Journal of Computers Vol. 32 No. 4, 2021, pp. 94-108
doi:10.53106/199115992021083204008

Using Improved Dense Trajectory Feature to Realize Action Recognition

Guo-Liang Xu, Hang Zhou*, Liang-You Yuan, Yue-Yu Huang

School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China
{18120027, hangzhou, 19125065, 20120005}@bjtu.edu.cn

Received 16 December 2020; Revised 11 April 2021; Accepted 21 April 2021

Abstract. Action recognition is prevalent in the field of intelligent image processing, and feature extraction is the key of action recognition. The number of features extracted by the improved dense trajectory algorithm (iDT) is huge and occupies a large amount of hardware storage space. Aiming to address this problem, this paper proposes to reduce the number of features by using trajectory deletion, feature clustering and salient feature extraction, while improving the accuracy of action recognition. Trajectory deletion is the deletion of trajectories with little or no information. Feature clustering is to cluster features of each action, and the cluster centers are used to represent the action. Salient feature extraction is to extract salient features within the same action category, and the salient features are used to train the codebook. In order to verify the effect of the algorithm, experiment is carried out in KTH and UCFSports datasets. After improvement, the features in KTH are reduced by 80.80% and the accuracy is increased by 0.93%. In UCFSports, the features are reduced by 79.68% and the accuracy is increased by 4.26%.

Keywords: action recognition, clustering, salient feature extraction, dense trajectory, trajectory deletion

1 Introduction

In accordance with the development of intelligent image processing, detection and tracking of targets, extraction and recognition of motion information have become important research topics [1-4]. The ultimate goal of action recognition is to determine what someone is doing at a given moment [5-7].

At present, action recognition usually needs to extract a lot of features to get good result of recognition. But a large number of features need to occupy too many storage spaces. This paper aims to reduce the number of features from action sequences and ensure the good accuracy of recognition. An action video can sample feature points intensively in each frame, and then the points are tracked to obtain the trajectories. When an action occurs, the action information can be carried around the trajectory. Feature extraction along trajectory is to obtain 3D features by combining one-dimensional time(T) and two-dimensional feature information (X-Y). Compared with two-dimensional features, the three-dimensional features have stronger feature representation ability and are more suitable to describe action sequences. In addition, the points are sampled intensively in each frame to cover as many scales and spatial positions as possible, so that more trajectories can be obtained. This method has been used in many literatures and achieved good effect of recognition [8-9]. Due to the advantages of 3D features, many features attempt to include time information to form 3D features. For example, appearance feature, spatiotemporal feature and motion feature.

The feature of appearance generally refers to the outline of the action, which can better describe the details of the action. Motion-energy images (MEI) and motion-history images (MHI) were first proposed to extract the changes in the position and appearance of action [10]. To describe the change of the appearance over time more accurately, scholars also proposed to use the spatiotemporal volume (STV) to

* Corresponding Author

represent the appearance and extract the information of action. The information of STV can be represented by calculating the differential geometric property [11] or the property obtained by solving the Poisson equation [12]. However, the feature of appearance relies heavily on foreground detection and tracking, and it is easily affected by the diversity of clothing.

Spatiotemporal features mainly refer to the algorithm that describes the action according to the particularity of the action. In recent years, spatiotemporal features have been widely used in action recognition, including spatiotemporal cube, spatiotemporal interest point and spatiotemporal context. Spatiotemporal cube is to map the feature points to the cube for representation. Seo et al. used 3D space time local regression kernels (3D LSKs) to represent the action, and completed the action recognition by matching the spatiotemporal cube [13]. The method based on spatiotemporal interest points is to extract some irrelevant points in the action sequence, and then the characteristics of these points is analyzed to realize the representation of action. The common spatiotemporal interest points include Harris [14], SIFT [15] and others. Literature [16] proposed a spatiotemporal interest points based on flow vorticity, which can extract obvious spatiotemporal points around the moving action. In literature [17], spatiotemporal interest points are constrained to reduce the influence of useless interest points on action recognition. The idea of spatiotemporal context is to construct the relationship between the action and the environment where the object is located, and then the action recognition is realized through the relationship information of the action in the environment. Literature [18] constructed the framework of action recognition by using the context features of “based on image”, “based on human” and “based on action”. However, the spatiotemporal feature points are easily affected by noise, which can reduce their effect of representation.

Motion features are commonly used at present, including trajectory, direction and optical flow, etc. These features are not affected by the shape of human body, and the extracted information is comprehensive. Wang et al. proposed dense trajectory [8] (DT) and improved dense trajectory [9] (iDT), which showed excellent result of action recognition. The algorithm based on the dense trajectory obtains the feature point by sampling the feature points intensively, and then the points are tracked by using the optical flow to form trajectories. Finally, trajectories (Traj), histograms of oriented gradients (HOG), histograms of optical flow (HOF) and motion boundary histograms (MBH) features are extracted along the trajectory.

Because of the abundant features are extracted along the trajectory, so the features have strong representation ability. However, it is due to the intensive sampling points in each frame, so the plenty of features will occupy a huge amount of hardware storage space. In order to overcome the above problems, the key research is that this work proposes trajectory deletion, feature clustering and salient feature extraction to reduce the number of features and improve the accuracy of action recognition.

The technical contributions of this paper are as follows:

(1) The trajectory deletion rule of iDT is improved. The trajectories extracted by iDT contain trajectories with little or no information. In this paper, the standard deviation and the length of the ten coordinate points in the trajectory are calculated, and trajectory that does not meet the rule is deleted.

(2) The features are clustered using the Mean-shift. The number of clustering centers is adaptive, and it is adjusted according to the number of trajectories in each action. Finally, the clustering centers of features are used to represent the action.

(3) Salient feature extraction is used to extract the salient features of the training set. The salient features are used to train the codebook so that the key features can be coded. In this way, the descriptors for each action are more representative.

By making above contributions to improve the iDT, the algorithm performance has been significantly improved. Trajectory deletion and feature clustering delete a lot of features, which significantly reduce the feature volume. Salient feature extraction improves the accuracy of action recognition. In KTH, the feature volume is reduced by 80.80% and the accuracy is increased by 0.93%. In UCFSports, the feature volume is reduced by 79.68% and the accuracy is increased by 4.26%.

The rest of this paper is organized as follows: In the second section, we introduce the related work of action recognition. In the third section, we introduce trajectory deletion, feature clustering and salient feature extraction. In the fourth section, we test the algorithm in KTH and UCFSports datasets. Finally, in the fifth section, we summarize the work of this paper.

2 Related Work

Action recognition can be divided into three parts: feature extraction, feature coding and action classification. Feature extraction is the key to realize action recognition. In this section, we focus on introducing feature extraction algorithms based on trajectory, and analyzing the problems of various algorithms. Finally, we briefly introduce feature coding and action classification.

