Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 4358728, 18 pages
https://doi.org/10.1155/2020/4358728

*Research Article*

# Using a Multilearner to Fuse Multimodal Features for Human Action Recognition

**Chao Tang**,[1] **Huosheng Hu**,[2] **Wenjian Wang**,[3] **Wei Li**,[4] **Hua Peng**,[5,6] **and Xiaofeng Wang**[1]

[1]*School of Artificial Intelligence and Big Data, Hefei University, Hefei 230601, China*
[2]*School of Computer Science and Electronic Engineering, University of Essex, Colchester CO4 3SQ, UK*
[3]*School of Computer and Information Science, Shanxi University, Taiyuan 030006, China*
[4]*School of Computer and Information Engineering, Xiamen University of Technology, Xiamen 361024, China*
[5]*Department of Computer Science and Engineering, Shaoxing University, Shaoxing 312000, China*
[6]*College of Information Science and Engineering, Jishou University, Jishou 416000, China*

Correspondence should be addressed to Chao Tang; tangchao77@sina.com

Received 23 March 2020; Revised 6 July 2020; Accepted 1 August 2020; Published 27 August 2020

Academic Editor: Marek Lefik



The representation and selection of action features directly affect the recognition effect of human action recognition methods. Single feature is often affected by human appearance, environment, camera settings, and other factors. Aiming at the problem that the existing multimodal feature fusion methods cannot effectively measure the contribution of different features, this paper proposed a human action recognition method based on RGB-D image features, which makes full use of the multimodal information provided by RGB-D sensors to extract effective human action features. In this paper, three kinds of human action features with different modal information are proposed: RGB-HOG feature based on RGB image information, which has good geometric scale invariance; D-STIP feature based on depth image, which maintains the dynamic characteristics of human motion and has local invariance; and S-JRPF feature-based skeleton information, which has good ability to describe motion space structure. At the same time, multiple K-nearest neighbor classifiers with better generalization ability are used to integrate decision-making classification. The experimental results show that the algorithm achieves ideal recognition results on the public G3D and CAD60 datasets.

## 1. Introduction

Human action recognition is an interdisciplinary research direction in the field of computer vision, involving image processing, computer vision, pattern recognition, machine learning, and artificial intelligence. With the rapid development of digital image processing technology and intelligent hardware manufacturing technology, human action recognition has wide application prospects in intelligent video monitoring [1–4], natural human computer interaction [5, 6], smart home products [7–9], and virtual reality [10]. The popularity of human action recognition has led to several survey articles that have appeared in refs [11–15]. These articles discuss various features and classifiers that have been used for human action recognition. In recent decades, computer vision research based on RGB image information is more and more abundant. However, RGB images usually provide only the apparent information of objects in the scene. When the foreground and background of an RGB image are similar in texture or color, it is difficult to perform accurate image recognition when relying on the limited RGB information. In addition, the appearance of the object described in the RGB image may not be robust to the common visual changes, such as illumination changes, which seriously hinder the use of the RGB-based visual algorithms in the real-world application environment.

With the continuous progress of science, Microsoft has released the Kinect sensor, which provides RGB information, scene depth information, and also human skeletal information in the scene. The depth image information is

only related to the distance between the object and the camera and is not affected by illumination variation, environmental changes, and shadows. The human action sequence, in the form of multimodal sensor data, contains rich temporal patterns that can be used to distinguish between different action categories.

This paper makes full use of the multimodal information provided by the Kinect sensor to extract effective human action features and uses a multilearner integration strategy based on the K-nearest neighbor algorithm to construct a classification model.

The main contributions of this article are as follows:

(1) The RGB modal information, based the histogram of oriented gradient (RGB-HOG), can maintain a good invariance to both geometric and optical deformation. The depth modal information, based on the space-time interest points (D-STIP), can keep dynamic stability of a human action feature, which maintains good local invariance characteristics of human movement. The skeleton modal information based on the joints' relative position feature (S-JRPF) can describe the spatial structure information of human action well. Three different modal features can effectively represent human behavior and provide reliable behavior representation.

(2) This work uses a multilearner ensemble to classify the prediction samples and makes full use of the learning biases of different learners to enhance the generalization ability of the overall model.

