ISSN 2616-5775 Vol. 6, Issue 6: 49-54, DOI: 10.25236/AJCIS.2023.060607

A VR User Behavior Classification Method Integrating Scene Information and Operation Information

Hao Zhou[1,*], Bo Mao[1]

[1]*College of Information Engineering, Nanjing University of Finance & Economics, Nanjing, China*
[*]*Corresponding author: 1437243780@qq.com*

Abstract: *With the rapid growth of the virtual reality industry, the number of users participating in virtual reality has rapidly increased, and analyzing user behavior in virtual reality scenarios has become increasingly important. Compared with general time series classification tasks, the input sources in virtual reality are more complex and diverse, and traditional time series classification models are difficult to handle this complexity. This article designs a series of methods for classifying VR user behavior and conducts experiments. Apply video motion recognition methods to VR motion recognition based on VR scene data. Due to the ability to obtain a series of image data from both videos and scenes, this article uses the method of video action recognition to process VR scene data. On this basis, a series of fusion methods were designed to fuse the operational information of VR. Experiments have shown that these methods can effectively handle action recognition problems in VR scenes.*

Keywords: *time series classification; VR User Behavior Classification; VR*

1. Introduction

Virtual Reality (VR) technology is a computer hardware based simulation system that can provide people with an interactive, virtual, and immersive environment. It allows people to see the virtual world through wearable headsets and interact with it through a series of input devices.

In recent years, the VR industry has developed rapidly, with terminal products becoming increasingly diverse and mature. In 2021, the global shipment volume of VR devices exceeded 10 million units, and the global virtual reality industry scale exceeded 80 billion yuan. The application scenarios of VR have also become diversified. In addition to traditional games, film and television, VR's influence in industries such as healthcare, education, labor training, and manufacturing has gradually increased.

The scale of VR industry is constantly expanding, and its application fields are becoming increasingly widespread. In the future, VR technology will be more closely linked to people's lives. Therefore, conducting behavioral analysis of users in VR scenarios is of great significance. However, current research is often limited. In current research, VR is often used as a means and tool for researchers to create scenarios and then study people's behavioral habits in a specific scenario. The current research lacks a general analysis method for user behavior in VR scenarios. In response to the above shortcomings, this article provides a VR scene user behavior classification model, which classifies user behavior by analyzing user operations and scene data. By studying user behavior in VR scenarios, it is possible to better analyze user operating habits and familiarity, thereby designing VR scenarios, enhancing user experience and comfort, and optimizing application development and improvement.

2. Related work

Video action recognition is an important task in video understanding, which mainly involves identifying the actions of people in the video. Deepvideo [1] was the earliest work to apply deep learning methods to video action recognition. It uses a single 2D convolutional neural network to process each frame in the video. Although Deepvideo's performance did not meet expectations, it proved that general 2D convolutional neural networks are difficult to handle temporal video data, and

ISSN 2616-5775 Vol. 6, Issue 6: 49-54, DOI: 10.25236/AJCIS.2023.060607

opened up ideas for subsequent work. Dual stream networks [2] are a mainstream approach for processing video action recognition. Optical flow is a representation of the movement of objects and scenes in an image. The idea of a dual stream network is to extract the optical flow graph of a video to describe the motion information in the video. It consists of two parts, one is the spatial flow and the other is the temporal flow. C3D [3] regards video data as a three-dimensional tensor with two spatial dimensions and one temporal dimension. By changing the 2D convolutional kernel to a 3D convolutional kernel and changing the 2D pooling to a 3D pooling, the training of video input is achieved. [4] Explored the combination of 2D network and 3D network. By replacing some underlying 3D networks with 2D networks, higher accuracy and faster training speed can be achieved. [5] MiCT was proposed, which reduces the complexity of spatiotemporal fusion of models by integrating 3D and 2D networks together. [6] Trajectory information was extracted in the time dimension, which uses trajectory convolution to extract features from trajectory information. At the same time, combining appearance and motion trajectory information, significant improvements were achieved. [7] A pseudo 3D network structure is proposed, which first performs 2D convolution in the spatial domain and then convolutions these feature maps in the temporal domain, improving computational efficiency. Vision Transformer [8] is the earliest work to extend the self-attention mechanism from the 2D spatial dimension of images to the 3D spatiotemporal dimension of videos. It attempts to integrate five structures of attention, including spatial attention mechanism, spatiotemporal common attention mechanism, split spatiotemporal attention mechanism, local global attention mechanism and axial self-attention mechanism.

3. Proposed method

3.1. Data pre-processing

Due to the fact that the data obtained in the current scene is divided into a series of image data of the scene and numerical data of user operation information, both of which can be regarded as temporal data. In order to facilitate batch processing, it is necessary to specify a fixed time series data length for each of them. In this experiment, a fixed time series length of 16 was selected for both types of data.

