

Article

# Grassmann Manifold Based State Analysis Method of Traffic Surveillance Video

Peng Qin, Yong Zhang, Boyue Wang \* and Yongli Hu 

Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China; pqin@emails.bjut.edu.cn (P.Q.); zhangyong2010@bjut.edu.cn (Y.Z.); huyongli@bjut.edu.cn (Y.H.)

\* Correspondence: wby@bjut.edu.cn; Tel.: +86-10-6739-6568 (ext. 2103)

Received: 20 February 2019; Accepted: 19 March 2019; Published: 29 March 2019



**Featured Application:** Based on Grassmann Manifold, a method for identifying congestion state of Traffic Surveillance Video has been proposed in this paper. Meanwhile, the effectiveness of the proposed method is verified via UCSD dataset and the Capital Airport Expressway dataset.

**Abstract:** For a contemporary intelligent transport system, congestion state analysis of traffic surveillance video (TSV) is one of the most crucial and intricate research topics because of the rapid development of transportation systems, the sustained growth of surveillance facilities on road, which lead to massive traffic flow data, and the inherent characteristics of our analysis target. Traditional methods on feature extractions are usually operated on Euclidean space in general, which are not accurate for high-dimensional TSV data analysis. This paper proposes a Grassmann manifold based neural network model to analysis TSV data, by mapping the video data from high dimensional Euclidean space to Grassmann manifold space, and considering the inner relation among adjacent cameras. The accuracy of the traffic congestion is improved, compared with several traditional methods. Experimental results are conducted to validate the accuracy of our method and to investigate the effects of different factors on performance.

**Keywords:** traffic surveillance video; state analysis; Grassmann manifold; neural network

## 1. Introduction

At present, state analysis of urban transportation is generally admitted as a progressively intricate issue due to the inherent complexity. Primarily, urban transport system is considered intricate and massive. Thus the corresponding data of traffic flow own its significant characteristics, which are distributed, stochastic and spatiotemporal correlated. Furthermore, the quantity of deployed surveillance cameras are increasing in leaps and bounds globally. Under this circumstance, much more accurate state analysis of traffic surveillance video (TSV) is the vital problem demanding a prompt solution. Since the development of artificial intelligence and pattern recognition has risen steadily, the internal law of urban transportation, such as state analysis and estimate, is able to be digged out to a greater extent than before [1]. Accordingly, the intelligent transport system can be much more efficient, more collaborative and more predictive.

TSV data can be used to extract massive amounts of meaningful traffic information including traffic density, average velocity of vehicles, traffic flux, lane-changing rate and other various statistical properties for traffic flow. In fact, there are some pioneer studies [2,3] that obtain those properties from traffic videos, which can be applied to validate mathematical models of traffic flow.

Currently, traditional traffic state classifications are roughly divided into three categories: “heavy”, “medium” and “light” in general, which however, need higher classification accuracy to meet the requirement of acceptable performance of an intelligent transport system. There are several

traditional methodologies of the traffic state classifications, such as Sparse Representation (SR), Support Vector Machine (SVM), Neural Networks (NN) and so on.

On one hand, SR is based on the principle that a signal can often be represented by a linear sum of a small amount of signals from dictionary. Numerous research has been made to construct the dictionary with specific attributes. Sparse representation model achieves great success in many applications, such as face recognition [4], image denoising [5], and image super-resolution reconstruction [6]. On the other hand, the main idea of SVM can be summarized into two points: (1) It is analyzed for linear separable case. In the case of linear inseparable, in order to make this case linear separable, linear inseparable low-dimensional input space is mapped to high-dimensional feature space through nonlinear mapping algorithm. (2) SVM creates the optimal segmentation hyperplane in the feature space based on the Structural Risk Minimization Theory, so that the learners get global optimization, and the expected risk of the entire sample space can meet a certain upper bound with a certain probability. Compared with the above methods, NN can reflect the relevancy of the TSV data more accurately, so NN is better than the other methods in the classification of traffic TSV.

The traditional methods of feature extraction, such as PCA, LDA and LPP, process original data in Euclidean space. However, TSV data is high-dimensional which is operated inaccurately in Euclidean space, thus the result of feature extraction in the Euclidean space is not effective. Scholars demonstrate that high-dimensional Euclidean space can be embedded in manifold space, so if we map the data from high dimensional Euclidean space to manifold space, the operation will be more accurate and the feature extraction will be more effective. At the same time, TSV is difficult to make a consistent length, and it is important to process different lengths of TSV data uniformly. In the manifold, there is a kind named Grassmann manifold, which can process different lengths of TSV data uniformly by unifying data while doing SVD decomposition.