The rest of this paper is organized as follows. Section 2 presents the related works. Section 3 describes method framework of human action recognition. In Section 4, three different behavioral descriptors are introduced. We introduce the human action recognition algorithm in Section 5. Experimental results are given in Section 6 to verify the feasibility and performance of the proposed method. Finally, a brief conclusion and the future work are given in Section 7.

## 2. Related Works

Although there have been many achievements in the research of action recognition, human action recognition in a real environment remains difficult. Video-based human action recognition can be divided into RGB data and RGB-D data-based human action recognition. Compared with RGB-D data, RGB data have more abundant appearance information and can better describe the interaction between human and object. However, RGB data are easily affected by background image, such as weather, light, shooting angle, and clothing, which makes it difficult to extract features from background image. Compared with the traditional RGB data, the RGB-D data are not affected by the change of illumination and the change of color and texture. More importantly, they can estimate the contour and skeleton of human body reliably.

Recently, with the development of RGB-D cameras, especially the Kinect sensor launched by Microsoft, recent research has focused on the use of deep images to solve the problem. Compared with traditional RGB data, the depth information provided by RGB-D images is more robust to changes in lighting conditions. The ever-growing popularity of the Kinect inertial sensors has prompted intensive research efforts on human action recognition. Since human actions are extracted from Kinect and inertial sensors, they can be characterized by multiple feature representations. By encoding the multiview features into a unified space, richer data are available for human action recognition.

In recent years, human action recognition based on video has made great progress. Many scholars have summarized and analyzed human action recognition methods based on RGB-D data [16, 17]. According to the different data, the method of human action recognition based on depth sensor can be divided into three parts: depth image sequence-based method, skeleton data-based method, and multimodal feature fusion-based method.

*2.1. Depth Image Sequence-Based Method.* In RGB-D video, depth data can be regarded as a spatiotemporal structure composed of depth information. The feature representation of action is the process of extracting features from this spatiotemporal structure. The method based on depth sequence mainly uses the action changes in the depth map of human body to describe the action. Sahoo et al. [18] applied depth history image to AlexNet to fine-tune the weights of the pretrained deep leaning architecture. To recognize the closely related actions, DHI alone is not sufficient. The 3D projected planes are extracted and trained separately on AlexNet for this purpose. Two types of projected planes are extracted in this work such as XT plane or side view and YT plane or top view of the action videos. The scores from both the learning techniques are fused to provide the final recognition score. Li et al. [19] proposed a real-time human action recognition system that uses depth map sequence as input. The system contains the segmentation of human, the action modeling based on 3D shape context, and the action graph algorithm. Xu et al. [20] proposed an effective method for human action recognition from depth images. A multilevel frame select sampling (MFSS) method is proposed to generate three levels of temporal samples from the input depth sequences first. Then, the proposed motion and static mapping (MSM) method is used to obtain the representation of MFSS sequences. After that, this paper exploits the block-based LBP feature extraction approach to extract feature information from the MSM. Finally, the fisher kernel representation is applied to aggregate the block features, which is then combined with the kernel-based extreme learning machine classifier. Chen et al. [21] proposed a human action recognition method by using depth motion maps (DMMs). Each depth frame in a depth video sequence is projected onto three orthogonal Cartesian planes. Under each projection view, the absolute difference between two consecutive projected maps is accumulated through an entire depth video sequence forming a DMM. An l2-regularized collaborative representation classifier with a distance-weighted Tikhonov matrix is then employed for action recognition. The developed method is shown to be computationally

efficient allowing it to run in real time. The above methods identify actions by analyzing and modeling the motion information in the depth sequence. However, because RGB-D video itself has more noise and lacks relevant appearance and texture information, the depth sequence-based method has not achieved ideal results in many datasets.