2.1 Feature Extraction Based on Trajectory

Feature extraction based on trajectory is one of the most effective methods to extract motion features for action recognition. Literature [8] proposed the DT to extract features. However, DT is easily affected by background changes and camera motion. To solve this problem, literature [9] further improved the DT to obtain the iDT. The feature extraction method of iDT is the same as DT, but iDT eliminates camera motion and the motion trajectory on the background by estimating the warped optical flow. Both DT and iDT are the method based on dense trajectory. At present, the method based on dense trajectory is the most commonly traditional method to realize action recognition. Z Lan et al. used the method of dense trajectory to take feature points, and they proposed Multi-skip Feature Stacking (MIFS) that used a set of parameterized multi time hopping differential filters to extract features [19]. Literature [20] used dense trajectory to extract action features, and then recognized action by using the method based on transfer learning. Since the dense trajectory method intensively sample the feature points, this method will extract plenty of features, which occupy a large amount of storage resources. Therefore, many scholars improved it by reducing the number of trajectories. The main improvement methods generally include replacing the dense feature points with the key points, combining the dense trajectory method with the deep network, and selecting the trajectory of the interesting region. Camarena F et al. selected and tracked six key points through gesture estimation of human body, and then extracted features along the key trajectories [21]. The method proposed by Liang X et al. is similar to the idea of trajectory deletion in literature [21]. They understood the action recognition problem as trajectory image recognition problem, and they used OpenPose to extract human bone data from RGB camera and selected eighteen key points [22]. Although literature [21-22] reduce the number of trajectories, but literature [21] needs to estimate the posture of human body, and literature [22] needs to analyze the bone data of human body in order to select key points. Dense trajectory methods can also be combined with deep network. Literature [23] combined the trajectory with the feature of deep network to obtain the feature of trajectory-pooled deep convolutional descriptor (TDD), which had the dual advantages of manual feature and deep learning feature. However, the feature extraction and calculation trajectory of TDD are independent, which are complex in calculation. In literature [24], it is proposed to stack the offset map for trajectory convolution and the original apparent feature map. In this way, the feature information is fused, and the ability of features to represent action is improved. In order to reduce the number of trajectories, it is a good idea to select the trajectory of the interesting region. Xiao x et al. reduced the number of trajectories by extracting trajectories in the center region, which is two-thirds of the image [25]. But literature [25] can't cope well with the situation that trajectories are outside the center region. Lu et al. selected the human region by salient region detection and human body detection respectively, and then they extracted the trajectory of the human region [26-27]. By selecting the trajectories of human regions, a large number of trajectories in background can be eliminated, which can effectively improve the accuracy of action recognition.

To sum up, the dense trajectory methods occupy a large amount of hardware resources due to the intensive sampling feature points. If only key points are extracted, it is necessary to analyze the human posture or skeleton data. Deep learning method can extract more abundant feature information. But if deep learning method is adopted, it needs a large amount of data for training and needs high hardware. In this paper, we propose to improve iDT by trajectory deletion, feature clustering and salient feature extraction. Feature clustering is used to extract the feature center. In this way, the feature center is used to replace the feature of intensive sampling, which can significantly reduce the feature volume. Compared with the related work, our algorithm does not need to set the interesting region and select the key points. In addition, our algorithm belongs to the traditional algorithm, and it needs lower hardware support compared with the deep learning algorithm.

2.2 Feature Coding

Because each action extracts a different number of features, each video is represented using a vector of the same dimension by an encoding algorithm. The common encoding algorithms include bag of feature (BOF) [28], vector of locally aggregated descriptors (VLAD) [29], and fisher vector (FV) [30]. These algorithms have their own characteristics and are widely used in action recognition. We use BOF to implement feature encoding.

2.3 Action Classification

Classifier is very important for action recognition. The common classifiers include naive Bayesian classifier (NBC) [31], logistic regress (LR) [32], support vector machine (SVM) [33], k-nearest neighbor (KNN) [34] and artificial neural network (ANN) [35], etc. For classification, we use the nonlinear SVM classifier. In order to implement multi-class classifier, we use one against rest approach to design the classifier, and select the highest score class as the result.

3 Algorithm Principle

Fig. 1 shows the flow chart of the algorithm in this paper. The blue parts are the main work of this paper. The steps of our algorithm are as follows:

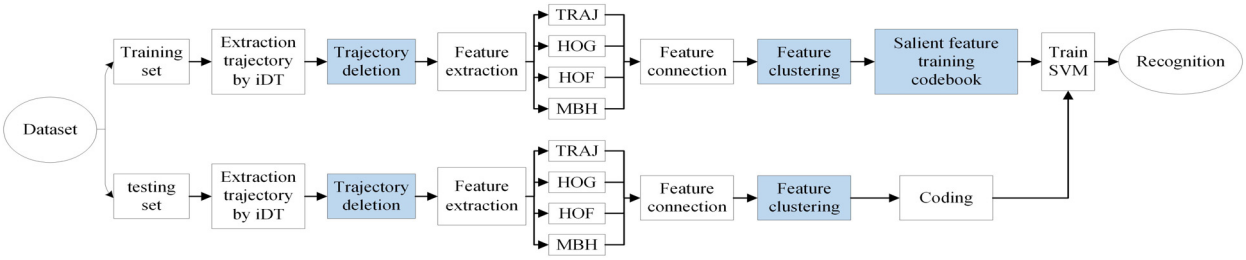

Fig. 1. Flow chart of the algorithm

(1) The iDT is used to extract the trajectories of the input video, and then the trajectories that do not conform to the rule are deleted by trajectory deletion.

(2) TRAJ, HOG, HOF and MBH features are extracted along the trajectory, and they are connected to form the final features.

(3) Cluster all features in each action and use the clustering center to describe the action.

(4) The salient features of the same action category are extracted from the training set, and they are used to train the codebook.

(5) The action is coded by codebook, and then the SVM classifier is trained to complete the classification.

3.1 Trajectory Deletion

Firstly, introduce the feature extraction process of iDT. Fig. 2 shows the flow chart of feature extraction.

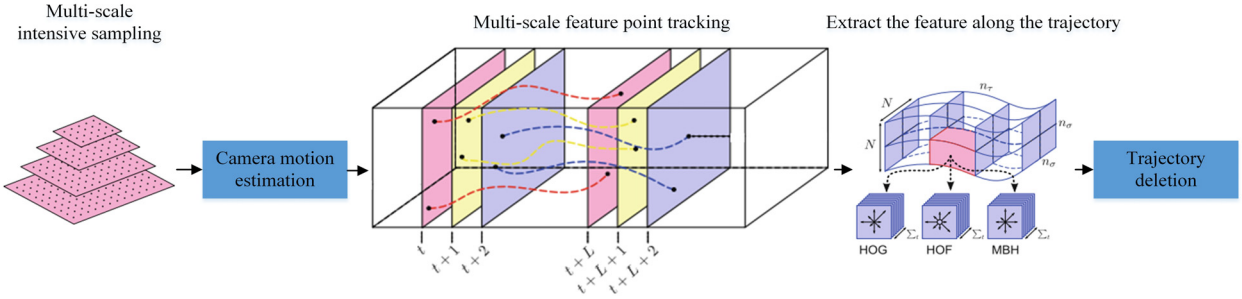

Fig. 2. Flow chart of feature extraction by iDT

At multiple scales of the image, iDT extracts feature points by a grid-based manner. Multiple scale sampling ensures that the feature points cover as many scales and spatial positions as possible. Then, the optical flow in the background area is eliminated by estimating the motion of the camera. Then the motion direction of the feature point is calculated, and the feature points are tracked continuously for L (L=15) frames to form the trajectory. MOG, MOF and MBH are extracted along the trajectory, and TRAJ are obtained by calculating the normalized position vector of the trajectory. Finally, iDT uses the standard deviation and length of each trajectory to determine whether to delete the trajectory.

Since iDT adopts the method of intensive sampling feature points, the extracted trajectories contain static trajectory and random trajectory. Static trajectory means that the trajectory has no obvious displacement, which is caused by the fact that the feature points are not in the action area. Random trajectory means that the change of trajectory is random and is caused by random noise. Since static and random trajectories do not contain action information, both trajectories can be deleted in iDT. However, some of the remaining trajectories can be deleted, and these trajectories only contain a little action information. In this paper, these trajectories are called invalid trajectories. The displacement of the invalid trajectory does not change until the end of the trajectory period, so it has no substantial effect on the recognition.

