler frequency increase, the accumulated interference in the receiver grows more. In certain area the density of nodes trends very fast to the saturation, which leads to the outage probability attaining

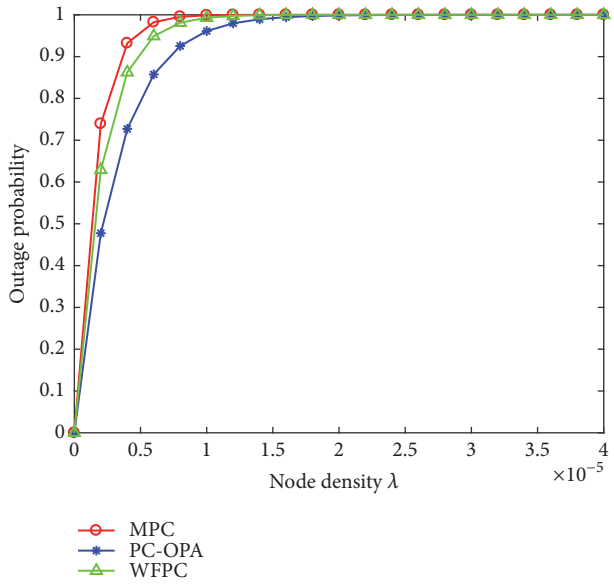


Figure 8: The outage probability for different algorithm versus λ density of nodes (Doppler frequency shift).

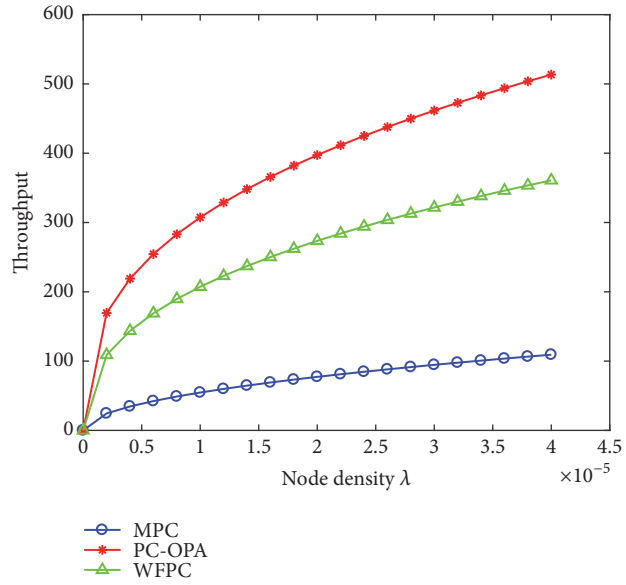


Figure 9: Throughput for different algorithm versus λ density of nodes.

to the maximum very fast. Therefore, the density of nodes is closer to the outage probability. The simulation results in Jack channel model show that when vehicle speed is equal to $50km/h$, the Doppler frequency is given for $f_d = 135Hz$. Compared with Figure 4, outage probability increased with the number of nodes for the same density. In the case of Doppler frequency, the outage probabilities of PC-OPA, WFPC, and MPC are, respectively, 0.83, 0.92, and 0.98. The outage probability is significantly decreased by the PC-OPA compared with that by MPC, which is decreased by 9%. The simulation results demonstrate that the reality of PC-OPA is better. The reason is that, considering the multiuser interference and joint with the feedback of CSI, the PC-OPA achieves the optimal outage probability.

The outage probability awareness algorithm is shown in Algorithm 1. Figure 9 shows that the throughput is varying as the node densities. With the increasing of the density nodes, the throughput grows more. Clearly, MPC is very fast trending to the saturation, and then WFPC is second. The PC-OPA achieves the most throughputs among the three algorithms. In the case of the same density of nodes, the network throughput of PC-OPA was significantly higher than that of WFPC and MPC, and then success delivery rate of PC-OPA is 600. The high delivery rate makes more throughputs, but results in more cumulative interference. As is shown, the PC-OPA can adjust the transmitter power according to CSI, in which the aim is to optimize the outage probability. Therefore, among three power control algorithms, the PC-OPA achieves the optimal outage probability and then achieves the most throughput.

5. Conclusion

In this paper, to address these issues, such as random mobility of nodes, interference in multiusers, and high outage probability, we proposed a power control algorithm, called simply PC-OPA. The PC-OPA analyzes the situation of multiple user interference through stochastic geometry and then establishes relationship between outage probability and channel accumulated interference. At last, the aim of the PC-OPA is to minimize the outage probability. Further, the throughputs increase, while the outage probability declines. Our simulation results validated the derived expression and confirmed the feasibility of the proposed PC-OPA. It is shown that, in general, not all the terminals need to use their maximum power consumption to achieve the best outage probability. If all the terminals use their maximum power consumption, it is easy to increase cumulative interference. Therefore, based on CSI, the PC-OPA in this paper is proposed. The simulation results show that the outage probability of the PC-OPA decreased by 23% and the throughput is increased by 25%, compared to MPC and WFPC.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the following projects: the National Natural Science Foundation of China through Grant 61571318, Guangxi Science and Technology Project (AC16380094, 1598008-29, and AA17204086), the Guangxi Nature Science Fund (2016GXNSFAA380226), Guangxi Nature Science Fund Key Project (2016 GXNSFDA380031),

