International Journal of Geo-Information

MDPI

Article

Traffic Command Gesture Recognition for Virtual Urban Scenes Based on a Spatiotemporal Convolution Neural Network

Chunyong Ma [1,2], Yu Zhang [1,*], Anni Wang [1], Yuan Wang [1] and Ge Chen [1,2]

[1] Marine Information Technology Laboratory (Ocean University of China), Ministry of Education, Qingdao 266100, China; chunyongma@ouc.edu.cn (C.M.); wanganni@stu.ouc.edu.cn (A.W.); wangyuan@stu.ouc.edu.cn (Y.W.); gechen@ouc.edu.cn (G.C.)

[2] Laboratory for Regional Oceanography and Numerical Modeling, Qingdao National Laboratory for Marine Science and Technology, Qingdao 266100, China

* Correspondence: zhangyu2599@stu.ouc.edu.cn

Received: 11 November 2017; Accepted: 16 January 2018; Published: 22 January 2018

Abstract: Intelligent recognition of traffic police command gestures increases authenticity and interactivity in virtual urban scenes. To actualize real-time traffic gesture recognition, a novel spatiotemporal convolution neural network (ST-CNN) model is presented. We utilized Kinect 2.0 to construct a traffic police command gesture skeleton (TPCGS) dataset collected from 10 volunteers. Subsequently, convolution operations on the locational change of each skeletal point were performed to extract temporal features, analyze the relative positions of skeletal points, and extract spatial features. After temporal and spatial features based on the three-dimensional positional information of traffic police skeleton points were extracted, the ST-CNN model classified positional information into eight types of Chinese traffic police gestures. The test accuracy of the ST-CNN model was 96.67%. In addition, a virtual urban traffic scene in which real-time command tests were carried out was set up, and a real-time test accuracy rate of 93.0% was achieved. The proposed ST-CNN model ensured a high level of accuracy and robustness. The ST-CNN model recognized traffic command gestures, and such recognition was found to control vehicles in virtual traffic environments, which enriches the interactive mode of the virtual city scene. Traffic command gesture recognition contributes to smart city construction.

Keywords: traffic command gesture recognition; VGE; spatiotemporal CNN; HCI

1. Introduction

People now have a strong dependence on traffic, and requirements with respect to such traffic have recently been put forward. The concept of smart traffic aids in governmental decision-making and management and reduces traffic accidents [1]. In traffic systems, traffic command gestures help to alleviate traffic jams. The virtual traffic command gestures experience system proposed in this paper is helpful. The intelligent recognition of traffic command gestures can promote traffic safety awareness. Users experience the traffic police command process in the virtual environment. When users actually walk or drive on roads, they are able to identify traffic police's actions accurately, so as to prevent traffic accidents.

An intelligent traffic command gesture recognition system cannot work without a virtual geographic environment (VGE). VGEs provide open virtual environments that correspond to the real world so as to assist in computer-aided geographic experiments. Four subenvironments include (1) the data environment, (2) the modeling and simulation environment, (3) the interactive environment, and (4) the collaborative environment [2]. At present, people pay more attention to

the first two environments, with a great deal of in-model building, scene loading, and numerical simulation for urban traffic simulation. Song et al. [3] proposed a graphics processing unit (GPU)-based mesoscopic simulation framework to handle large-scale dynamic traffic assignment problems. In this study, a 3D interactive environment is the focus. This interactive environment is designed to provide interactive channels between users and the VGE to facilitate the convenient participation of public users and to convey a sense of satisfaction [2]. There have been studies in this area. Yang et al. [4] presented a method to reflect the rapidly changing behaviors of the traffic flow simulation process and inserted virtual vehicles into real data. To better meet the urban scene objectives of traffic-managing systems, virtual reality human–computer interaction (HCI) techniques are applied. With the development of HCI technology, more natural interactive products are in demand. Gestures, being interactive, deliver more natural, creative, and intuitive methods of communicating with our computers [5]. The traffic police gesture recognition proposed in this paper adds interactivity to virtual urban scenes and makes them more user-friendly.

Human action recognition is fundamental in traffic gesture estimation. The study of traffic police action recognition is mainly divided into two categories: accelerator-based and vision-sensor-based. Wang et al. [6] used two three-axis accelerometers to provide arm movements and hand positions of the gravity vector signal and designed a hierarchical classifier to identify traffic police actions. Accelerator-based hand gesture recognition has high data stability and less noise, but the user is required to carry the equipment. Le et al. [7] proposed a real-time traffic gesture recognition test platform system based on support vector machine (SVM) training data. In their method, traffic police command gestures are captured in the form of a depth image based on a visual sensor, and the human skeleton is then constructed by a kinematic model. Vision-based gesture recognition is more suitable for stationary applications and often requires a specific camera set-up and calibration [8].

Research on traffic police action recognition, an important method of human–computer interaction, has made progress; however, it is difficult to accurately recognize actions because of complex backgrounds, occlusions, viewpoint variations, etc. [9]. Target tracking and motion recognition technology based on deep learning have developed at an unprecedented rate. Human action recognition based on video streaming has also been improved and updated [10–12]. Deep learning simulates the operational mechanisms of a human brain, extracts features, and exhibits efficient and accurate classification, detection, and segmentation. The main goal of this study was to improve the traditional interactive mode of VGEs. The main algorithm innovation is a novel spatiotemporal convolution neural network model in order to recognize traffic command gestures. The main contributions of this paper include three parts: a traffic police command gesture skeleton (TPCGS) dataset, a spatiotemporal convolution neural network (ST-CNN) model, and a real-time interactive urban intersection scene.

This paper proposes a virtual police gesture command system that consists of two parts, a virtual geographic interactive environment and a gesture recognition algorithm. The gesture recognition algorithm is used to judge the user's action. The movement of the traffic police model and the manner in which traffic is run is controlled. In order to improve the accuracy and robustness of the gesture recognition algorithm, this paper provides the TPCGS dataset of Chinese traffic police command gestures, which was completed by 10 volunteers. The dataset records the consecutive frames' action trajectories of multiple skeletal points, which will be highlighted in Section 3. Furthermore, a novel ST-CNN algorithm is presented that investigates a different architecture based on spatial and temporal convolution kernels. The ST-CNN model is trained based on the TPCGS dataset, which contains six skeletal points' locational information. As has been noted, temporal features can be obtained by recording the skeleton position in consecutive frames, extracting spatial features by recording 3D skeleton locational information, and analyzing relative positions of multiple skeletal points. Based on this, eight standard traffic police command actions are efficiently recognized. Section 4 details the ST-CNN architecture. The ST-CNN model is applied to the intersection of a virtual urban

traffic scene. Section 5 mainly describes the experimental process. The virtual police gesture command system is shown in Figure 1.