Fig. 3 shows the points of a trajectory period. In Fig. 3, five points circled by a rectangular box are taken as examples to illustrate the invalid trajectory. The displacement of the five points do not change until the 11th to 15th frames. therefore, an invalid trajectory carries information only in the last few frames of a trajectory period. Since iDT extracts a large number of useful trajectories, the information carried by invalid trajectories could not have a substantial effect on the action recognition. In order to save the storage resources of hardware, trajectory deletion can delete the invalid trajectories. The trajectory deletion is shown as algorithm 1.

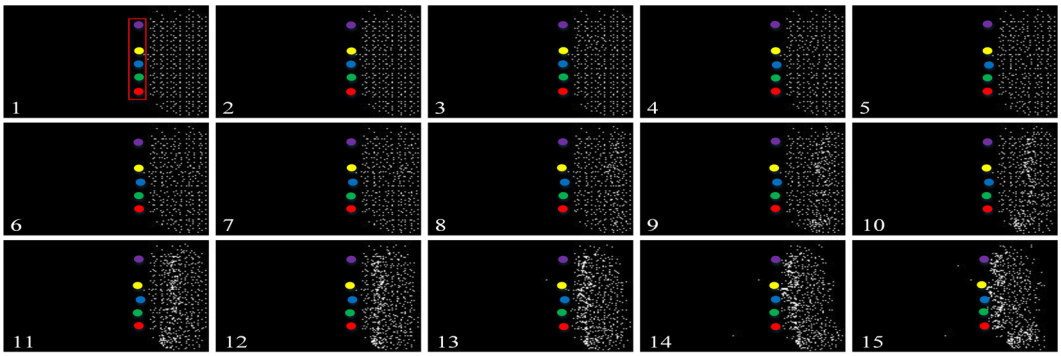

Fig. 3. The points extracted and tracked by the "jogging" action in KTH. In the figure, the points in 1-15 frames of the image are connected to form the trajectory

Algorithm 1:

(1) Calculate the length of the trajectory as L. Calculate the standard deviation $PartVar\_x$ and $PartVar\_y$ in the x and y directions of the first ten points of each trajectory, and calculate the trajectory length of the first ten points as $PartL$.

(2) Define the threshold $PartMin\_var$ of the standard deviation of the first ten points, where $PartMin\_var = 1$.

(3) Delete invalid trajectory: If formula (1) is satisfied, delete the trajectory.

$$PartVar\_x < PartMin\_\text{var} \,\&\, PartVar\_y < PartMin\_\text{var} \,\&\, PartL / L > 0.35 . \quad \textbf{(1)}$$

3.2 Feature Clustering

Fig. 4 shows the trajectories extracted by iDT. Furthermore, the green areas represent the trajectories. It can be found from Fig. 4 that the trajectories extracted by iDT are very dense. Therefore, features will carry plenty of repeated information and will occupy a large amount of hardware storage resources. In order to make features occupy less hardware resources and delete repeated information, this paper uses

Mean-shift to cluster the features of each action, and then the cluster centers are used to represent the action. The advantage of Mean-shift is that there is no need to set the number of cluster centers, which can be adjusted according to the bandwidth.

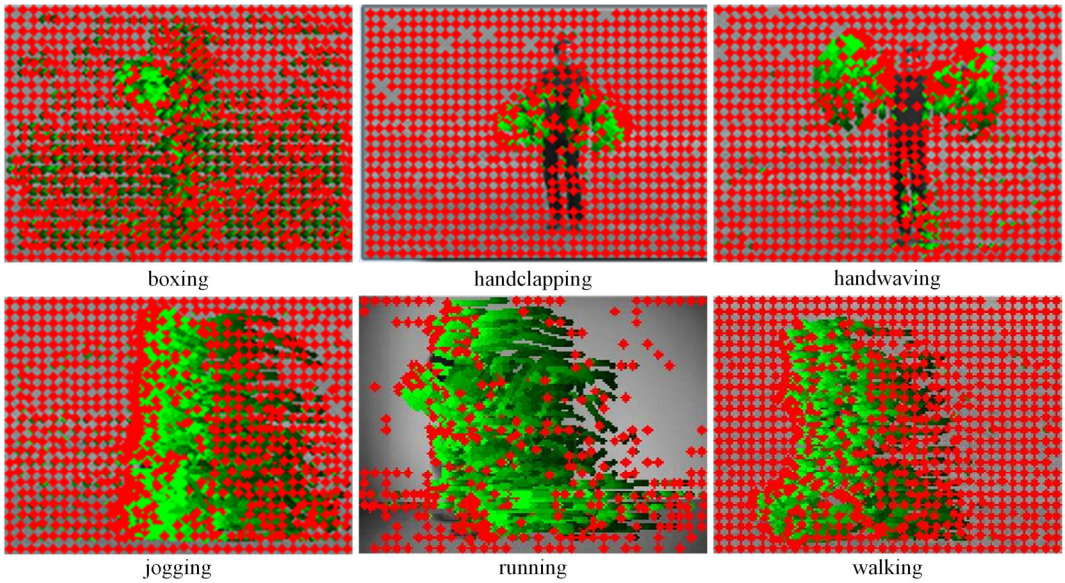

Fig. 4. Trajectory extracted by iDT, and the green area is the motion trajectory (the actions are from KTH dataset)

In this paper, the algorithm extracts Traj (30 dimensions), HOG (96 dimensions), HOF (108 dimensions) and MBH (192 dimensions) features along each trajectory, and then each of features is connected to obtain the final features (426 dimensions). Firstly, features need to L2 normalization, and then the features are clustered using Mean-shift to retain all cluster centers. When searching for the cluster center, we calculate the mean shift vector according to formula (2).

$$M_h(x)=\frac{\sum_{i=1}^{n} G(\| \frac{x_i-x}{h} \|^2)\omega(x_i)(x_i-x)}{\sum_{i=1}^{n} G(\| \frac{x_i-x}{h} \|^2)\omega(x_i)}. \tag{2}$$

Where, $G(\| \frac{x_i-x}{h} \|^2)$ is the kernel function, h is the bandwidth, $\omega(x_i)$ is the sample weight, and n is the number of samples. In this paper, the uniform kernel function is used and the sample weight within the bandwidth is 1, as shown in formula (3). Therefore, formula (3) is substituted into formula (2) to obtain the mean shift vector, which can be expressed by formula (4).

$$G(\| \frac{x_i-x}{h} \|^2)=\begin{cases} 1 & \| \frac{x_i-x}{h} \|^2 \le 1 \\ 0 & else \end{cases}. \tag{3}$$

$$M_h(x)=\frac{1}{k}\sum_{x_i \in S_h} x_i - x. \tag{4}$$

S_h is the high-dimensional sphere with x as the center and h as the radius, and k represents the number of samples in the sphere. In this paper, the clustering process using Mean-shift is shown in algorithm 2.

Algorithm 2:

(1) Enter all features contained in the action.

(2) The mean distance of all features is calculated as *MeanDis*, and the bandwidth of Mean-shift is set

as $h = 0.45 MeanDis$.

(3) The initial feature is randomly selected from the features that are never visited, and the mean shift vector is calculated by formula (4) henceforth iterated continuously. The termination condition of iteration is that the difference between the new clustering center and the old clustering center is less than T, $T = 1 \times 10^{-3} * h$.