*2.2. Skeleton Data-Based Method.* The method of action recognition based on skeleton data is an important direction in the field of depth data research. Based on the skeleton sequence of the human body, this method uses the changes of human joints between video frames to describe the movement, including the changes of joint position and appearance. The skeleton model of the human body can be quickly and accurately estimated from the depth data, so the method of human posture estimation based on RGB-D data is widely used. Wan et al. [22] extracted the orientation vectors from several groups of skeleton joints and used a stacked residual bidirectional long-short term memory (LSTM) network to build modal. Liu et al. [23] proposed a new action recognition LSTM network based on skeleton data, that is, global context aware attention LSTM network. By using the global context memory unit, the network can selectively focus on the information nodes in each frame. In order to further improve the attention ability of the network, a recursive attention mechanism is introduced, through which the attention performance of the network can be gradually improved. Liu et al. [24] proposed a method of human motion recognition based on the skeleton data collected by depth sensor. In order to make full use of the skeleton data of human body, the movement features such as position, speed, and acceleration are extracted from each frame to capture the dynamic and static information of human action. Finally, k-nearest neighbor algorithm based on weighted voting method is used to realize action recognition, and pose specificity is used as voting weight. Phyo et al. [25] used the skeleton motion history image to build a deep learning model to recognize human behavior. The experimental results show that this method can achieve high recognition accuracy with low calculation cost in all kinds of environments. Because the skeleton information is not affected by background light and other factors, it has certain robustness and can be quickly and accurately estimated from the depth data. In recent years, with the development of deep learning, the application of convolutional neural network (CNN), recurrent neural network (RNN), LTSM, and other frameworks has brought progress to the skeleton-based motion recognition, which will make greater progress in the future.

*2.3. Multimodal Feature Fusion-Based Method.* Each feature extraction method has its own advantages and is independent of each other. If different features can be fused effectively, a more discriminative feature vector can be obtained, and the recognition performance will be improved. Therefore, in recent years, the fusion method has attracted the attention of scholars. There are two kinds of fusion methods: feature level fusion and decision level fusion.

Feature lever fusion is an early fusion method. Firstly, the feature vectors are extracted by different methods, and then the extracted features are standardized, selected, or transformed, so as to generate a new feature vector with more discrimination. Zhang et al. [26] proposed a method of action recognition which combines gradient information and sparse coding. Firstly, the feature of coarse depth skeleton is extracted by using depth gradient information and skeleton joint distance. Then, sparse coding and maximum pool are combined to refine the rough coarse depth skeleton features. Finally, the random decision forests are used to identify the actions. El Din El Madany et al. [27] proposed a human action recognition framework by using global locality that preserves canonical correlation analysis (GLPCCA); their work fuses depth and RGB modalities, which includes the hierarchical pyramid of depth motion map deep convolutional neural network (HP-DMM-CNN) used for the depth images and the optical flow convolutional neural network to model the RGB videos. Guo et al. [28] proposed a new unsupervised feature fusion method for human action recognition, termed the multiview Cauchy estimator feature embedding (MCEFE). By minimizing empirical risk, MCEFE integrates the encoded complementary information in multiple views to find the unified data representation and the projection matrices. To enhance robustness to outliers, the Cauchy estimator is imposed on the reconstruction error. Asteriadis et al. [29] presented a novel, multimodal human action recognition method to handle a sensing device's noise and person-specific characteristics. Each action is represented by a basis vector and spectral analysis is performed on an affinity matrix of new action feature vectors. Using modality-dependent kernel regressors for computing the affinity matrix, the complexity is reduced by forming robust low dimensional representations. Gao et al. [30] proposed pyramid appearance and global structure action descriptors on both RGB and depth motion history images as a way to construct a model-free method for human action recognition. In this algorithm, they first construct a motion history image for both the RGB and depth channels while simultaneously depth information is employed to filter RGB information; next, different action descriptors are extracted from the depth and RGB MHIs to represent these actions, and then a multimodality information collaborative representation and recognition model is built in which multimodality data are put into an objective function naturally. In this method, information fusion and action recognition are done together, with the goal to classify human actions.