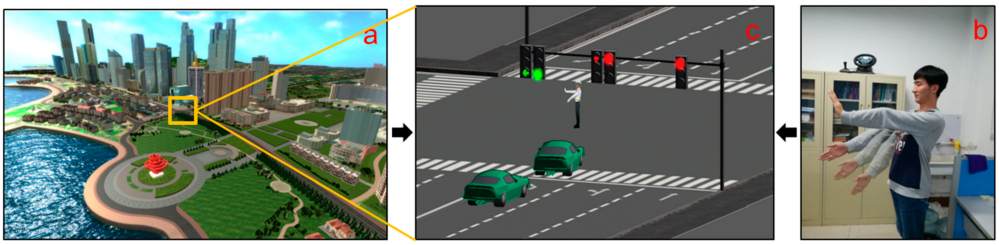

Figure 1. The virtual city traffic scene is constructed, and the intersection modeling is emphasized. (**a**) is a part of the virtual urban scene. The traffic police command gesture recognition system is set up to facilitate real-time human–computer interaction (HCI). Communication between the traffic scene and the identification system is achieved. A volunteer is making traffic police command gestures in the real environment in (**b**), which is mapped to the virtual scene in (**c**).

The key contributions of this study can be summarized as follows:

1. We built a virtual traffic interaction environment with virtual reality technology. Users can have interactions between their actions and the objects in the virtual traffic environment through a communication interface, experiencing and interacting with "real traffic crossroads."
2. We created a TPCGS dataset. The dataset uses depth trajectory data based on skeleton points. Compared with the video stream, the depth trajectory data features are more precise. The dataset provides a new means of identifying traffic police command gestures.
3. The ST-CNN model performs convolution operations on 3D position data well and has strong portability. A convolution kernel extracts temporal features of the skeleton point positional information from consecutive frames and extracts spatial features from the relationship between multiple skeleton points.

2. Related Work

Gestures are expressive body motions involving physical movements of the fingers, hands, arms, head, face, or body with the intent of (1) conveying meaningful information or (2) interacting with the environment [13]. Study purposes and objectives are different, and the postures of different parts of the body are identified. Based on depth images, Raheja et al. [14] focused on the detection of palm and fingertip positions. Liu et al. [15] focused on skin color detection and employed a K-nearest neighbor algorithm to recognize hand pose and obtain corresponding semantic information. Wang et al. [16] innovatively applied genetic algorithms to the recognition of finger and arm movement direction. The main research object of Wang et al. [17] was the body; they tracked the trajectory of the human body and achieved gait recognition. In order to construct a virtual traffic interaction environment, we wanted to simulate traffic police command gestures, so the motion trajectory of the human arm was determined as the main research object.

Feature extraction of motion trajectory is a key step in traffic police command gesture recognition. Feature extraction methods of action recognition include human geometric characteristics [18] and motion features [19]. After feature extraction, researchers usually use common pattern recognition algorithms, such as the hidden Markov model (HMM) and support vector machine (SVM), to classify them. Jie Yang et al. [20] developed a method to model actions using a hidden Markov model (HMM) representation. However, HMM is based on probability statistics. The HMM model's computational requirements for training the transition matrix and confusion matrix are too large to simulate complex actions. Schuldt et al. [21] constructed video representations in terms of local space–time features and integrated such representations with SVM classification schemes for recognition. Mathematical

models of human behavior recognition based on probability statistics are unable to practically simulate complex behavior. Deep learning provides new ideas for human behavior recognition, including convolution neural networks (CNNs) and recurrent neural networks.

Recently, several human activity recognition methods based on the CNN have been proposed. The CNN is a deep learning model in which trainable convolution filters extract picture features and neighborhood pooling operations to reduce the amount of data and avoid over-fitting, resulting in a hierarchy of increasingly complex features. It has been shown that, when trained with appropriate regularization, a CNN can achieve superior performance on picture recognition tasks than most machine learning methods. In addition, CNN has been shown to be invariant to certain variations, such as angle, lighting, and surrounding clutter [22]. Jiang et al. [23] proposed a deep convolution neural network (DCNN) method to learn the picture features of the signal sequences of accelerometers and gyroscopes to achieve human activity recognition. The DCNN model showed a performance of 97.59%, 97.83%, and 99.93% using standard UCI, USC, and SHO datasets, respectively. Yang et al. [24] achieved human activity recognition by extracting convolution features from multichannel time series data. Their CNN model, compared with a SVM, showed improved performance on the Opportunity Human Activity Recognition Challenge and other benchmark datasets. Ronao et al. [25] utilized a 1D convolution neural network to separately recognize a six-axis accelerometer and gyroscope triaxial sensor data. The model could achieve a 94.79% accuracy based on an activity dataset provided by 30 volunteers. Afterwards, they added fast Fourier transformation to the model, and the accuracy increased to 95.75%. Lee et al. [26] proposed a one-dimensional CNN to recognize human activities that included walking, running, and staying still. The accuracy rate of the 1D CNN-based method was as high as 92.71%, which is superior to the that of the random forest algorithm. These series of 1D CNN models provided us with a new idea; thus, our traffic police command gesture recognition solution is proposed here.

3. Virtual Urban Traffic Environment

The data of an actual three-dimensional urban space framework have increasingly become an important foundation of urban construction and development [27]. The virtual urban traffic scene in this study was built based on the geographical data of Qingdao, as shown in Figure 2. We visualized the terrain data and the surface model data to build a virtual traffic geographic environment, especially detailed with respect to traffic crossings and traffic police. In this system, an interface was reserved for traffic crossing and was used to communicate between the user action recognition system and the VGE. The users interacted with vehicles in the virtual scene by making traffic gestures and hence were able to learn about the traffic in the VGE.

Figure 2. Our virtual traffic geographic environment scene.