(4) Repeat step 3 until all features are visited. If the distance between two cluster centers is less than $h/2$, the two clusters are merged. Finally, all the cluster centers are saved.

Because different actions will extract different number of features. So, feature clustering determines h by calculating the average distance of all features within the action, and the number of cluster centers for each action can be adjusted adaptively according to h. Eventually, all of the cluster centers for each action are retained to represent the action, which reduces the number of features and removes repetitive information.

3.3 Salient Feature Extraction

This paper shows that the classification of action is mainly based on the dissimilar features between different actions, which are called salient features. Fig. 5 is used to illustrate how to extract salient features.

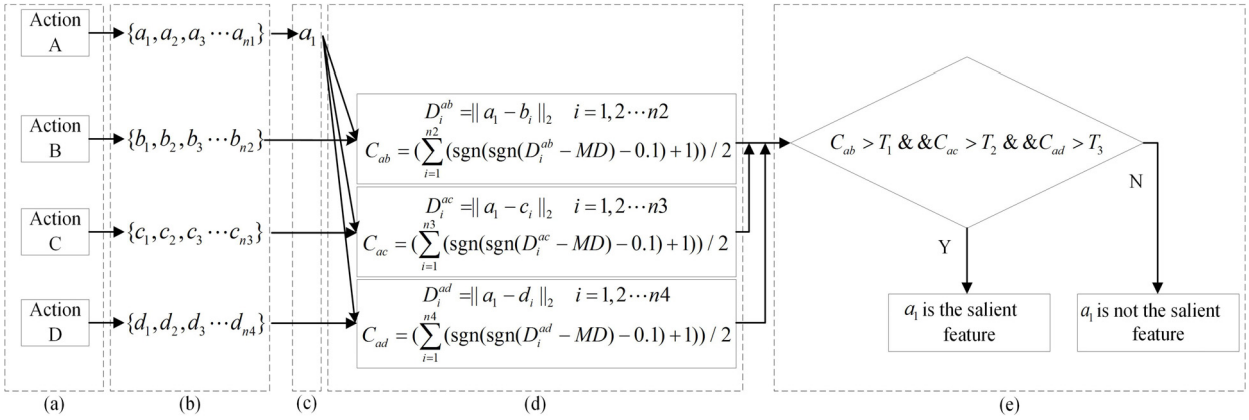

Fig. 5. The process of salient feature extraction

In Fig. 5, suppose the idea that part (a) has four action categories. Part (b) represents the features contained in each action, and the dimension of feature of all actions is consistent. Part (c) refers to the a_1 feature of action A as an example to judge whether it is a salient feature. Part (d) represents the calculated Euclidean distance between a_1 and the features of each actions. Then, determine whether the distance is greater than MD. If the distance is greater than MD, count plus 1, otherwise count plus 0. C_{ab}, C_{ac} and C_{ad} respectively represent the counting results of a_1 in each action category. Part (e) means that when the count result of a_1 is greater than the threshold value, a_1 is a salient feature. Otherwise, a_1 is not a salient feature. Where, MD is the average Euclidean distance of all features within the training set. The threshold value is introduced with T_1 as an example. $T_1 = \alpha * n_2$, α is the coefficient, n_2 is the number of features of action B. Similarly, $T_2 = \alpha * n_3$, $T_3 = \alpha * n_4$. The process of extracting salient features for each action category is to remove the similar features among different action categories and retain salient features (features with large differences). For example, in the KTH, "Jogging" and "Running" actions are quite similar, and there could be more similar features. After extracting the salient features, the similar features of these two action categories can be removed, and the salient features can be kept.

4 Experimental Simulation and Analysis

In the experiment, the experimental environment is i5-3230M CPU, VS2017+OpenCV and Matlab. The datasets used in the experiment are KTH and UCFSports. KTH contains 6 action categories, respectively

"Walking", "Jogging", "Running", "Boxing", "Handwaving" and "Handclapping". Each action is completed by 25 people in four different conditions. The dataset contains 2391 sequences, but background change is simple. The UCFSports dataset contains 10 class actions, including "Diving", "Golf Swing", "Kicking", "Weightlifting", "Riding Horse", "Running", "Skate", "Swing Bench", "Swing Side" and "Walking". UCFSports dataset contains 150 sequences, and a large number of videos come from different websites. The background of the action is relatively complex, and the difference of the actions within the same category is large. So, the UCFSports poses a great challenge to the action recognition. Due to the small number of video sequences in UCFSports, the experiment, like literature [8], uses the horizontal image sequence of each sequence to expand the dataset. The training set and test set of the above two datasets are divided according to the suggestions in the literature.

TD, FC and SFE are abbreviations for trajectory deletion, feature clustering and salient feature extraction, respectively. They are all represented similarly in the rest. Experiments verify the effectiveness of this algorithm from the following four aspects:

(1) TD: Verify that trajectory deletion can effectively delete invalid trajectories.

(2) FC: On the basis of trajectory deletion, it is proved that feature clustering can effectively reduce the number of features and reduce the occupation of hardware storage spaces.

(3) SFE: On the basis of trajectory deletion and feature clustering, the Codebook is trained by salient feature extraction, which can improve the accuracy of action recognition.

(4) TD+FC+SFE: It is proved that the improved iDT by trajectory deletion, feature clustering and salient feature extraction can improve the accuracy of action recognition and can reduce the hardware resource occupation.

Firstly, experiment first verifies the effectiveness of trajectory deletion. Fig. 6 shows the trajectory extracted by iDT, in which the result circled by red rectangle is the result after trajectory deletion. By comparison, it can be seen that a large number of invalid trajectories can be deleted. As can be seen from Table 1, the feature volume is reduced after trajectory deletion. In KTH, the feature volume is decreased by 12.84% ($(V_{\text{iDT+TD}} - V_{\text{iDT}})/V_{\text{iDT}}$, the change of other feature volume is calculated in the same way), and in UCFSports the feature volume is decreased by 2.09%. In terms of accuracy, KTH increased by 0.93% ($P_{\text{iDT+TD}} - P_{\text{iDT}}$, the change of other accuracy is calculated in the same way) and UCFSports decreased by 2.13%. After analysis, it is found that trajectory deletion can effectively delete invalid trajectories. The decrease of accuracy in UCFSports is due to the direction of some actions is almost the same as the direction of video shooting. In two-dimensional coordinate system, this will make the displacement of the trajectory look like one point.