In this paper we propose a neural network model based on Grassmann manifold. Our model can directly identify the congestion status without calculating traffic flow parameters such as vehicle speed and number of vehicles. The data processing schematic is shown in Figure 1. Furthermore, the contributions of the paper are as follows:

- Introducing neural networks model into traffic state classification applications.
- Connecting Grassmann manifold with neural networks according to the temporal characteristics of traffic data.
- Giving a reliable and efficient algorithm to neural networks model based on Grassmann manifold.

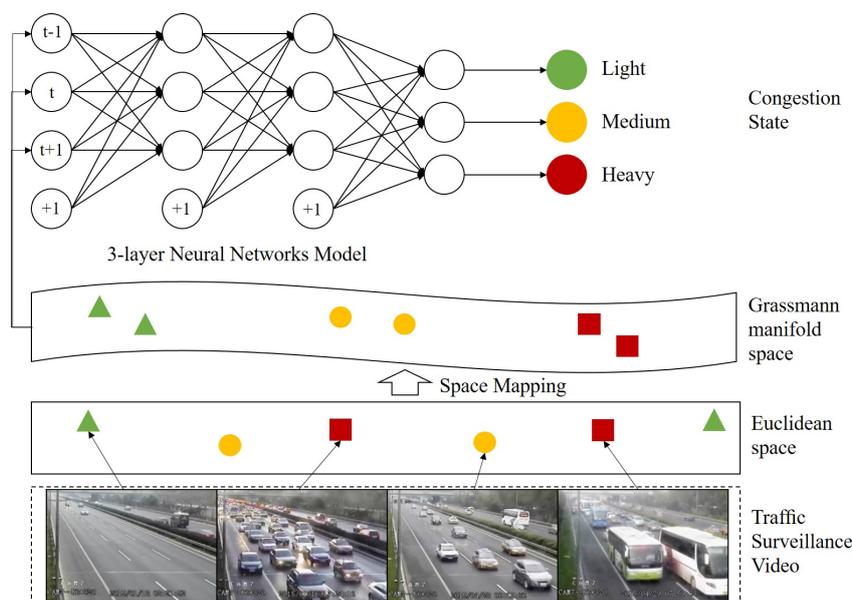


Figure 1. The data processing schematic.

The rest of the paper is organized as follows. In Section 2, we review the background of neural networks and Grassmann manifold. In Section 3, we introduce the principles and application of Grassmann based neural network method in the traffic state analysis. In Section 4, we validate the performance of our traffic state analysis method on two datasets, the UCSD dataset and Capital Airport Expressway dataset. Finally, we conclude the paper and come up with other research directions in Section 5.

## 2. Preliminaries

### 2.1. Grassmann Manifold

Recently, some discriminant analyses on Grassmann manifold [7] have been achieved by researchers. The basic principle of the Grassmann manifold algorithm is compressing datasets that are loose distributed on Grassmann Manifold into a low-dimensional Grassmann manifold through non-linear transformation, so that the distribution of the dataset is more compact on the low-dimensional Grassmann manifold. Hamm proposed Grassmann manifold Discriminant Analysis (GDA) [8], in which the subspace is represented by points on Grassmann manifold, to achieve nonlinear discriminant analysis on Grassmann manifold by measuring the similarity of subspace using Grassmann kernel. Although GDA performs superior properties, it does not explicitly consider the intrinsic geometry of data [9–11], which affects the ability of discriminant for the dataset and the generalization. In view of this, Harandi [12] proposed a Graph Embedding Discriminant Analysis on Grassmannian Manifolds (GEDAGM). In the case where global solutions do not work, the local learning method provides an effective solution.

Some fatal concepts on Grassmann manifolds of Riemannian geometry are supposed to be illustrated before presenting our algorithm. Details on Grassmann manifolds and related topics can be found in [13].