Decision level fusion is different from feature level fusion. First, the classifier trained by each method outputs the classification results, and then the classification results are fused to get the final classification results. In order to effectively combine the joint, RGB, and depth information of Kinect sensor, Seddik et al. [31] proposed local and global support vector machine model using multilayer fusion scheme to connect different features. Malawski and Kwolek [32] proposed a new motion description called joint motion history context, which is based on depth and bone data. The decision level fusion method based on support vector machine and multilayer perceptron is used to effectively fuse the motion mode information of multiple feature sets. Imran and Raman [33] proposed a multimodal action recognition method based on deep learning paradigm. Firstly, for RGB video, a new image-based descriptor is proposed, which is called stacked dense flow difference image (SDFDI), which can capture the temporal and spatial information in video sequence. Then, they train various kinds of deep two-dimensional CNN and compare SDFDI with the latest image-based representation. Secondly, aiming at skeleton flow, a data enhancement technology based on 3D transformation is proposed to train deep neural network on small dataset. A RNN model based on bidirectional gating recursive unit (BiGRU) is proposed. Thirdly, for the inertial sensor data, a data enhancement method based on Gaussian white noise jitter is proposed, and the action classification is combined with the deep one-dimensional CNN network. The outputs of these three heterogeneous networks are combined by multiple model fusion methods based on fraction and feature fusion.

Although the existing action recognition method using depth information has made great progress, the reliability of recognition is still unsatisfactory for practical engineering. The primary reason is that human action recognition has great within-class differences but nonobvious between-class differences, and distinguishing the differences of human movement speed requires higher computational complexity.

## 3. Method Framework

In order to improve the robustness and practicability of the recognition system and to make full use of the advantages of different features, we use the different modality data provided by the Kinect sensor. Three kinds of features are used as human action descriptors, and then the multilearner ensemble algorithm is used to recognize the action. The system flow is shown in Figure 1. This method preserves the efficiency of performing computation on simple features and also guarantees robustness of the recognition system and the discrimination ability of the action feature. The system framework includes the following steps:

(1) Obtain synchronous RGB image, depth image, and skeleton data from the Kinect sensor.

(2) Transform the RGB image data to gray image data to reduce the scale of data processing, use classical filtering methods to reduce image noise, and then extract a histogram of oriented gradients from the processed image. Space-time interest points are extracted as features from the depth image data, and data describing the relative position of joints from the 3D skeleton data are also extracted as features.

(3) Classify the action described in the above three features using three different k-nearest neighbor classifiers $\mathrm{KNN}_i\,(i = 1, 2, 3)$ based on three different distance measurement formulas. Select the $\mathrm{Top}k_i\,(i = 1, 2, 3)$ actions with highest similarity results from each classifier $\mathrm{KNN}_i\,(i = 1, 2, 3)$ and set the action class that has the largest number of samples in $\{\mathrm{top}\_k_1, \mathrm{top} + k_2, \mathrm{top}\_k_3\}$ as the final prediction result.

## 4. Feature Extraction

The output of Microsoft Kinect camera is a multimodal signal, which can provide RGB video, depth mapping image sequence, and skeleton joint information at the same time. Thus, it can effectively overcome the loss of depth information and spatial position relationship between objects due to the traditional RGB camera projecting the 3D physical world onto the 2D image plane. The characteristics of different modes are independent but complementary. In order to obtain better recognition performance, this paper effectively fuses the features under multimodality and designs a description vector with high discriminability, that is, using visual information, the depth, and skeleton to improve the recognition results. In this section, three different behavioral descriptors are introduced.

*4.1. RGB-HOG.* Histogram of oriented gradients (HOG) is a feature descriptor for object detection in computer vision and image processing [34]. HOG descriptors can effectively extract the local gradient and direction information of the image to describe the key characteristics of human behavior. The traditional HOG feature extraction process is a pyramid structure, which consists of three layers: cell, block, and image. The top and bottom steps are as follows: (1) construct the feature vector of cell; (2) construct the feature vector of block; and (3) construct the feature vector of image. In the process of constructing cell histogram of traditional HOG operator, the influence of neighborhood pixel gradient is not considered, so the "aliasing effect" is easy to appear. To solve this problem, Dalal et al. [34] used the block overlap method, but the calculation is large; Pang et al. [35] used the linear interpolation method to adjust the voting rights of the pixels in the block, but it does not consider the influence of the pixels in the block neighborhood. In fact, based on the cell, only part of the gradient information of its neighborhood is used, which leads to the problem of insufficient information utilization. In this paper, based on the cell, the neighborhood range of cell is planned, and the voting method of neighborhood pixels is further improved. The histogram of the