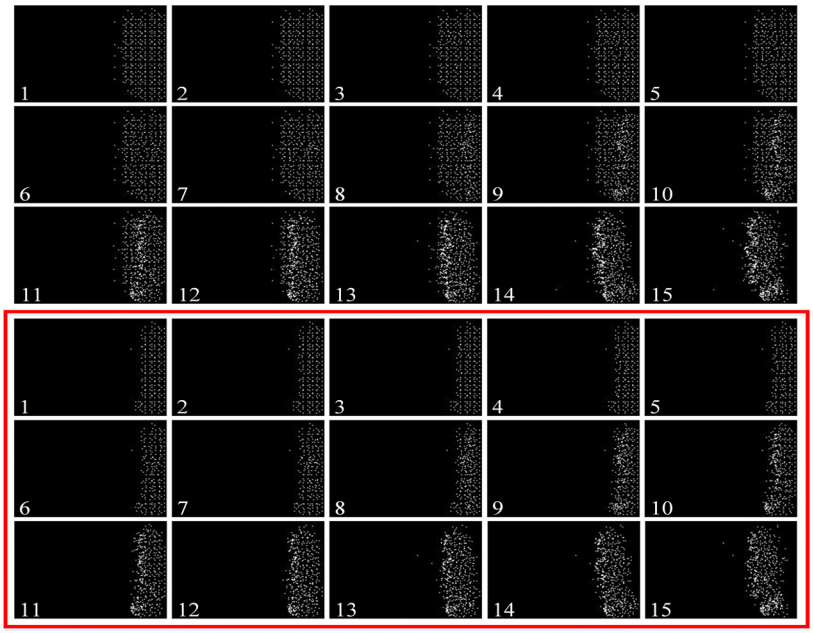

Fig. 6. Feature points are extracted and tracked by iDT, and the feature points in 1-15 frames are connected to form the trajectory (the action is from KTH, the feature points circled by the red rectangle is after trajectory deletion)

Table 1. Comparison of volume and accuracy before and after TD

Dataset	Volume (iDT)	Volume (iDT+TD)	Volume change	Accuracy (iDT)	Accuracy (iDT+TD)	Accuracy change
KTH	17.29G	15.07G	↓ 12.84%	0.949074	0.958333	↑ 0.93%
UCFSports	31.60G	30.94G	↓ 2.09%	0.851064	0.829787	↓ 2.13%

In order to analyze the effect of trajectory deletion in every action category, experiment compares the feature volume of every action category before and after trajectory deletion, as shown in Fig. 7 and Fig. 8. In the KTH, the feature volume of each action category is decreased after trajectory deletion. In Fig. 7, the feature volume is decreased by 12.62%, 14.81%, 17.48%, 6.79%, 13.12% and 10.19% for each action category (the order of category from left to right). Among them, the effect of "Running" is most obvious. In UCFSports, the feature volume of "Golf Swing" and "Skate Boarding" after trajectory deletion is increased slightly, which are 0.91% and 4.85% respectively. The feature volume of "Swing Bench" remains unchanged. In Fig. 8, the feature of remaining action categories is decreased by 0.56%, 2.27%, 9.16%, 1.04%, 0.83%, 0.58% and 8.43% respectively. Among them, the effect of "lifting" is most obvious. Through the analysis, it is found that trajectory deletion has the most obvious effect of invalid trajectory when the action direction is perpendicular to the video shooting direction. This is because when the action direction is perpendicular to the video shooting direction, the displacement of action is the most obvious. This will lead to some feature points that move only at the end of the trajectory period. These feature points are invalid trajectories.

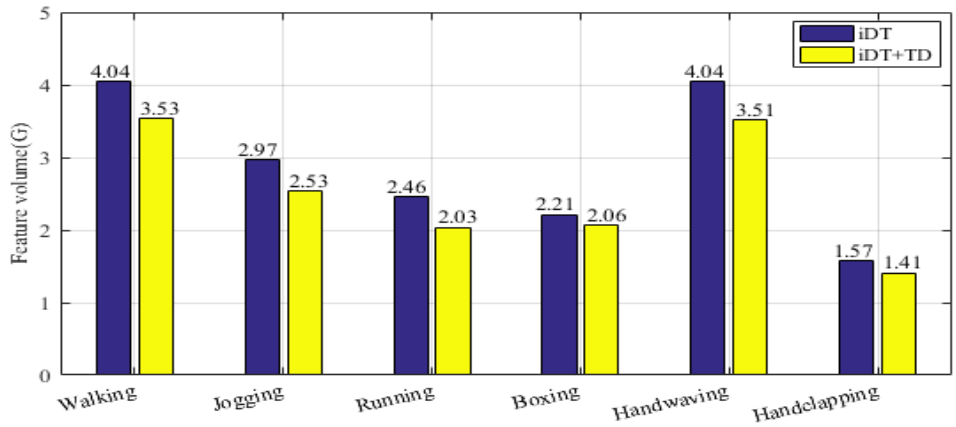

Fig. 7. The feature volume comparison of each action category before and after TD in KTH

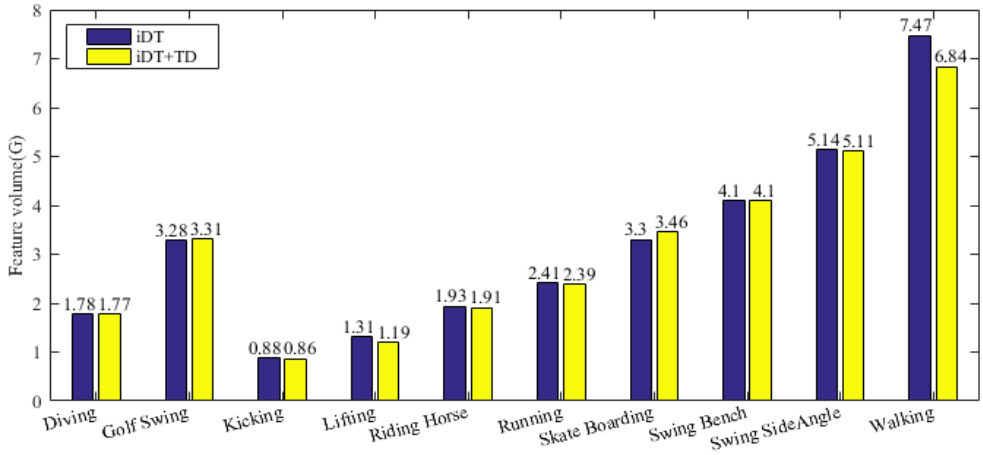

Fig. 8. The feature volume comparison of each action category before and after TD in UCFSports

Secondly, experiment verifies the effectiveness of feature clustering. Table 2 shows the influence of feature clustering on the feature volume and accuracy. As can be seen from Table 2, after feature

clustering, the feature volume is reduced obviously. In KTH, the feature volume is decreased by 77.97%, and in UCFSports the feature volume is decreased by 79.25%. In terms of accuracy, KTH is decreased by 0.93% and UCFSports is increased by 6.38%. Finally, the cluster centers are used to represent each action. Feature clustering is a voting process, and the most representative features are selected. On the basis of significantly reducing the feature volume, feature clustering does not significantly reduce the accuracy, whereas effectively increases the accuracy in UCFSports.

Table 2. Comparison of volume and accuracy before and after FC

Dataset	Volume (iDT+TD)	Volume (iDT+TD+FC)	Volume change	Accuracy (iDT+TD)	Accuracy (iDT+TD+FC)	Accuracy change
KTH	15.07G	3.32G	↓ 77.97%	0. 958333	0.949074	↓ 0.93%
UCFSports	30.94G	6.42G	↓ 79.25%	0. 829787	0.893617	↑ 6.38%

Similarly, in order to analyze the effect of feature clustering in every action category, experiment compares the feature volume before and after feature clustering, as shown in Fig. 9 and Fig. 10. In KTH, the feature volume of each action category is decreased significantly after feature clustering. In Fig. 9, the feature volume is decreased by 80.45%, 88.14%, 86.70%, 54.85%, 83.19% and 61.70% for each action category (the order of category from left to right). In UCFSports, the feature volume of each action category is also decreased significantly after feature clustering. In Fig. 10, the feature volume is decreased by 71.75%, 85.20%, 81.40%, 93.28%, 79.58%, 70.71%, 86.99%, 79.51%, 72.02% and 79.82% for each action category (the order of category from left to right). Feature clustering can extract the cluster center of each action to represent the action. So through the analysis, it can be found that feature clustering has significant effect on each action category, which can greatly reduce the occupation of hardware resources.