1: set $P_0 = P_{max}$
2: $Pr_{outage} = 1 - Pr_{s,n}Pr_{s,I} = 1 - Pr_{s,n}\exp(-\lambda c_d d^{\alpha}\beta^{\delta}\Gamma(1+\delta)\Gamma(1-\delta))$
3: $Pr_{outage1} = 1 - Pr_{s,n}Pr_{s,I} = 1 - Pr_{s,n}\exp(-\lambda c_d d^{\alpha}\beta^{\delta}\Gamma(1+\delta)\Gamma(1-\delta))(CSI)$
4: **if** $Pr_{outage1} = Pr_{outage}$ **then**
5: $P = P_0 h^{-\omega}$
6: $Pr'_{outage} = 1 - Pr_{s,n}Pr'_{s,I} = 1 - Pr_{s,n}\exp(-\lambda c_d d^{\alpha}\beta^{\delta}\Gamma(1+\delta)\Gamma(1-\omega\delta)\Gamma(-(1-\omega)\delta))$
7: **else**
8: $Pr_{outage1} = 1 - Pr_{s,n}Pr_{s,I} = 1 - Pr_{s,n}\exp(-\lambda c_d d^{\alpha}\beta^{\delta}\Gamma(1+\delta)\Gamma(1-\delta))(CSI)$
9: **end if**

Algorithm 1: The flow diagram of power control algorithm based on outage probability awareness.

and Guangxi University Science Research Project (ZD 2014146).

References

[1] Y. Zhang, W.-Q. Xu, J.-M. Chen, and Y.-X. Sun, "Steepest descent method based transmission power control in vehicular networks," *Journal of Electronics and Information Technology*, vol. 32, no. 10, pp. 2536–2540, 2010.

[2] F. Cunha, L. Villas, A. Boukerche et al., "Data communication in VANETs: Protocols, applications and challenges," *Ad Hoc Networks*, vol. 44, pp. 90–103, 2016.

[3] G. Samara, T. Alhmiedat, and Salem. A. O. A., "Dynamic Safety Message Power Control in VANET Using PSO," *Computer Science*, vol. 3, 2014.

[4] Y. Wu, L. Shen, Z. Shao, Q. Su, and X. Lin, "Power control algorithm based on probe message in vehicular ad hoc network," *Journal of Southeast University (Natural Science Edition)*, vol. 41, no. 4, pp. 659–664, 2011.

[5] T. Hengliang, *Research on the Key Technology of Vehicular Ad Hoc Network in Urban Transportation Environment*, Beijing Jiaotong University, 2013.

[6] G. Jia, "Design and Implementation of Vehicular Ad Hoc Network Communication System," *University of Electronic Science and Technology of China*, 2012.

[7] A. Nabeel, S. C. Ergen, and O. Ozkasap, "Vehicle mobility and communication channel models for realistic and efficient highway vanet simulation," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 1, pp. 248–262, 2014.

[8] W.-B. Yu, D.-W. Niu, Z.-C. Mi, and C. Dong, "Research on fairness in vehicle networks based on power adjusting," *Journal of the University of Electronic Science and Technology of China*, vol. 40, no. 5, pp. 706–710, 2011.

[9] X. Guan, R. Sengupta, H. Krishnan, and F. Bai, "A feedback-based power control algorithm design for VANET," in *Proceedings of the 2007 Mobile Networking for Vehicular Environments (MOVE '07)*, pp. 67–72, May 2007.

[10] L. Cheng and R. Shakya, "VANET Adaptive power control from realistic propagation and traffic modeling," in *Proceedings of the 2010 IEEE Radio and Wireless Symposium (RWW '10)*, pp. 665–668, January 2010.

[11] C. Shuqun and L. Xiaohua, "Power control method based on network condition information for VANET," *Transducer and Microsystem Technologies*, vol. 36, no. 11, pp. 44–46, 2017.

[12] X. Zhexin, L. Shijie, L. Xiao, and Y. Wu, "Power control mechanism for vehicle status message in VANET," *Journal of Computer Applications*, vol. 36, no. 8, pp. 2175–2180, 2016.

[13] S. Sou, "A Power-Saving Model for Roadside Unit Deployment in Vehicular Networks," *IEEE Communications Letters*, vol. 140, no. 7, pp. 623–625, 2010.

[14] Y.-F. Mo, D.-X. Yu, S.-N. Bao, and S.-T. Gao, "Beacon Transmission Power Control Algorithm Based on the Preset Threshold in VANETs," *Dongbei Daxue Xuebao*, vol. 38, no. 3, pp. 331–334, 2017.

[15] X. Ruifeng, F. Ming, and T. Yulong, "Routing Optimization Scheme Based on L in k Reliability in Vehicular Ad HOC Network," *Computer Engineering*, vol. 42, no. 3, pp. 13–17, 2016.

[16] F. Qu, Z. Wu, F.-Y. Wang, and W. Cho, "A security and privacy review of VANETs," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 6, pp. 2985–2996, 2015.

[17] B. L. Arkin and L. M. Leemis, "Nonparametric estimation of the cumulative intensity function for a nonhomogeneous Poisson process from overlapping realizations," *Management Science*, vol. 46, no. 7, pp. 989–998, 2000.

[18] J.-F. Coeurjolly, J. Møller, and R. Waagepetersen, "A Tutorial on Palm Distributions for Spatial Point Processes," *International Statistical Review*, vol. 85, no. 3, pp. 404–420, 2017.

[19] M. Schlather, "On a class of models of stochastic geometry constructed by random measures," *Mathematische Nachrichten*, vol. 213, pp. 141–154, 2000.

[20] C. Campolo, C. Sommer, F. Dressler, and A. Molinaro, "On the impact of adjacent channel interference in multi-channel VANETs," in *Proceedings of the 2016 IEEE International Conference on Communications, ICC 2016*, pp. 1–7, Malaysia, May 2016.

[21] C. Jiang, H. Zhang, Z. Han, Y. Ren, V. C. M. Leung, and L. Hanzo, "Information-Sharing Outage-Probability Analysis of Vehicular Networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 12, pp. 9479–9492, 2016.