Grassmann manifold  $G(p, d)$  [13] consists of the set of all linear  $p$ -dimensional subspaces of  $\mathbb{R}^d$  ( $0 < p < d$ ), which stands on a space of  $p \times d$  matrices with  $d$  orthogonal columns:

$$G(p, d) = \{X \in \mathbb{R}^{d \times p} : X^T X = I_p\} \quad (1)$$

There are two methods to measure the distance in Grassmann manifold. One is to map the Grassmann points into tangent spaces where there exist measures [14,15]. Another method is to embed the Grassmann manifold into the symmetric matrix spaces while the Euclidean distance is available. The latter one is easier and more effective in practice, and the mapping relation can be represented as [16]:

$$\Pi : G(p, d) \rightarrow \text{Sym}(d), \Pi(X_i) = X X^T \quad (2)$$

The embedding  $\Pi(X)$  is diffeomorphism [17]. In this paper, we adopt the second strategy on Grassmann manifold to define the following distance inherited from the symmetric matrix space under the (2) mapping:

$$d_g^2(X, Y) = 1/2 \|\Pi(X) - \Pi(Y)\|_F^2 \quad (3)$$

One point on Grassmann manifold is actually an equivalent class of all orthogonal high matrices in  $\mathbb{R}^{d \times p}$ , any one of which can be converted to the other by a  $p \times p$  orthogonal matrix.

### 2.2. Neural Networks

Once the features are extracted separately from both the visible and IR images, they are combined together and fed to a neural network classifier. In order to accommodate temporal changes, the neural network should be fast in learning and be plastic to accommodate changes in the data. The unsupervised Kohonen Self Organizing Feature Map (SOM) [18] and the supervised Probability neural network (PNN) [19] are chosen and their effectiveness are examined for this problem.

On one hand, Kohonen SOM is one of the most popular unsupervised networks which exploits the natural structure of the input feature space without using any priori class information. This method is suitable for the cloud classification tasks where ground truth is often not available or reliable and a huge amount of satellite data is available. The training process of the SOM is based on the competitive learning rule. When a new input is present, the neurons will compete with each other by comparing the distance of the input and the weight vector of that neuron. For the winning neuron, winner-takes-all weight adjustment rule is applied. The neurons, which are close enough to the winning neuron in a neighborhood, will also participate in learning. As the learning progresses, the neighborhood shrinks to declare only one winning neuron. In this way, the SOM represents the feature space by a set of well organized neurons which corresponds to the centroid of each cluster. These neurons later are mapped into physical classes.

On the other hand, PNN is a kind of supervised network which is closely related to Bayes classification rule and Parzen non-parametric probability density function (PDF) estimation. Comparing with the well-known back-propagation (BP) network, PNN has a very fast one-pass learning scheme while it has a comparable generalization ability [19].

### 2.3. Single Neuron Involved Neural Networks

For training sample set  $(x^i, y^i)$ , neural network algorithm can provide a complex and non-linear assumption model  $h_{W,b}(x)$ , in which  $W$  and  $b$  are used to fit the data.

Neurons are the basis elements of the neural network, the most simple neural network mode contains only one neuron. For the neuron network which only contains neurons, the input is  $x_1, x_2, x_3$  and the intercept is +1. This Neural network operates as follows:

$$h_{W,b}(x) = f(W^T x) = f\left(\sum_{i=1}^3 W_i x_i + b\right) \quad (4)$$

where  $W$  is the connection parameter (the weight of the cable), and  $b$  is the offset term. The function  $f$  is activation function. It can be seen that a single neuron input - output mapping relationship is actually a logistic regression.

## 3. The Proposed Method

TSV data feature extraction and spatial mapping process are shown in Figure 1. When processing the raw data of TSV, the feature extraction in the European space will affect the effect because of the inaccurate operation, so the features extracted in the European space are irregularly distributed, which is not conducive to the subsequent processing. In order to solve the problem of inaccurate calculation of data in European space, we map the data of European space to the Grassmann manifold space. Since the operation is more accurate in the Grassmann manifold space, the extracted feature distribution is very regular.

After the feature extraction and spatial mapping processing based on Grassmann Manifold, in order to express the spatio-temporal correlation of traffic information, we input the features in the Grassmann manifold space into the neural network. Through the three-layer neural network, the extracted features are classified according to three states: "light", "medium" and "heavy". Finally, the traffic status is obtained, as shown in Figure 1.

Neural networks couple many single neurons together, so that an output is inputted to another neuron. The sketch map of the neural network architecture is shown in Figure 2.