4. The TPCGS Dataset

The TPCGS dataset was constructed in order to efficiently and accurately recognize gestures. The TPCGS dataset is comprised of skeleton point positional information, which was obtained via a Kinect 2.0 sensor. It covers all eight kinds of Chinese traffic police command gestures.

Kinect skeletal tracking was created by Microsoft to obtain depth images and subsequently position and track human joint points, as shown in the middle picture in Figure 3. The Kinect sensor can be used as a virtual environment (VE) interface for viewpoint control, and Kinect skeleton recognition performs well in terms of accuracy and latency [28]. The Kinect 2.0 sensor can detect up to 20 human skeleton joints. Kinect has built-in support for joint tracking, which is beneficial in converting actual hand gestures into sequences of XYZ coordinates [29]. Its depth image resolution is 512 × 424 pixels, and the frame speed is 30 fps. The suitable measurement range was 0.5–4.5 m, and we collected data within a normal range of 1.5 m. We assume the human is facing the Kinect 2.0 sensor. From a kinematic point of view, each joint of a human body has different degrees of freedom, resulting in different contributions of gestures to human movements. For the characteristics of different postures, extraction of the region of interest can reduce the computational complexity of the whole system, thus increasing recognition speed. Through observation, it was found that the trunk part of the body is always upright, and the lower limbs transmit little effective information. Traffic police mainly use upper limb movements to convey information while directing traffic, involving arm movements and rotation of the head, so we abandoned the lower part of the key skeletal point data. Because the Kinect 2.0 sensor cannot recognize the rotation of the head, this dataset only examines the positional data of the left shoulder, left elbow, left wrist, right shoulder, right elbow, and right wrist. Therefore, in the traffic police gesture recognition algorithm, the number of joints was reduced to 6: the right hand joint, left hand joint, left elbow joint, right elbow joint, right shoulder joint, and left shoulder joint, as shown in Figure 3. Thus, the eigenvectors of a gesture at a given point can be expressed as

$$\mathrm{An} = \{\mathrm{P_i}\}_{\mathrm{i=1}}^{6},\ (\mathrm{P_i} = (\mathrm{x_i}, \mathrm{y_i}, \mathrm{z_i})), \tag{1}$$

where $\mathrm{P_i} = (\mathrm{x_i}, \mathrm{y_i}, \mathrm{z_i})$ is the vector of $n \times 1$. The gesture feature vector is 18 dimensional, where n represents the gesture feature vector in the first n frame of an image. The dataset shows both temporal continuity and spatiality. Based on the Kinect 2.0 advanced human joint recognition and tracking technology, the positional signals of skeletal points were obtained. Compared with the scenic picture, signal data were more suitable for use as input data with the ST-CNN model; data quantity of the input signals was greatly reduced, and training and testing times were decreased.

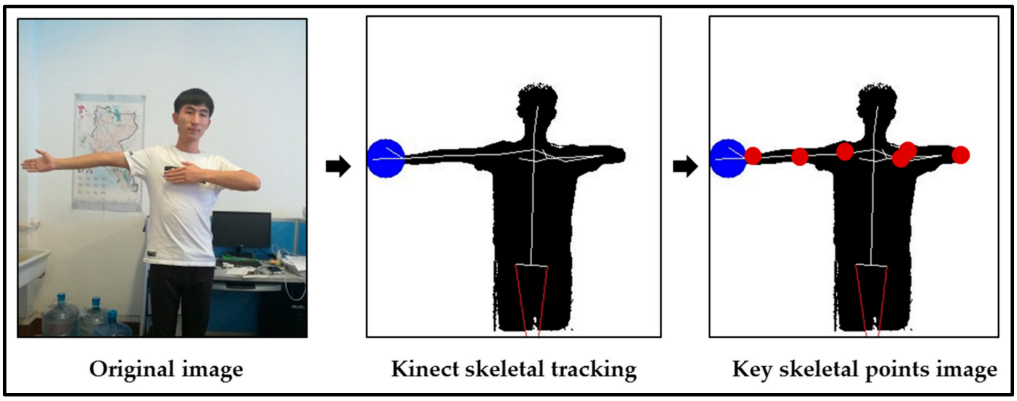

Figure 3. The acquisition process of key skeleton point positions from Kinect 2.0.

The TPCGS dataset contains the following information: volunteer ID, traffic police action name, normalized frame interval, and the 3D position information of 6 skeletal points. The dataset was collected from 10 graduate students between 20 and 30 years of age. The male to female ratio of volunteers was 1:1. The range of volunteer height was from 158 to 180 cm. Each volunteer made 8 standard traffic police gestures positioned 1.5 m away from the sensor. Ultimately, the TPCGS dataset contains 155,000 frame data, 70% of which are training samples and 30% are test samples.

5. A Novel Spatiotemporal Convolution Neural Network Model

After the TPCGS dataset with attributes of time and spatial domains was obtained, using the ST-CNN model proposed here, the spatiotemporal characteristics could be fully analyzed, and the gestures of traffic police could be recognized.

The model input is the signal data of skeleton positions. The output is the traffic police command gestures. The network architecture consists of a convolution layer, a pooling layer, a convolution layer, a fully connected layer, and an output layer. In addition, we added a dropout function that randomly sampled the parameters of the weight layer according to probability, and updated the target network to avoid overfitting. The main algorithm pipeline is shown in Figure 4.

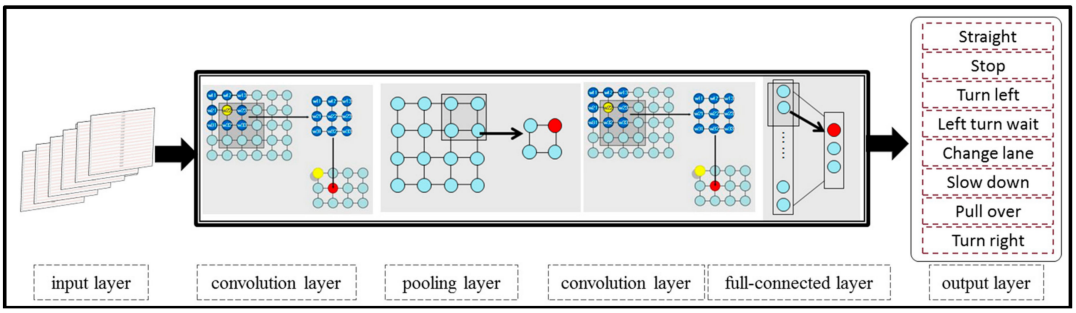

Figure 4. Spatiotemporal convolutional neural network algorithm pipeline.