As shown in Figure 1, each sample corresponds to a series of images, and due to the different durations of each sample, the number of captured images also varies. For the convenience of batch processing, the number of scene images corresponding to each sample is now adjusted to a fixed size of 16. Randomly select 16 images from the corresponding samples, and if there are less than 16, copy the last image several times to make the total number of images 16. Adjust the image size and keep the relative order of the images unchanged.

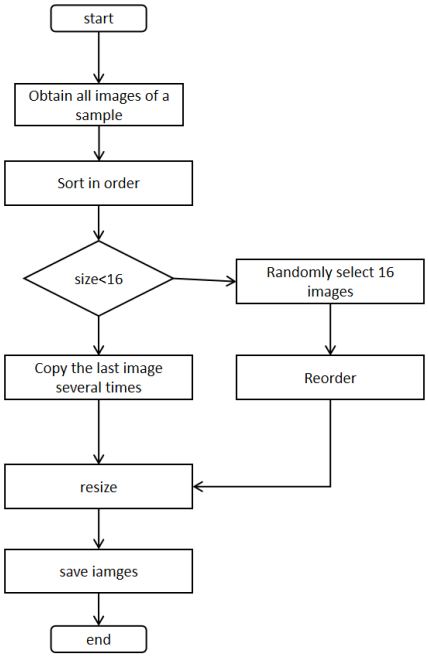

Figure 1: Scene image preprocessing flowchart.

ISSN 2616-5775 Vol. 6, Issue 6: 49-54, DOI: 10.25236/AJCIS.2023.060607

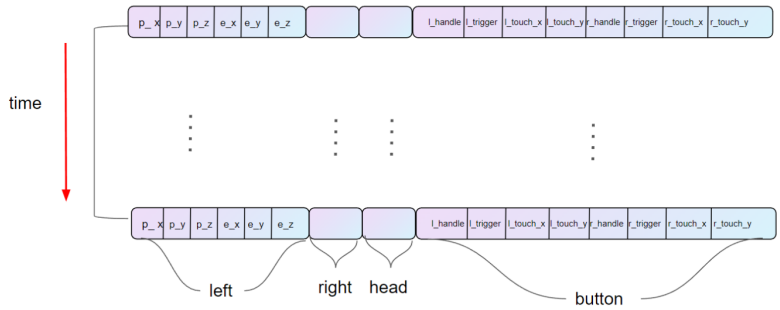

Figure 2: Processed operational data structure.

Figure 2 shows the data structure of the processed operational data. The operational data of each sample can be represented as 16 records. These records are arranged in order. Each record can be divided into four parts. The first three parts represent the position and angle information of the left hand controller, right hand controller, and headband display, and the last part is the button information. For positional angle information, p_ x,p_ y,p_z represents the position coordinate, e_ x,e_ y,e_ z represents the angular coordinate. For button, l_ handle represents holding button of the left controller, l_ trigger represents the left hand trigger button, l_ touch_ x l_ touch_ y represents the coordinates of the left hand touchpad. r_ handle，r_ trigger，r_ touch_ x，r_ touch_ y represents the right hand for the button.

3.2. A two-dimensional representation method for temporal data

Assuming the input temporal data$T = (t_1, t_2, \cdots, t_n)$， $t_i \in R, i \in [1, n]$, n is the length of the temporal data, is the value of the temporal data at the i-th moment, the value of the sequence is a real number, and T represents a univariate temporal sequence. Next, we will process the temporal data T and represent it as a two-dimensional matrix. Firstly, subtract the values of two adjacent moments of sequence T to obtain a new sequence S. The new temporal sequence represents the changes in the original sequence, with a length of L.

$$s_i = t_i - t_{i+1}, s_i \in R, i \in [1, L], L=n-1 \tag{1}$$

Now adjust the length of S, assuming the length of the target sequence to be adjusted is m. If the length of S is greater than m, reduce the dimensionality of S. If the length of S is less than m, supplement the sequence eined is denoted as X, $X = (x_1, x_2, \cdots, x_m), x_i \in R, i \in [1, m]$ and the calculation method for X is as follows.

$$x_i = \begin{cases} \frac{m}{L} \sum_{j=\left\lfloor \frac{i-1}{m} \times L \right\rfloor + 1}^{\left\lceil \frac{i}{m} \times L \right\rceil} |s_j|, L > m \\ |s_i|, \quad L = m \\ |s_r|, r = \left\lfloor \frac{i}{m} \times L \right\rfloor, L < m \end{cases} \tag{2}$$