The leftmost layer in neural network is called the input layer, the rightmost one is called the output layer, and the superscript +1 circle is called the bias node, which is intercept. The network layer between the input layer and the output layer is called the hidden layer, since we do not concentrate on the values of the middle layer when training sample observations. At the same time, we can see that the above neural network has three input units (not counting the bias unit), two hidden layers (each

with three units) and three output units. In our experiments, the input layer is traffic video feature matrix which is extracted by Grassmann manifold, and then through the appropriate training and classification rules, traffic state eventually is divided into “heavy”, “medium” and “light”. These traffic states mean that the neural network has three outputs.

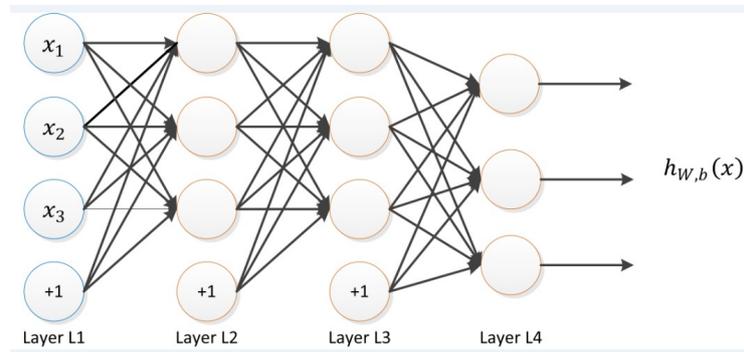


Figure 2. The sketch map of the neural network architecture.

We use  $n_l$  to represent the layers of the network. The  $l$ th layer is recorded as  $L_l$ , and the input layer is  $L_1$  and the output layer is  $L_{n_l}$ . There is  $(W, b) = (W^{(1)}, b^{(1)}, W^{(2)}, b^{(2)})$  as parameters in the above neural network,  $W_{ij}^{(l)}$  is the connection parameter between unit  $j$  in layer  $l$  and unit  $i$  in layer  $l + 1$ .  $b_i^{(l)}$  is offset term of unit  $i$  in layer  $l + 1$ .  $s_l$  is used to represent the number of nodes in layer  $l$  (not counting the bias unit).

$a_i^{(l)}$  is used to represent the activation value (output value) of unit  $i$  in layer  $l$ .  $a_i^{(1)} = x_i$  while  $l = 1$ , means the  $i$ th input value (the  $i$ th feature of the input value). For a given set of parameters  $W, b$ , we can calculate the output according to  $h_{W,b}(x)$  through neural network. Neural network computation steps in Figure 2 are obtained as follows.

$$a_1^{(2)} = f(W_{1,1}^{(1)}x_1 + W_{1,2}^{(1)}x_2 + W_{1,3}^{(1)}x_3 + b_1^{(1)}) \tag{5}$$

$$a_2^{(2)} = f(W_{2,1}^{(1)}x_1 + W_{2,2}^{(1)}x_2 + W_{2,3}^{(1)}x_3 + b_2^{(1)}) \tag{6}$$

$$a_3^{(2)} = f(W_{3,1}^{(1)}x_1 + W_{3,2}^{(1)}x_2 + W_{3,3}^{(1)}x_3 + b_3^{(1)}) \tag{7}$$

$$h_{W,b} = a_1^{(3)} = f(W_{1,1}^{(2)}x_1 + W_{1,2}^{(2)}x_2 + W_{1,3}^{(2)}x_3 + b_1^{(2)}) \tag{8}$$

$z_i^{(l)}$  is used to represent the weighted sum of input for unit  $i$  in layer  $l$  (counting the bias unit).

$$z_i^l = \sum_{j=1}^n W_{ij}^{(l-1)}x_j + b_i^{(l-1)} \tag{9}$$

So

$$a_i^{(l)} = f(z_i^{(l)}) \tag{10}$$

Then we expand the activation function  $f(\cdot)$  to vectors, which means  $f([z_1, z_2, z_3]) = [f(z_1), f(z_2), f(z_3)]$ . Therefore, the above equations can be more succinctly expressed as follows.

$$z^{(2)} = W^{(1)}x + b^{(1)} \tag{11}$$

$$a^{(2)} = f(z^{(2)}) \tag{12}$$

$$z^{(3)} = W^{(2)}a^{(2)} + b^{(2)} \tag{13}$$

$$h_{W,b}(x) = a^{(3)} = f(z^{(3)}) \tag{14}$$

The above calculation is called propagation step forward. Given the activation value  $a^{(l)}$  in layer  $l$ , the activation value  $a^{(l+1)}$  in layer  $l + 1$  can be calculated as :

$$z^{(l+1)} = W^{(l)}a^{(l)} + b^{(l)} \quad (15)$$

$$a^{(l+1)} = f(z^{(l+1)}) \quad (16)$$

We quickly solve the neural network problems with the help of linear algebra by using a matrix-vector operating method.