- Convolution Layer

A neural network is the mathematical model of a biological nerve. Its basic unit is a neuron, and the structure of the neuron is shown in Figure 5. $p1$, $p2$, $p3$ . . . , pn are the inputs of the neuron, and l represents the current layer. The neuron sums the weighted input w, plus the offset b. After that, it calculates a function f, which is

$$p_i^l = f(\sum_{i=1}^{n} p_i^{l-1} w_i^l + b^l) = f\left(\left(w_1^l, w_2^l \ldots w_n^l\right)\begin{pmatrix} p_1^{l-1} \\ p_2^{l-1} \\ \vdots \\ p_n^{l-1} \end{pmatrix} + b^l\right) = f\left(w^{l^T} P^{l-1} + b^l\right). \quad (2)$$

The last part is the vector form of this formula. P is the input vector, W is the weight vector, and b is the offset value scalar. $f(.)$ is called the activation function.

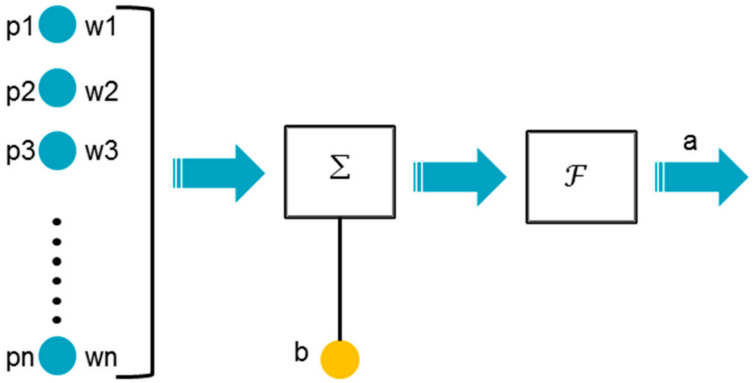

Figure 5. Neuronal structure.

The first convolution layer extracts features from input signals. The filter of $k \times k$ to convolute skeletal position signals of n frames is used to generate feature maps. To adequately access information, the sliding window moves at a speed of stride (stride = 1). The horizontal movement extracts spatial information, and the vertical movement extracts time information. The feature extraction process is illustrated in the convolution layer in Figure 4. In this study, the parameters were chosen as $k = 3$ and $n = 30$; the function tanh was chosen as the activation function.

- Pooling Layer

For the pooling layer, there are p input maps. There are p output maps, but each output map is smaller.

$$p_i^l = f\left(\beta_i^l down\left(p_i^{l-1}\right) + b_i^l\right) \tag{3}$$

where $down(\cdot)$ represents a sub-sampling function. The output map is reduced by p times in two dimensions (row and column). Each output map corresponds to a multiplicative bias β and an additive bias b. $f(\cdot)$ is the activation function.

The second layer is the down-sampling layer. The purpose of this layer is to ignore the relative positional changes such as the tilt and rotation of the target. Meanwhile, it reduces the computational load. The general operation includes max pooling, average pooling, and global average pooling. To best preserve the features of images, we choose the max pooling function in which the size of the down-sampling kernel is $p \times p$ ($p = 2$).

- Dropout Layer

Dropout loses hidden layer neurons each time. This is equivalent to training on different neural networks. Thus, the dependence between neurons is reduced. Therefore, neural networks can learn more diverse and more robust features. To prevent the neural network from over-fitting and improve generalization, the dropout layer was set up with a dropout rate of 0.5.

- Fully Connected Layer

The softmax function was applied as a fully connected layer before the output layer. The softmax function maps the output of multiple neurons into (0,1) intervals, which expresses the probability of each activity. The activity with the highest probability is then set as the predicted activity and the activity label is outputted to the final node (in red), as shown in the fully connected layer in Figure 4.

$$P(i) = \frac{exp\left(\theta_i^T x\right)}{\sum_{k=1}^{K} exp\left(\theta_k^T x\right)} \left(\sum_{k=1}^{K} P(i) = 1\right) \tag{4}$$

where θ_i and x are column vectors, and the $\theta_i^T x$ may be replaced by function $f_i(x)$. The softmax function sets the range of $P(i)$ between [0,1]. In the output layer, the model divides the pictures into eight classes that represent eight traffic command gestures.

6. Experiments

6.1. Data Preprocessing and Experimental Setup

The gesture recognition algorithm was performed with Python on a PC with a 3.40 GHZ CPU and an 8 GB memory. The skeletal point data were collected by a Kinect 2.0 sensor. The 10 volunteers simulated the traffic police command gestures for 6 key skeletal points. Based on the 3D skeleton point data provided by Kinect 2.0, the input signals of the ST-CNN model were obtained by normalization. The input signals of the eight gestures were visualized as shown in Figure 6.