Now, standardize the sequence X, and record the sequence with normal distribution as U,$U = (u_1, u_2, \cdots, u_m), u_i \in R, i \in [1, m]$ and the ith element in U can be expressed asu_i. Among them, the length of the sequence is m, μandσare the mean and variance of the temporal data X.

$$u_i = \frac{x_i - \mu}{\sigma}, i \in [1, m] \tag{3}$$

After obtaining the standardized series u, generate a two-dimensional matrix as follows:

$$M = \begin{pmatrix} 2u_1 & u_2 + u_1 & \cdots & u_m + u_1 \\ u_1 + u_2 & 2u_2 & \cdots & u_m + u_2 \\ \cdots & \cdots & \cdots & \cdots \\ u_1 + u_m & u_2 + u_m & \cdots & 2u_m \end{pmatrix} \tag{4}$$

Finally, through min max standardization, the matrix M is transformed into a grayscale matrix with values ranging from 0 to 255. The specific method is as follows, which is the final representation of the two-dimensional matrix of time series data T.

$$F = \frac{M - \min(M)}{\max(M) - \min(M)} \times 255 \tag{5}$$

ISSN 2616-5775 Vol. 6, Issue 6: 49-54, DOI: 10.25236/AJCIS.2023.060607

3.3. Overall structure

Compared to image data, the amount of data required for manipulating data is much smaller. For the preprocessed temporal data in this chapter, its data volume is much smaller than the image data of the sample. Therefore, in order to integrate operational data and image data, it is necessary to continuously expand the operational data to ensure that the shape of the operational data is consistent with the shape of the image.

The time step of the preprocessed operation in this chapter is 16, which is consistent with the number of images. The vector length for each time step of the operation data is 26, and it needs to be converted into a two-dimensional matrix. By using the two-dimensional representation method of sequences proposed in this section, the operational data is transformed into tensors. Then expand the channel dimension to make the operational data consistent with the image data in the channel dimension. At this point, the shapes of the two are consistent, and the fusion is performed by directly adding them together. Finally, the C3D model is used for processing.

Figure 3 shows the network structure that integrates scene images and operational information. By using the two-dimensional representation method of sequences proposed in this article, the operational data is transformed into tensors. Then expand the channel dimension to make the operational data consistent with the image data in the channel dimension. Fusion is performed through direct addition, and finally processed using the C3D model.

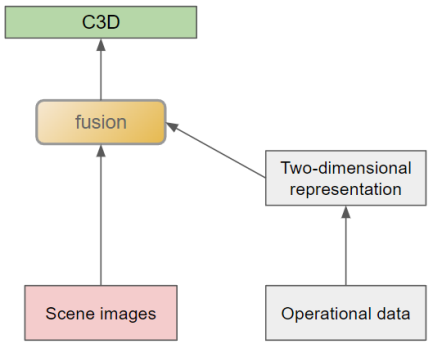

Figure 3: Network structure diagram integrating scene images and operational information.

4. VR Dataset

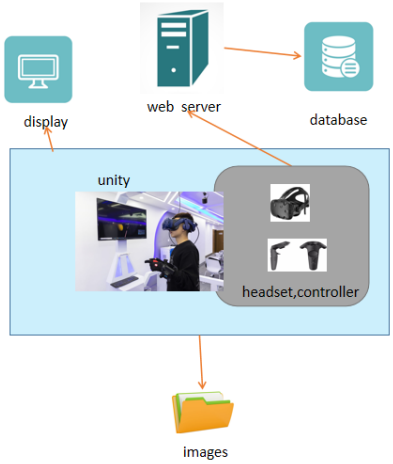

Figure 4: This caption has more than one line so it has to

As shown in Figure 4, the overall structure of the VR data collection system in this article is shown. The VR device used in this article is the HTC VIVE Pro2, which includes a head mounted display, left and right hand controllers, and two locators. This article obtains scene information through screenshots. When processing video type data, frames are extracted at a certain frequency to transform a video into

ISSN 2616-5775 Vol. 6, Issue 6: 49-54, DOI: 10.25236/AJCIS.2023.060607

several ordered images.

As shown in Figure 5, this dataset contains four scenarios, namely archery, rotary table, bowling, and remote control car. Each scene has several actions, one action is a category, totaling 10 categories.

Archery: Users pick up the bow, attach the arrow to the bow, stretch the bow, aim at the target, release the trigger button, and shoot the bow and arrow. There are two situations here: hit and miss. The actions of users who hit and those who did not hit are completely consistent and need to be judged based on scene information.

Bowling: The user throws the bowling ball in a consistent motion. When the ball collides with the cup, the cup will fall down, and after a while, the cup will return to its standing position. If there is no collision, the cup remains stationary. It is difficult to judge based solely on operational information, and classification needs to be combined with scene information.