In our experiment, we use two methods to input Grassmann manifold space features into the neural network. First, in order to be able to show the temporal correlation of traffic video data features, we input the characteristics of  $t - 1$ ,  $t$  and  $t + 1$  moments under the same camera into the neural network simultaneously, so that the results are classified accurately by front-to-back video frame constraints. In this case,  $t$  in Figure 1 represents time. Secondly, in order to be able to express the spatial correlation of traffic video data features, we input the feature of camera  $t$  and the features of the cameras  $t - 1$  and  $t + 1$ , which are connected to the camera  $t$ , into the neural network simultaneously, so that the results are classified accurately by front and rear camera constraints. In this case,  $t$  in Figure 1 represents camera number.

#### 4. Experiment Results and Discussion

In this section, to test the effectiveness of our proposed method, we conduct several unsupervised experiments for TSV clips classification. The two datasets used in our experiments are listed below:

- UCSD dataset (<http://www.svcl.ucsd.edu/projects/traffic/>).
- Capital Airport Expressway dataset (We establish the dataset by ourselves).

To demonstrate the performance of Grassmann Neural Networks (GNN) method, we compare GNN method with several state-of-the-art clustering methods. Since our method is related to Grassmann Sparse Representation (GSR) models, we select GSR based methods as baselines, which are listed below:

GDA: A transform over the Grassmann manifold is learned to simultaneously maximize interclass distance and minimize intra-class distance in GDA.

Graph-embedding Grassmann Discriminant Analysis (GGDA): It is considered as an extension of GDA, where a local discriminant transform over Grassmann manifolds is learned. This is achieved by incorporating local similarities/dissimilarities through within-class and between-class similarity graphs.

Kernel Affine Hull method (KAHM): Images are considered as points in a linear or affine feature space, while image sets are characterized by a convex geometric region (affine or convex hull) spanned by their feature points in KAHM.

Our proposed methods are also compared against Linear Dynamical System (LDS), Compressive Sensing Linear Dynamical System (CS-LDS) and Grassmann Sparse Representation (GSR).

In our experiments, the performances of different algorithms are measured by the following clustering accuracy  $Accuracy = \frac{M}{N} \times 100\%$ , where  $M$  means the number of correctly classified points, and  $N$  means the total number of points. All algorithms are run in matlab 2014a environment, and workstation is configured for Intel Core i7-4770K 3.5GHz CPU and 16G RAM. We first introduce the database and experimental set-up, then report and analyze our results.

##### 4.1. Information of Datasets

(1) UCSD Traffic Dataset: The dataset includes 254 segments of highway video data, with different statuses in different weather conditions, such as cloudy, sunny, rainy days and so on. Each video has a resolution of  $320 \times 240$  pixels and 42 to 52 frames. We unify the resolution into a  $48 \times 48$  standardized grayscale when we use the dataset. The standardized operation of the video clip includes subtracting

average image and standardizing pixel intensity by the method of unit variance. This standardized operation has a better effect for reducing the impact of changes in illumination.

The dataset is divided into three levels according to the degree of traffic congestion, which are heavy, medium and light. There are 44 video segments in heavy state, 45 video segments in medium state, and 165 video segments in light state. Some examples of the video segments are shown in Figure 3.



**Figure 3.** Examples of UCSD dataset.

(2) Capital Airport Expressway Dataset: The dataset is established by ourselves. We first select suitable videos from a large number of TSVs of Beijing Capital Airport Expressway. After the selection of record clear, situation mutative, all the required status contained videos, we convert videos from .ts format into .avi format. Then we choose videos from the same camera at different time and intercept each video segment into 5 s. Then we save these video segments as original data matrices by specific algorithm, forming our own dataset. After extracting the Grassmann manifold characteristics, the dataset is used in traffic status classification based on sparse representation and neural network. We select the videos of Capital Airport Expressway by filtering videos from 19 surveillance video cameras. The map of airport expressway in Beijing and the 19 cameras on the expressway are shown in Figure 4.