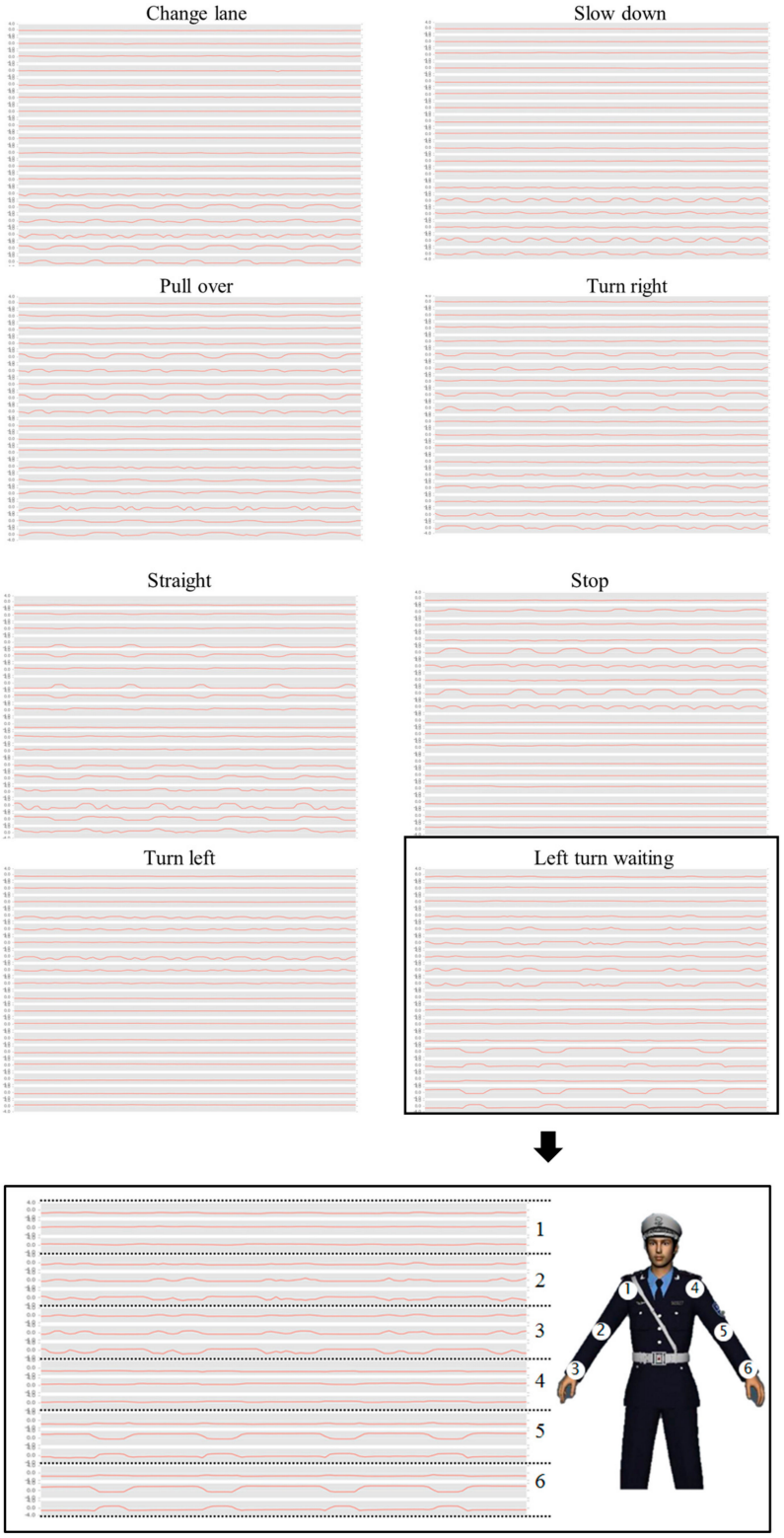

Figure 6. Input signals of the spatiotemporal convolution neural network (ST-CNN) model are visualized. The signals of eight traffic command actions are listed, respectively. Among them, the signal of left turn waiting is introduced in detail. Each three rows of signals represent a skeletal point position change, defined by xyz. Six skeletal points correspond to the right model.

The TPCGS dataset includes the sequential position information in continuous space. The features of the spatial dimension and the temporal dimension were extracted continuously by the movement of the convolution kernel. The ST-CNN model was thus established. The model could be migrated to other PC devices or mobile devices for real-time recognition of traffic police command gestures, servicing intelligent transportation and a smart city. We set up the virtual urban scene, as shown in Figure 1, and the traffic police command scene was the area of focus. The actions of volunteers were mapped to traffic police actions in the scene that controlled vehicle operations. For urban scene construction, we focused on the implementation of traffic police gesture interaction.

6.2. Results

In the traffic police gesture recognition module, we obtained a scientific deep learning model based on TPCGS dataset after continuous experiments and parameter adjustment. The size of the model was 29.79 M, which is portable and can be connected to the VGE. Users of different heights (160 cm–180 cm), ages (20–30), and genders made traffic command gestures and stood 1.5 m from the sensor. The ST-CNN model recognized gesture semantics in real time, and its accuracy and robustness were high.

In the course of the experiment, we took convolutional kernels of different sizes into consideration. Figure 7 shows the effect of changing the filter size on performance. Filter sizes that achieved high performance on the test set range from 2×2 to 8×8. It can be concluded from this plot that the convolutional kernel size is between 3×3 and 5×5 and that the accuracy of the model is higher. From the experimental results, a convolution kernel size of 3×3 was chosen. Figure 8 shows the effect of pooling size on performance. Unlike filter size, pooling size does not have much potential to increase the performance of the over-all classifier. Based on our multiple runs, a setting of 2×2 was best. From the previous best result configuration, the results of tuning the learning rate are shown in Figure 9. The model learning rate presented in this paper is 0.001. Correspondingly, in the fully connected layer, a multilayer perceptron with a 1000 node fully connected layer was set up.

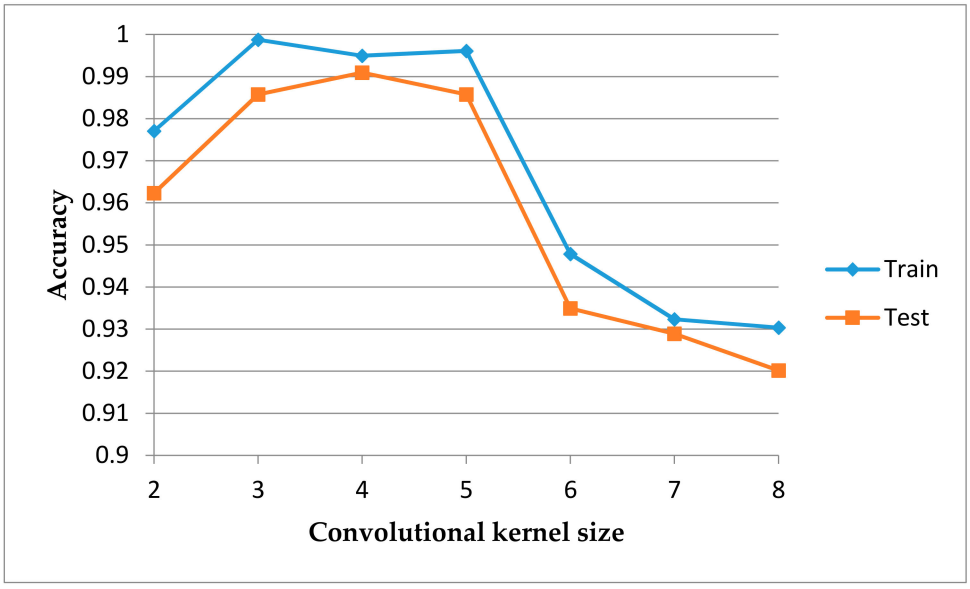

Figure 7. Effects of changing the ST-CNN model convolutional kernel size.