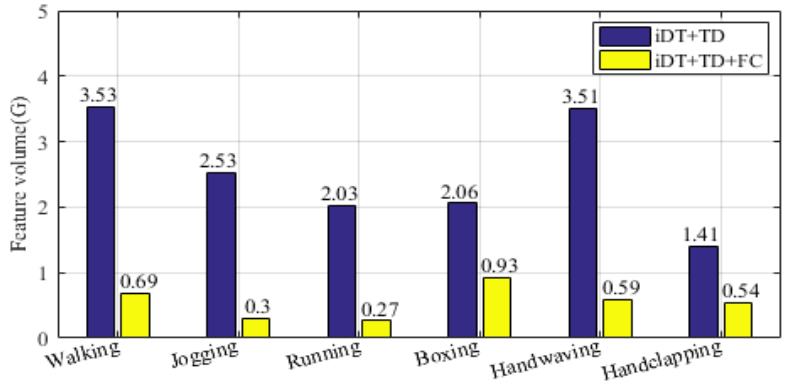

Fig. 9. The feature volume comparison of each action category before and after TD and FC in KTH

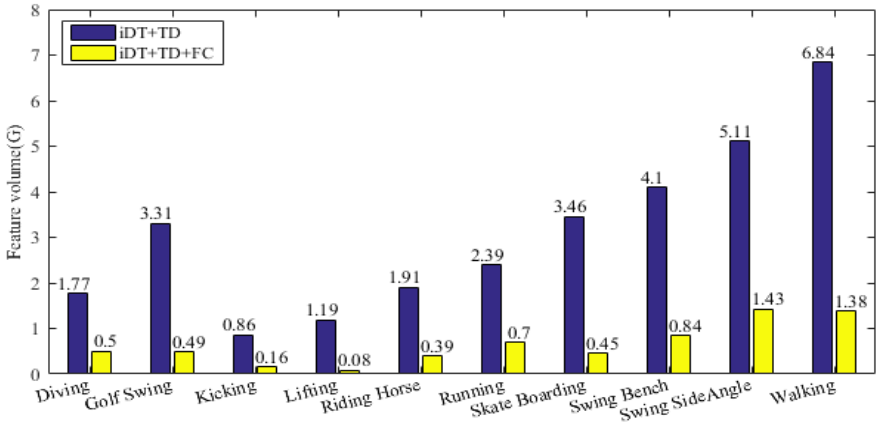

Fig. 10. The feature volume comparison of each action category before and after TD and FC in UCFSports

Thirdly, experiment verifies the effectiveness of salient feature extraction. Salient feature extraction does not change the number of features, so it has no effect on feature volume. As can be seen from Table 3, after salient feature extraction, the accuracy is increased by 0.93% in KTH. In the UCFSports, the accuracy remains constant. It can be seen that the salient feature extraction can improve the accuracy of action recognition to a certain extent.

Table 3. Comparison of volume and accuracy before and after salient feature extraction

Dataset	Volume (iDT+TD+FC)	Volume (iDT+TD+FC+SFE)	Volume change	Accuracy (iDT+TD+FC)	Accuracy (iDT+TD+FC+SFE)	Accuracy change
KTH	3.32G	3.32G	None	0.949074	0.958333	↑ 0.93%
UCFSports	6.42G	6.42G	None	0.893617	0.893617	None

Fourthly, Table 4 shows the improvement of trajectory deletion, feature clustering and salient feature extraction on iDT. As can be seen from Table 4, after improving the iDT, the feature volume in KTH is decreased by 80.80%, and the accuracy is increased by 0.93%. In UCFSports, the feature volume is decreased by 79.68%, and the accuracy is increased by 4.26%. It can be seen that the improved algorithm proposed in this paper can not only reduce the feature volume but also improve the accuracy of action recognition. Fig. 11 shows the confusion matrix of iDT and iDT+TD+FC+SFE in different datasets. The confusion matrix can show the classification result of each action category. In the KTH, "Jogging" is similar to "Running". As can be seen from confusion matrix, after improving the iDT, the recognition of "Jogging" and "Running" is improved. In the UCFSports, the recognition of "Kicking" and "Running" is improved.

Table 4. Comparison of volume and accuracy before and after TD, FC and SFE

Dataset	Volume (iDT)	Volume (iDT+TD+FC+SFE)	Volume change	Accuracy (iDT)	Accuracy (iDT+TD+FC+SFE)	Accuracy change
KTH	17.29G	3.32G	↓ 80.80%	0.949074	0.958333	↑ 0.93%
UCFSports	31.60G	6.42G	↓ 79.68%	0.851064	0.893617	↑ 4.26%

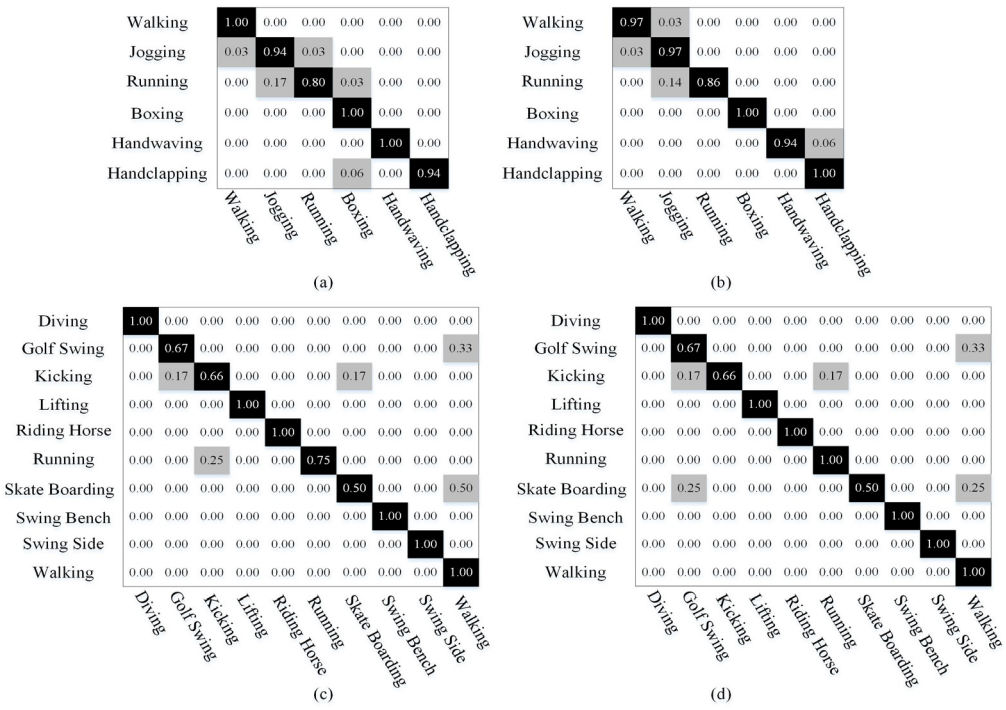

Fig. 11. Confusion matrices of iDT and iDT+TD+FC+SFE in different datasets. (a) confusion matrix of iDT in KTH;(b) confusion matrix of iDT+TD+FC+SFE in KTH; (c) confusion matrix of iDT in UCFSports;(d) confusion matrix of iDT+TD+FC+SFE in UCFSports

In Fig. 12 and Fig. 13, experiment compares the feature volume before and after trajectory deletion, feature clustering and salient feature extraction. From the comparison of the data, it can be seen that the feature volume has been significantly reduced after the improved algorithm. In Fig. 12, the feature volume is decreased by 82.92%, 89.90%, 89.02%, 57.92%, 85.40% and 65.61% for each action category (the order of category from left to right). In Fig. 13, the feature volume is decreased by 71.91%, 85.06%, 81.82%, 93.89%, 79.79%, 70.95%, 86.36%, 79.51%, 72.18% and 81.53% for each action category (the order of category from left to right).