After considering the clarity, the congestion level and other conditions of the video, we select TSVs in Wuyuanqiao North, which is number 6 on the map. Then we intercept and select the large number of TSVs of this camera, in order to form a relative and representative video database. Some examples are shown in Figure 5.

For the selected TSVs, first, we screened a large number of surveillance videos provided by the Beijing Traffic Information Center. After selecting the surveillance videos that meet the requirements of clear recording, diversified conditions, and including all states, we chose videos from the same camera at different time and intercepted each video segment into 5 s. Then we saved these video segments into the original data matrix. Then we extract features in original data matrix and save these features as a new data matrix. There are five datasets in this part of the experiment, which are shown in Table 1.

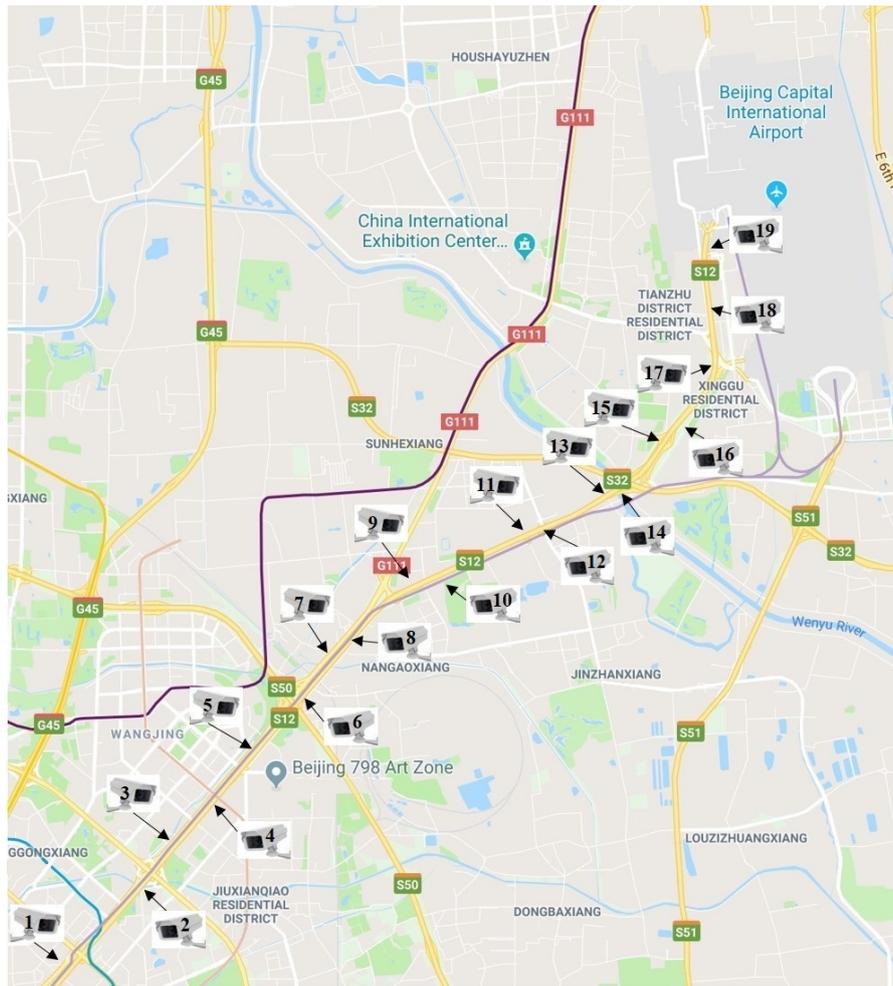


Figure 4. Map of Capital Airport Expressway in Beijing.



Figure 5. Surveillance video of Capital Airport Expressway.

**Table 1.** Information of the dataset.

Name	Number of Video	Including Daytime Video	Including Night Video
Data-video-original	120	yes	no
Data-video-expand	270	yes	no
Data-video-darklight	300	yes	yes
Data-video-darklight-only	300	no	yes
Data-video-mix	900	yes	yes

#### 4.2. Analysis of Traffic Video Status

(1) Analysis of the UCSD Dataset: We put each video into Grassmann manifold by using auto-regressive moving-average (ARMA) model. ARMA model is a common model which is used to describe stationary random sequence. This model includes three forms: (1) Auto-regressive (AR) model. (2) Moving-average (MA) model. (3) ARMA model. The observing size is set to 5 and the dimensions of the subspace are set to 10 in ARMA model. We compare our method with GDA, GGDA, KAHM, LDS, CS-LDS and GSR, the results shown in Table 2.