Rotary table: divided into left hand rotation and right hand rotation. Using operational information can easily analyze which hand is moving. The use of scene information can be judged based on the different 3D models of the walking right hand, especially the direction of the thumb is significantly different, which is more difficult compared to manipulating data.

Remote control car: Use the remote control to control the car's movement in the scene, divided into four categories: forward, backward, left, and right movements. The direction of movement here is based on the car itself, rather than relative to the operator, and it is difficult to recognize the car at a distant location, so using scene information for classification is more difficult and using operational data is easier.

Figure 5: VR Scenes

5. Experiment

This chapter of the experiment was implemented using Pytorch, using a GTX 2070 device with a graphics memory size of 8GB. This article conducts experiments on the obtained VR scene dataset, which is divided into training and testing sets according to 7:3. The implementation of 3D convolutional neural network refers to the open-source method of the network, and only necessary modifications have been made to the input and output parts in order to run in this dataset. The Timsformer section refers to the official implementation of Facebook, and this article selects the network implementation section. Due to insufficient graphics memory on the experimental machine, the complete Time transformer network cannot be run. Therefore, this article has reduced the scale of the model, which is not consistent with the original implementation of Time transformer. Different models have different learning rates, and this article selects appropriate learning rates for different models through experiments. The number of iterations selected in this article is 50, and different methods can basically converge.

In the results of Table 1, the 2D convolutional neural network of the first three scenarios achieved high accuracy, indicating that these scenarios have little discrimination on the model. In the scenario of remote controlled cars, the performance of 2D networks is poor. Distinguishing which direction the car is moving in depends on the changes in the front and back frames, and requires a certain level of

ISSN 2616-5775 Vol. 6, Issue 6: 49-54, DOI: 10.25236/AJCIS.2023.060607

temporal analysis ability, which is consistent with the expectations of this article. The accuracy of C3D in various scenarios is higher than that of 2D networks, especially in remote control vehicle scenarios. C3D has achieved relatively high accuracy, indicating that C3D networks have good performance in extracting local features and temporal information of images. From the above table, it can be seen that the representations of the timetransformer are slightly worse than those of C3D, possibly due to the relatively small dataset, which did not fully utilize the advantages of the model.

Table 1: Classification accuracy of each method (unit:%).

models	Archery	Bowling	Rotary table	Car	All
2D CNN	98.23	81.19	92.20	53.41	77.56
C3D	99.11	98.29	97.40	81.36	92.52
timesformer	95.57	96.58	87.01	62.73	83.11
transformer-encoder	64.40	65.80	97.40	95.34	80.70
Ours	99.56	98.71	98.05	96.27	98.18

The method proposed in this article integrates image information and user operation information of the scene, achieving good accuracy in various scenarios, indicating that the method can effectively recognize user behavior in VR scenes and has certain practical value.

6. Conclusion

For VR scene data, this article designs a VR scene data collection system that can save dynamic information of the scene when users use VR devices. In order to verify the ability of different models to understand scenes, this article designed four scene tasks and conducted experiments using traditional 2D convolutional neural networks, 3D convolutional neural networks in the field of video classification, and network models such as timers and transformer encoders to verify the feasibility of different models in VR scene classification tasks.

This article combines the characteristics of video classification models and VR scene data to design a VR scene user behavior classification model that integrates scene information and operation information. The classification performance is tested in VR scenes.

References

[1] Karpathy A, Toderici G, Shetty S, et al. Large-scale video classification with convolutional neural networks [J]. Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. 2014: 1725-1732.

[2] Simonyan K, Zisserman A. Two-stream convolutional networks for action recognition in videos [J]. Advances in neural information processing systems, 2014, 27.

[3] Tran D, Bourdev L, Fergus R, et al. Learning spatiotemporal features with 3d convolutional networks [C]. Proceedings of the IEEE international conference on computer vision. 2015: 4489-4497.

[4] Tran D, Wang H, Torresani L, et al. Video classification with channel-separated convolutional networks [C]. Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 5552-5561.

[5] Xie S, Sun C, Huang J, et al. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification [C]. Proceedings of the European conference on computer vision (ECCV). 2018: 305-321.

[6] Zhou Y, Sun X, Zha Z J, et al. Mict: Mixed 3d/2d convolutional tube for human action recognition [C]. Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 449-458.

[7] Zhao Y, Xiong Y, Lin D. Trajectory convolution for action recognition[C].Advances in neural information processing systems, 2018, 31.

[8] Bertasius G, Wang H, Torresani L. Is space-time attention all you need for video understanding? [C]. ICML. 2021, 2(3): 4.