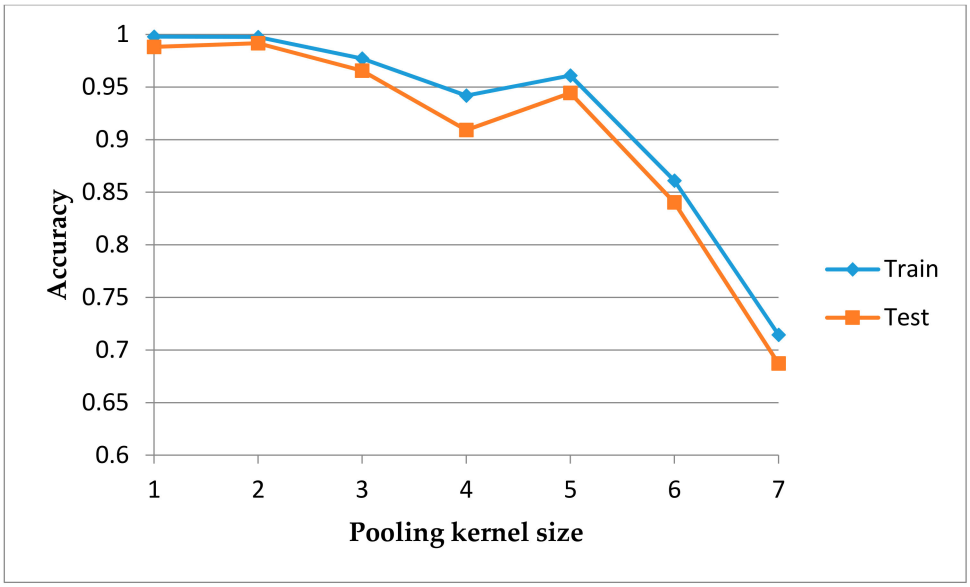

Figure 8. Effects of changing the ST-CNN model pooling kernel size.

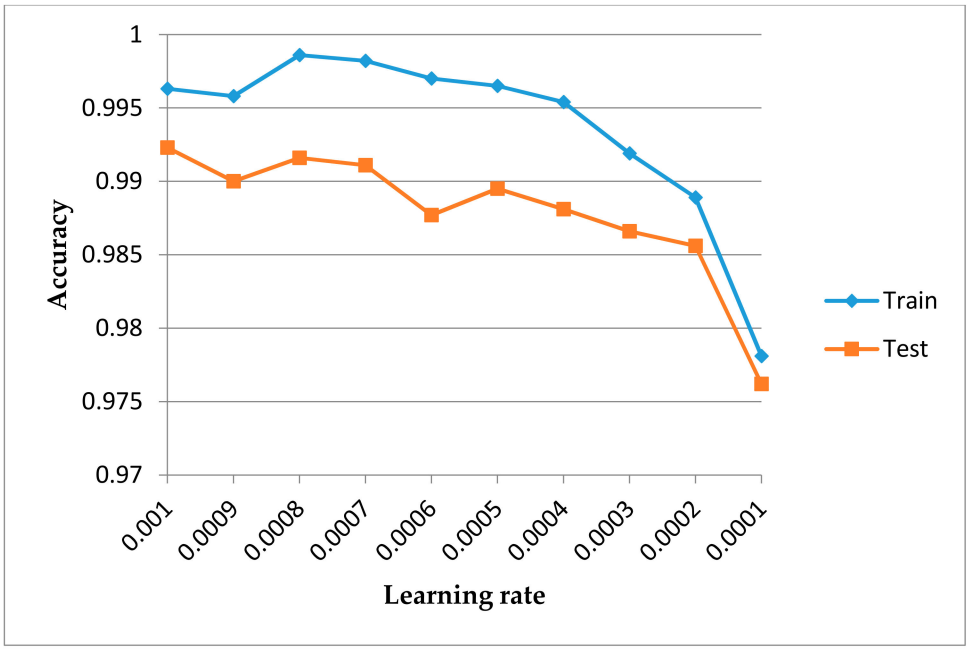

Figure 9. Effects of changing the ST-CNN model learning rate.

In the model training stage, the model load and training process lasted 20 min, and a forward propagation training process lasted 15 s. The experiment was a supervised learning process. The performance evaluators were the similarity between the experimental results and the label data of the training process. The results of the training process are shown in Figure 10. The training occurred 50 times. When training the tenth time, the accuracy rate reached 90%. As training times increased, the accuracy rate increased but at a slower rate. Finally, the training accuracy rate was approximately 97.78%. In total, 70% of the data in the TPCGS dataset was randomly selected as the training data set, and the remaining 30% was used as the test dataset. The test accuracy rate of the test dataset was approximately 96.67%.

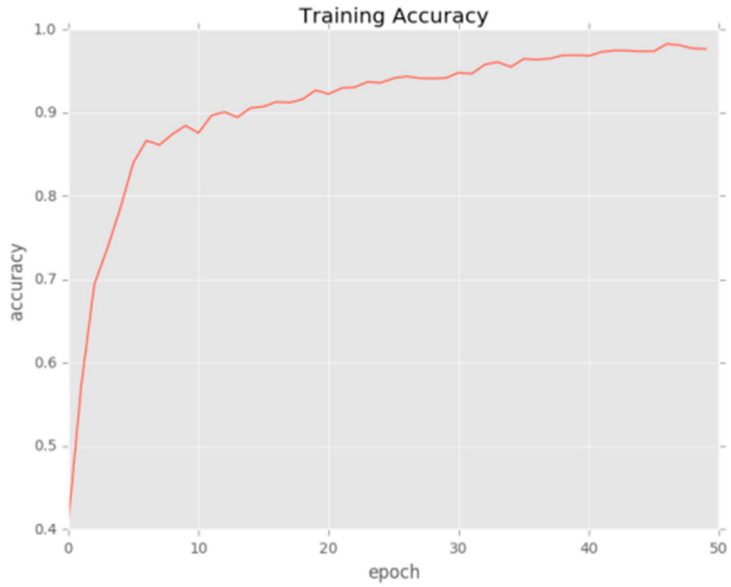

Figure 10. Train accuracy of the ST-CNN model.