**Table 2.** Results of the experiment.

Methods	Accuracy (%)
GDA (Hammand Lee 2008)	79.1
GGDA (Harandi et al. 2011)	89.4
KAHM (Cevikalp and Triggs 2010)	82.7
LDS (Sankaranarayanan et al.2010)	87.8
CS-LDS (Sankaranarayanan et al. 2010)	86.6
GSR (Grassmann Sparse Reprasetation)	82.8
GNN (Grassmann Neural Networks)	99.5

The videos in the dataset are from the same camera, and the dataset contains different weather and light conditions. Through Table 2, we can see that the accuracy of our method (GNN) is pretty good, and the method gets the highest accuracy.

(2) Analysis of Capital Airport Expressway Dataset: After preparing the dataset and thinking through the experimental design work, we begin to conduct the experiment. First of all, we divide the traffic states into three levels, “heavy”, “medium” and “light”. Then we extract features by Grassmnn manifold and combine these features with sparse representation and neural network. Finally, we classify the dataset by using methods of machine learning, where 20 percent of the data are selected for training and 80 percent of the data are selected for testing. To verify the effect of our proposed method in a variety of situations, this Expressway Dataset is divided into five datasets, and each dataset contains different numbers of videos and different conditions.

We select two datasets named Data-video-original and Data-video-expand, which respectively include 120 and 270 videos, and both of the datasets are just videos in daytime. The traffic video status classification accuracy of these two datasets is shown in Table 3.

**Table 3.** The accuracy of these two datasets.

Dataset	Data-Video-Original (%)	Data-Video-Expand (%)
GGDA	45.24	82.54
GSR	95.80	99.50
GNN	98.96	99.54

After taking into consideration that the actual situation is not only daytime, but also contains a lot of night time (bad light conditions), we expand the datasets with Data-video-darklight dataset,

Data-video-darklight-only dataset and Data-video-mix dataset. The traffic video status classification accuracy of the three datasets is shown in Table 4.

**Table 4.** The accuracy of these three datasets.

Dataset	Data-Video-Darklight (%)	Data-Video-Darklight-Only (%)	Data-Video-Mix (%)
GGDA	40.48	34.29	48.73
GSR	97.08	97.08	99.30
GNN	99.58	99.58	99.86

There are both daytime and nighttime TSVs in the dataset, and the use of headlight is inconsistent, so the light conditions are complicated. On the other hand, obviously, the weather is not always the same, ranging from sunny, cloudy, foggy and so on. All the reasons above show that it is especially difficult to get high accuracy using this dataset. Although there are many difficulties, considering the inner relation through the front-neighbor camera and behind-neighbor camera, our method (GNN) achieves the desired effect.

According to Tables 3 and 4, there is no doubt that our methodology acquired higher classification accuracy among all the methods.

In the experiment, we used the UCSD dataset and built Capital Airport Expressway dataset. The experiment proves that our method can be applied to analyzing congestion state of traffic surveillance video data, and our method is simple and effective.

## 5. Conclusions

In this paper, a neural network model based on Grassmann manifold has been proposed to analyze TSV statuses. Our algorithm can directly identify the congestion status without calculating traffic flow parameters such as vehicle speed and number of vehicles. We map the data of European space to the Grassmann manifold space, to solve the problem of inaccurate calculation of high-dimensional data in European space. Then we use the video data of the adjacent camera to improve the accuracy of traffic status analysis under the current camera. The Capital Airport Expressway dataset was established to facilitate our experiments. Experimental results show that the proposed method is superior compared to other state-of-the-art traffic data analysis methodologies. For future work, the traffic surveillance video database is expected to be an expansion on various monitory spots.

**Author Contributions:** Conceptualization, P.Q., Y.Z. and Y.H.; Data curation, P.Q.; Formal analysis, P.Q. and B.W.; Funding acquisition, Y.Z.; Investigation, P.Q. and B.W.; Methodology, P.Q., Y.Z., B.W. and Y.H.; Project administration, Y.Z.; Resources, Y.Z. and Y.H.; Software, P.Q. and B.W.; Supervision, Y.Z. and Y.H.; Validation, P.Q.; Visualization, P.Q.; Writing—original draft, P.Q. and B.W.; Writing—review & editing, Y.Z. and Y.H.