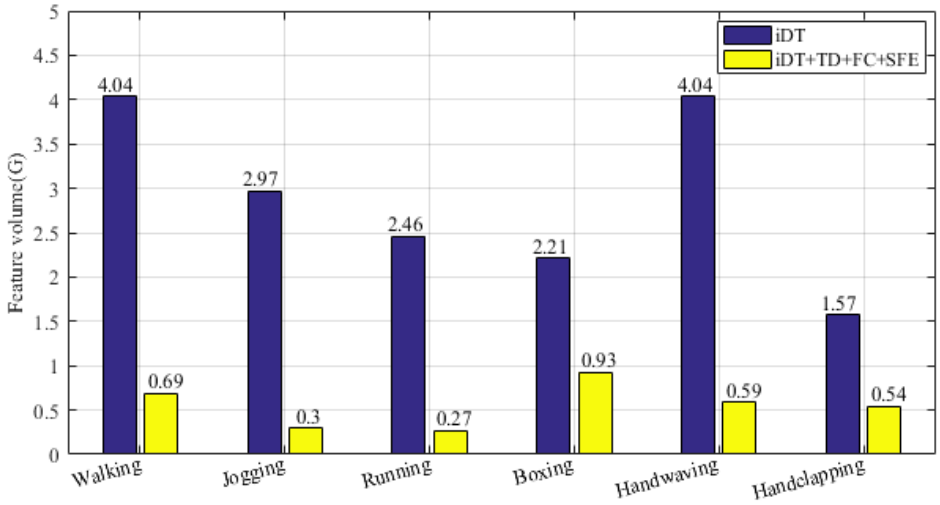

Fig. 12. The volume comparison of each action category before and after TD, FC and SFE in KTH

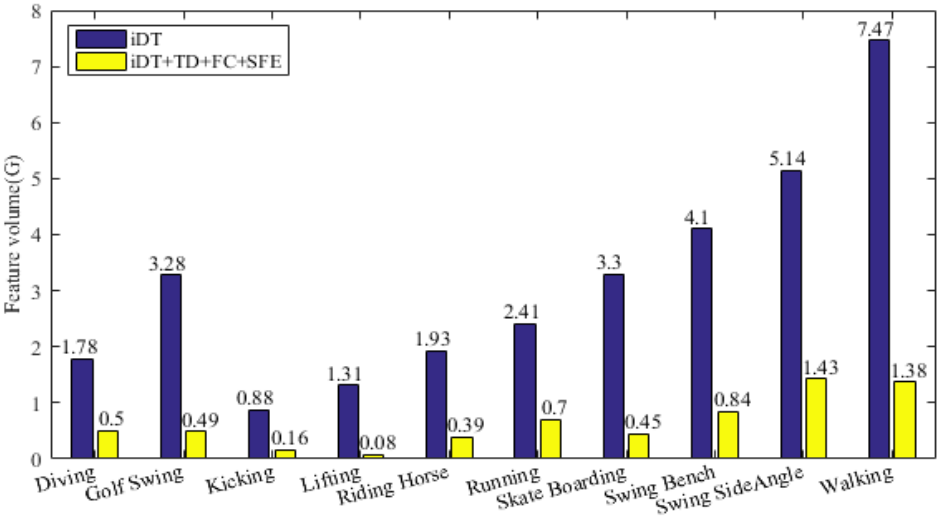

Fig. 13. The volume comparison of each action category before and after TD, FC and SFE in UCFSports

In summary, it can be seen that the improvement of iDT by trajectory deletion, feature clustering and salient feature extraction can significantly reduce the feature volume for each action category.

Finally, iDT+TD+FC+SFE is compared with advanced algorithms, include literature [16-17, 26-27, 36-37]. Literature [36] proposed a STC-HOG (scattering transform coefficients-based histogram of oriented gradients) descriptor to capture more effective motion information, and literature [37] proposed the ReHAR to realize person activities and group activities prediction. Table 5 shows the accuracy of our algorithm and other algorithms. In KTH, the accuracy of our algorithm is 95.8%, which is 1.8%, 0.4% and 0.2% higher than algorithms in the literature [16, 26-27]. In literature [36], the deep learning method is introduced, and its accuracy is 2% higher than our algorithm. However, the deep learning method has many parameters and its parameter optimization is relatively complex. In UCFSports, the accuracy of our algorithm is 89.4%, which is 0.7%, 0.9% and 1% higher than algorithms in the literature [16-17, 26]. The

accuracy of literature [37] is 3.4% higher than our algorithm. However, literature [37] uses LSTM network and takes optical flow image as input. Therefore, the training of this network requires high hardware requirements. Our algorithm uses the feature of manual design to realize action recognition. The training process is simple and the hardware requirement is low.

Table 5. Comparison of iDT+TD+FC+SFE with other advanced algorithms

Algorithm	Accuracy (KTH)/%	Algorithm	Accuracy (UCFSports)/%
Chen YB at al. [16]	94.0	Chen YB at al. [16]	88.7
Lu TR at al. [26]	95.4	Ding ST at al. [17]	88.5
Lu TR at al. [27]	95.6	Lu TR at al. [26]	88.4
Lin B at al. [36]	97.8	X. Li at al. [37]	92.8
iDT+TD+FC+SFE	**95.8**	**iDT+TD+FC+SFE**	**89.4**

Experiment shows that trajectory deletion, feature clustering and salient feature extraction can reduce the feature volume and improve the accuracy of recognition. By comparing with other advanced algorithms, it can be seen that the performance of our algorithm has been significantly improved.

5 Conclusion

In this paper, trajectory deletion, feature clustering and salient feature extraction are proposed to improve the iDT. The algorithm is verified in KTH and UCFSports. In KTH and UCFSports, the experiment results show that the feature volume is reduced by 80.80% and 79.68%, and the accuracy is increased by 0.93% and 4.26%, respectively. In conclusion, the algorithm reduces the number of features and improves the accuracy of action recognition, which effectively reduces the occupation of hardware resources. However, the limitation of this work is that feature clustering and salient feature extraction are two new links in the process of action recognition. These reduce the feature volume, but complicate the process of action recognition.

The ultimate goal of action recognition is to determine what someone is doing at a given moment. To achieve this goal, we still need to overcome several difficulties. These difficulties are also directions for future research, such as:

(1) At present, there is an increasing variety of actions in the published dataset, so the algorithm needs to keep robustness when dealing with multi classification.

(2) Action recognition is not only the identification of individual action, but also includes complex actions such as human to human interaction, human to object interaction and group action. Complex action recognition has gradually become a hot research topic.

(3) Action recognition is affected by the complexity of scene, such as the noise of occlusion and shadow. How to reduce the impact of noise on action recognition is also an important research direction.

(4) Different video shooting angle have an impact on action recognition. There may be large differences in shooting the same action from different angles. Therefore, one of the future research directions is to address the impact of angle on action recognition.

Acknowledgments

This work was supported by the Beijing Jiaotong University Graduate joint training base construction project under No. 275210529109, the National Natural Science Foundation of China under Grant No. 61872027, and free reporting of basic scientific research operating expenses under No. 2020JBM005.