To better demonstrate the experimental results, a real-time experiment was implemented. The user stood 1.5 m away from the sensor—the same as done in the training. Each frame and the previous 29 frames together formed an input unit, calculated by the ST-CNN model, and an action classification result, an output unit, was obtained. When the user changed actions, the output unit was unstable. We entered the reprocessing operation to stabilize the results. Each output unit and the previous two output units jointly judged the action of the current frame. When the three output units were inconsistent, the results of the n frame were selected. When the results of the three output units were consistent with two or more results, consistent results were selected. This estimation method preserved the continuity of actions, and the statistical results were more stable on time scales. The estimate method is shown in Table 1.

Table 1. Real-time test result estimate method.

Output Unit ($n-2$)	Output Unit ($n-1$)	Output Unit (n)	Result
a	b	c	Action c
a	a	b	Action a
a	b	a	Action a
a	b	b	Action b
a	a	a	Action a

After analyzing the test results, as shown in Table 2, it was found that the recognition results of action Turn_left and action Change_lane are often confused. Analysis of the motion trajectory image showed that the amplitude of the left arm movement was small in the Turn_left dataset, so the Turn_left dataset was optimized, and the test results were in turn optimized. The accuracy of each action in the test dataset and the real-time test accuracy are shown in Table 2. Among them, the average test accuracy of test dataset is 96.67%, and the average accuracy of the real-time test is 93.0%.

In order to compare this method with state-of-the-art methods, we adopted the same experimental setting and the same dataset as used in [30–32]. We divided the 20 actions in the MSR-Action3D data set into 3 groups (AS1, AS2, and AS3). Each group contained eight actions, and similar actions were all divided into the same group. The data of Subjects 1, 3, 5, 7, and 9 were for training, and the others were for the test. What is clear from the data in Table 3 is that, compared with the random forest, RNN, and SVM algorithms, the ST-CNN model proposed in this paper has a higher accuracy rate.

Table 2. Test results of the ST-CNN model.

Action	Straight	Stop	Turn_Left	Turn_Left_Waiting
Test dataset accuracy (%)	97.2	96.5	95.0	98.3
Real-time test accuracy (%)	90.3	94.6	85.4	93.3
Action	**Change_Lane**	**Slow_Down**	**Pull_Over**	**Turn_Right**
Test dataset accuracy (%)	94.8	98.5	96.5	96.6
Real-time test accuracy (%)	95.1	88.5	98.5	98.3

Table 3. Recognition rates for various methods on the MSR-Action3D dataset.

Dataset	Random Forests Method [30]	Recurrent Neural Network [31]	Linear SVM [32]	ST-CNN
AS1 (%)	94.87	93.33	95.29	95.41
AS2 (%)	87.00	94.64	83.87	95.45
AS3 (%)	100.00	95.50	98.22	97.50
Average (%)	93.96	94.49	92.46	96.12

7. Conclusions and Discussion

After a TPCGS dataset with time and spatial domains was obtained, the ST-CNN model proposed in this paper could be used to fully analyze spatiotemporal characteristics and recognize the gestures of traffic police.

A new traffic command gesture recognition method is thus presented. We built the ST-CNN model based on the depth data provided by a depth camera. Real-time traffic gesture signals were applied to a virtual urban scene. The recognition module result was connected to the reserved interface of the virtual traffic environment module, and the signals controlled vehicles at traffic crossroads. Traffic police models were changed according to differences in the signals, so that the traffic police in the virtual scene were more realistic.

The TPCGS dataset was built to compensate for the lack of a gesture dataset for traffic police command gestures. A new deep learning method for extracting features is presented here for the recognition of traffic police command gestures. Ultimately, a virtual urban traffic intersection environment was built to test the model, and the model was found to be stable and robust.

Future works employing the real-time traffic command gesture recognition method will take the pose (position, size, and orientation), deformation, motion speed, sensor frame rate, texture [33], and other factors into account. Different types of people, such as children, will be added to the TPCGS dataset so that more people will be involved in the 3D interaction of this virtual traffic environment.

Acknowledgments: Qingdao major projects of independent innovation (No. 16-7-1-1-zdzx-xx), Qingdao source innovation program (No. 17-1-1-6-jch), The Fundamental Research Funds for the Central Universities (No. 201762005) and the National Key Scientific Instrument and Equipment Development Projects of National Natural Science Foundation of China (No. 41527901).

Author Contributions: Chunyong Ma is mainly responsible for study designing and data analysing. Yu Zhang gives literature search and writes. Anni Wang and Yuan Wang help to support the experimental datasets and analyzed the data. Ge Chen contributes to the design of study. All authors participated in the editing of the paper.

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

References

1. Li, X.; Lv, Z.; Hu, J.; Zhang, B.; Yin, L.; Zhong, C.; Wang, W.; Feng, S. Traffic management and forecasting system based on 3D GIS. In Proceedings of the 2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, Shenzhen, China, 4–7 May 2015; pp. 991–998.
2. Lin, H.; Chen, M.; Lu, G. Virtual geographic environment: A workspace for computer-aided geographic experiments. *Ann. Assoc. Am. Geogr.* **2013**, *103*, 465–482. [CrossRef]