**Funding:** This work was supported in part by the National Natural Science Foundation of China under Grant U1811463, 4162009, 61602486, 61632006, and 61772049, in part by the Beijing Municipal Science and Technology Project under Grant Z171100004417023, in part by the Beijing Municipal Education Project under Grant KM201610005033.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Zhang, H.; Zhang, Y.; Hu, D. Study on the architecture and methods for large amount of data management. *Comput. Eng. Appl.* **2004**, *11*, 26–131.
2. Fukuda, E.; Tanimoto, J.; Iwamura, Y.; Nakamura, K.; Akimoto, M. Field measurement analysis to validate lane-changing behavior in a cellular automaton model. *Phys. Rev. E* **2016**, *94*, 052209. [[CrossRef](#)] [[PubMed](#)]
3. Tanimoto, J.; Fujiki, T.; Kukida, S.; Ikegaya, N.; Hagishima, A. Acquisition of the field measurement data relating to lane change actions. *Int. J. Mod. Phys. C* **2015**, *26*, 1550072. [[CrossRef](#)]

4. Wright, J.; Yang, A.; Ganesh, A. Robust face recognition via sparse representation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2009**, *31*, 210–227. [[CrossRef](#)] [[PubMed](#)]
5. Elad, M.; Aharon, M. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Trans. Image Process.* **2006**, *15*, 3736–3745. [[CrossRef](#)] [[PubMed](#)]
6. Mairal, J.; Elad, M.; Sapiro, G. Sparse representation for color image restoration. *IEEE Trans. Image Process.* **2008**, *17*, 53–69. [[CrossRef](#)] [[PubMed](#)]
7. Shirazi, S.; Harandi, M.; Sanderson, C. Clustering on grassmann manifolds via kernel embedding with application to action analysis. In Proceedings of the IEEE International Conference on Image Processing, Orlando, FL, USA, 30 September–3 October 2012.
8. Hamm, J.; Lee, D. Grassmann discriminant analysis: A unifying view on subspace-based learning. In Proceedings of the International Conference on Machine Learning, Helsinki, Finland, 5–9 July 2008; pp. 376–383.
9. Chen, J.; Ye, J.; Li, Q. Integrating global and local structures: A least squares framework for dimensionality reduction. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, USA, 17–22 June 2007; pp. 1–8.
10. Yin, X.; Hu, E. Semi-supervised locality dimensionality reduction. *J. Image Graph.* **2011**, *16*, 1615–1624.
11. Harandi, M.; Ahmadabadi, M.N.; Araabi, B.N. Optimal local basis: A reinforcement learning approach for face recognition. *Int. J. Comput. Vis.* **2009**, *81*, 191–204. [[CrossRef](#)]
12. Harandi, M.; Sanderson, C.; Shirazi, S. Graph embedding discriminant analysis on grassmannian manifolds for improved image set matching. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Colorado Springs, CO, USA, 20–25 June 2011; pp. 2705–2712.
13. Absil, P.; Mahony, R.; Sepulchre, R. *Optimization Algorithms on Matrix Manifolds*; Princeton University Press: Princeton, NJ, USA, 2008.
14. Cetingul, H.; Wright, M.; Thompson, P.; Vidal, R. Segmentation of high angular resolution diffusion mri using sparse riemannian manifold clustering. *IEEE Trans. Med. Imaging* **2014**, *33*, 301–317. [[CrossRef](#)] [[PubMed](#)]
15. Goh, A.; Vidal, R. Clustering and dimensionality reduction on riemannian manifolds. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 23–28 June 2008; pp. 1–7.
16. Harandi, M.; Sanderson, C.; Shen, C.; Lovell, B. Dictionary learning and sparse coding on grassmann manifolds: An extrinsic solution. In Proceedings of the International Conference on Computer Vision, Sydney, Australia, 1–8 December 2013; pp. 3120–3127.
17. Helmke, J.; Huper, K. Newton's method on grassmann manifolds. *arXiv* **2007**, arXiv:0709.2205.
18. Kohonen, M.T. The self-organizing map. *Proc. IEEE* **1990**, *78*, 1464–1480. [[CrossRef](#)]
19. Specht, D.F. Probabilistic neural network. *Neural Netw.* **1990**, *3*, 109–118. [[CrossRef](#)]



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).