References

[1] M.-E. Zan, H. Zhou, D. Han, G. Yang, G.-L. Xu, Survey of particle filter target tracking algorithms, Computer Engineering and Applications 55(5)(2019) 8-17.

[2] X.-F. Liu, H. Zhou, Q. Han, M.-E. Zan, D. Han, A survey of vision - based gait recognition, Small Microcomputer Systems 39(8)(2018) 1685-1692.

[3] H.-L. Luo, C.-J. Wang, An improved VLAD coding method based on fusion feature in action recognition, Chinese Journal of Electronics 47(1)(2019) 49-58.

[4] H.-Y. Wang, H. Zhou, H.-J. Chen, New edge detection method based on omni-directional and multi-scale mathematical morphology, Minicomputer System 35(5)(2013) 1196-1200.

[5] X.-F. Liu, H. Zhou, Gait recognition technology and application in video surveillance, Video Engineering 35(1)(2011) 1685-1692.

[6] D. Ludl, T. Gulde, C. Curio, Enhancing data-driven algorithms for human pose estimation and action recognition through simulation, IEEE Transactions on Intelligent Transportation Systems 4(29)(2020) 1-10.

[7] H.-L. Luo, C.-J. Wang, F. Lu, Survey of video behavior recognition, Journal of Communications 39(6)(2008) 169-180.

[8] H. Wang, A. Klser, D. Schmid, C.-L. Liu, Dense trajectories and motion boundary descriptors for action recognition, International Journal of Computer Vision 103(1)(2013) 60-79.

[9] H. Wang, C. Schmid, Action recognition with improved trajectories, in: Proc. IEEE International Conference on Computer Vision, 2013.

[10] A.-F. Bobick, J.-W. Davis, The recognition of human movement using temporal templates, IEEE Transactions on Pattern Analysis and Machine Intelligence 23(3)(2001) 257-267.

[11] A. Yilmaz, M. Shah, Actions as objects: a nover action representation, in: Proc. Computer Vision and Pattern Recognition, 2005.

[12] L. Gorelick, M. Blank, E. Shechtman, E.-M. Irani, R. Basri, Actions as space-time shapes, IEEE Transactions on Pattern Analysis and Machine Intelligence 29(12)(2007) 2247-2253.

[13] H.-J. Seo, P. Milanfar, Action recognition from one example, IEEE Transactions on Pattern Analysis & Machine Intelligence 33(5)(2011) 867-882.

[14] J. Feng, C.-J. Ai, Z. An, Z.-Q. Zhou, Y.-S. Shi, A feature detection and matching algorithm based on harris algorithm, in: Proc. 2019 International Conference on Communications Information System and Computer Engineering, 2019.

[15] Y. Wang, Y. Yuan, Z. Lei, Fast sift feature matching algorithm based on geometric transformation, IEEE Access 8(4)(2020) 88133-88140.

[16] Y.-B. Chen, Z.-X. Li, X. Guo, Y.-Y. Zhao, A.-N. Cai, A spatio-temporal interest point detector based on vorticity for action recognition, in: Proc. Multimedia and Expo Workshops, 2013.

[17] S.-T. Ding, S.-R. Qu, Human action recognition algorithm based on improved spatio-temporal interest point detection, Journal of Northwestern Polytechnical University 34(5)(2016) 886-892.

[18] D.-J. Moore, I.-A. Essa, M. H. I. Hayes, Exploiting human actions and object context for recognition tasks, in: Proc. IEEE International Conference on Computer Vision, 1999.

[19] Z. Lan, M. Lin, X. Li, A.-G. Hauptmann, B. Raj, Beyond gaussian pyramid: multi-skip feature stacking for action recognition, in: Proc. Computer Vision and Pattern Recognition, 2015.

[20] H. Wang, L. Wang, Cross-agent action recognition, IEEE Transactions on Circuits & Systems for Video Technology 28(10)(2018) 2098-2919.

[21] F. Camarena, L. Chang, M. Gonzalez-Mendoza, Improving the dense trajectories approach towards efficient recognition of simple human activities, in: Proc. International Workshop on Biometrics and Forensics, 2019.

[22] X. Liang, H.-B. Zhang, Y.-X. Zhang, H.-B. Zhang, JTCR: joint trajectory character recognition for human action recognition, in: Proc. 2019 IEEE Eurasia Conference on IOT Communication and Engineering, 2019.

[23] L. Wang, Y. Qiao, X. Tang, Action recognition with trajectory-pooled deep-convolutional descriptors, in: Proc. Computer Vision and Pattern Recognition, 2015.

[24] Y. Zhao, Y. Xiong, D. Lin, Trajectory convolution for action recognition, in: Proc. Neural Information Processing Systems, 2018.

[25] X. Xiao, H. Hu, W. Wang, Trajectories-based motion neighborhood feature for human action recognition, in: Proc. International Conference on Image Processing, 2017.

[26] T.-R. Lu, F.-Q. Yu, H.-Z. Yang, Y. Chen, Human action recognition based on dense trajectories with saliency detection, Computer Engineering and Applications, 54(14)(2018) 163-167.

[27] T.-R. Lu, F.-Q. Yu, Y. Chen, A human action recognition method based on lsda dimension reduction, Computer Engineering 45(3)(2019) 237-241.

[28] C. Tsai, Two strategies for bag-of-visual words feature extraction, in: Proc. International Congress on Advanced Applied Informatics, 2018.

[29] W.-W. You, J.-Q. Guo, K. Shan, Y.-Z. Dai, A novel trajectory-vlad based action recognition algorithm for video analysis, Procedia Computer Science 147(5)(2019) 165-171.

[30] L.-P. Fang, N.-G. Liang, W.-X. Kang, Z.-Y. Wang, D.-D. Feng, Real-time hand posture recognition using hand geometric features and fisher vector, Signal Processing Image Communication 82(3)(2020) 1-13.

[31] J.-W. Yu, P.-Y. Ping, L. Wang, L.-A. Kuang, X.-Y. Li, Z.-L. Wu, A novel probability model for lncrna-disease association prediction based on the naïve bayesian classifier, Genes 9(7)(2018) 1-21.

[32] W.-F. Liu, H.-L. Liu, T.-P. Tao, Y.-J. Wang, K. Lu, Multiview hessian regularized logistic regression for action recognition, Signal Processing 110(5)(2015) 101-107.

[33] F.-P. An, Human action recognition algorithm based on adaptive initialization of deep learning model parameters and support vector machine, IEEE Access 6(10)(2018) 59405-5942.

[34] M. Ehatisham-ul-haq, A. Javed, M.-A. Azam, H. M. A. Malik, A. Irtaza, I.-H. Lee, M.-T. Mahmood, Robust human activity recognition using multimodal feature-level fusion, IEEE Access 7(4)(2019) 60736-60751.

[35] W.-W. Myo, W. Wettayaprasit, P. Aiyarak, Designing classifier for human activity recognition using artificial neural network, in: Proc. 2019 IEEE 4th International Conference on Computer and Communication Systems, 2019.

[36] B. Lin, B. Fang, W.-B. Yang, J.-Y. Qian, Human action recognition based on spatio-temporal three-dimensional scattering transform descriptor and an improved vlad feature encoding algorithm, Neurocomputing 348(5)(2018) 145-157.

[37] X. Li, M.-C. Chuah, Rehar: robust and efficient human activity recognition, in: Proc. 2018 IEEE Winter Conference on Applications of Computer Vision, 2018.