3. Song, X.; Xie, Z.; Xu, Y.; Tan, G.; Tang, W.; Bi, J.; Lie, X. Supporting real-world network-oriented mesoscopic traffic simulation on GPU. *Simul. Model. Pract. Theory* **2017**, *74*, 46–63. [CrossRef]
4. Yang, X.; Li, S.; Zhang, Y.; Su, W.; Zhang, M.; Tan, G.; Zhang, Q.; Zhou, D.; Wei, X. Interactive traffic simulation model with learned local parameters. *Multimedia Tools Appl.* **2017**, *76*, 9503–9516. [CrossRef]
5. Rautaray, S.S.; Agrawal, A. Vision based hand gesture recognition for human computer interaction: A survey. *Artif. Intell. Rev.* **2015**, *43*, 1–54. [CrossRef]
6. Wang, B.; Yuan, T. Traffic Police Gesture Recognition Using Accelerometer. In Proceedings of the IEEE Sensors Conference, Lecce, Italy, 26–29 October 2008; pp. 1080–1083.
7. Le, Q.K.; Pham, C.H.; Le, T.H. Road traffic control gesture recognition using depth images. *IEEK Trans. Smart Process. Comput.* **2012**, *1*, 1–7.
8. Kela, J.; Korpipää, P.; Mäntyjärvi, J.; Kallio, S.; Savino, G.; Jozzo, L.; Di Marca, S. Accelerometer-based gesture control for a design environment. *Pers. Ubiquitous Comput.* **2006**, *10*, 285–299. [CrossRef]
9. Ji, S.; Xu, W.; Yang, M.; Yu, K. 3D convolutional neural networks for human action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 221–231. [CrossRef] [PubMed]
10. Wang, H.; Oneata, D.; Verbeek, J.; Schmid, C. A robust and efficient video representation for action recognition. *Int. J. Comput. Vis.* **2016**, *119*, 219–238. [CrossRef]
11. Feichtenhofer, C.; Pinz, A.; Zisserman, A. Convolutional two-stream network fusion for video action recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 27–30 June 2016; pp. 1933–1941.
12. Li, Q.; Qiu, Z.; Yao, T.; Mei, T.; Rui, Y.; Luo, J. Action Recognition by Learning Deep Multi-Granular Spatio-Temporal Video Representation. In Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval, New York, NY, USA, 6–9 June 2016; pp. 159–166.
13. Mitra, S.; Acharya, T. Gesture recognition: A survey. *IEEE Trans. Syst. Man Cybern.* **2007**, *37*, 311–324. [CrossRef]
14. Raheja, J.L.; Chaudhary, A.; Singal, K. Tracking of Fingertips and Centers of Palm Using Kinect. In Proceedings of the IEEE 2011 Third International Conference on Computational Intelligence, Modelling and Simulation (CIMSiM), Chennai, India, 14–16 December 2011; pp. 248–252.
15. Liu, Y.; Wang, X.; Yan, K. Hand gesture recognition based on concentric circular scan lines and weighted K-nearest neighbor algorithm. *Multimedia Tools Appl.* **2016**, *77*, 209–233. [CrossRef]
16. Wang, X.; Yan, K. Immersive human–computer interactive virtual environment using large-scale display system. *Future Gener. Comput. Syst.* **2017**. [CrossRef]
17. Wang, X.; Wang, J.; Yan, K. Gait recognition based on Gabor wavelets and (2D) 2PCA. *Multimedia Tools Appl.* **2017**. [CrossRef]
18. Fujiyoshi, H.; Lipton, A. J.; Kanade, T. Real-time human motion analysis by image skeletonization. *IEICE Trans. Inf. Syst.* **2004**, *87*, 113–120.
19. Chaudhry, R.; Ravichandran, A.; Hager, G.; Vidal, R. Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009), Miami, FL, USA, 20–25 June 2009; pp. 1932–1939.
20. Yang, J.; Xu, Y.; Chen, C.S. Human action learning via hidden Markov model. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **1997**, *27*, 34–44. [CrossRef]
21. Schuldt, C.; Laptev, I.; Caputo, B. Recognizing human actions: A local SVM approach. In Proceedings of the 17th International Conference on Pattern Recognition (ICPR 2004), Cambridge, UK, 23–26 August 2004; Volume 3, pp. 32–36.
22. Yu, K.; Xu, W.; Gong, Y. Deep Learning with Kernel Regularization for Visual Recognition. In Proceedings of the Advances in Neural Information Processing Systems, Whistler, BC, Canada, 11 December 2009; pp. 1889–1896.
23. Jiang, W.; Yin, Z. Human activity recognition using wearable sensors by deep convolutional neural networks. In Proceedings of the 23rd ACM International Conference on Multimedia, Brisbane, Australia, 26–30 October 2015; pp. 1307–1310.
24. Yang, J.; Nguyen, M.N.; San, P.P.; Li, X.L.; Krishnaswamy, S. Deep Convolutional Neural Networks on Multichannel Time Series for Human Activity Recognition. In Proceedings of the IJCAI 2015, Buenos Aires, Argentina, 25–31 July 2015; pp. 3995–4001.

25. Ronao, C.A.; Cho, S.B. Human activity recognition with smartphone sensors using deep learning neural networks. *Expert Syst. Appl.* **2016**, *59*, 235–244. [CrossRef]
26. Lee, S.M.; Yoon, S.M.; Cho, H. Human activity recognition from accelerometer data using Convolutional Neural Network. In Proceedings of the 2017 IEEE International Conference on Big Data and Smart Computing (BigComp), Jeju Island, Korea, 13–16 February 2017; pp. 131–134.
27. Lv, Z.; Li, X.; Zhang, B.; Wang, W.; Zhu, Y.; Hu, J.; Feng, S. Managing big city information based on WebVRGIS. *IEEE Access* **2016**, *4*, 407–415. [CrossRef]
28. Livingston, M.A.; Sebastian, J.; Ai, Z.; Decker, J.W. Performance measurements for the Microsoft Kinect skeleton. In Proceedings of the 2012 IEEE Virtual Reality Short Papers and Posters (VRW), Costa Mesa, CA, USA, 4–8 March 2012; pp. 119–120.
29. Raheja, J.L.; Minhas, M.; Prashanth, D.; Shahb, T.; Chaudhary, A. Robust gesture recognition using Kinect: A comparison between DTW and HMM. *Optik-Int. J. Light Electron Opt.* **2015**, *126*, 1098–1104. [CrossRef]
30. Zhu, Y.; Chen, W.; Guo, G. Fusing spatiotemporal features and joints for 3D action recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Portland, OR, USA, 23–28 June 2013; pp. 486–491.
31. Du, Y.; Wang, W.; Wang, L. Hierarchical recurrent neural network for skeleton based action recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1110–1118.
32. Vemulapalli, R.; Arrate, F.; Chellappa, R. Human action recognition by representing 3D skeletons as points in a lie group. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 588–595.
33. Bau, O.; Mackay, W.E. OctoPocus: A dynamic guide for learning gesture-based command sets. In Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology, Monterey, CA, USA, 19–22 October 2008; pp. 37–46.