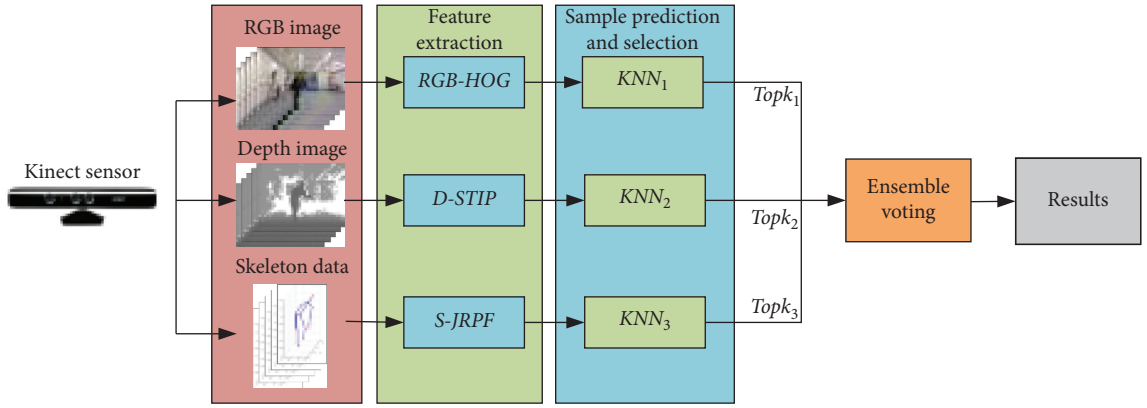


Figure 1: Frame diagram of human action recognition.

original cell is modified by the gradient amplitude of all pixels in the neighborhood of the cell. The HOG feature extraction algorithm flow is shown in Figure 2.

Step 1: input image and region of interest extraction.

In the research of human behavior recognition, the region of interest (ROI) is selected as a smaller region from an image. This region is the most important part of human motion analysis. The region can be cropped from the full-size image to reduce processing time and increase accuracy. In this paper, first an input image $G(x, y)$ is analyzed using a region of interest detection algorithm to predict the approximate position of the target and to select the minimum rectangular boundary around the target as the region of interest. Next, a series of operations is carried out, including feature extraction in the ROI corresponding to the original image.

$$F(x, y) \longleftarrow \mathrm{ROI}(G(x, y)). \tag{1}$$

Step 2: image graying and gamma correction.

Due to the varied factors of image acquisition devices and environments, image of faces may be unclear and prone to either failed detection or false detection. Consequently, it is necessary to preprocess the collected human image, mainly to deal with the situations where the image is either not luminous enough (too dark) or too luminous (too light). There are two processes used to deal with this issue: image graying and gamma correction.

(a) Image graying: for a color image, the RGB component is converted into a grayscale image. The conversion formula is as follows:

$$F(x, y) \longleftarrow \mathrm{RGB2Gray}(F(x, y)). \tag{2}$$

(b) Gamma correction: in the case of uneven illumination, gamma correction can be used to improve or reduce the overall brightness of the image. In practice, we can use two different methods to standardize gamma, employing either the square root or logarithm. In this paper, we use the square root method. The formula is as follows (where $\gamma = 0.5$):

$$I(x, y) \longleftarrow F(x, y)^{\gamma}. \tag{3}$$

Step 3: gradient calculation.

For the normalized image, the gradient and gradient direction are obtained via the following equations:

$$\begin{aligned} I_x(x, y) &= I(x + 1, y) - I(x - 1, y), \\ I_y(x, y) &= I(x, y + 1) - I(x, y - 1), \\ |\nabla I(x, y)| &= \sqrt{I_x^2(x, y) + I_y^2(x, y)}, \\ \Phi(x, y) &= \arctan^{-1}\frac{I_y(x, y)}{I_x(x, y)}. \end{aligned} \tag{4}$$

Step 4: histogram of oriented gradients.

The gradient direction image $\Phi(x, y)$ is divided into $N$ cells, with $8 \times 8 = 64$ pixels as one cell. Adjacent cells do not overlap. The gradient direction of each pixel is counted in each cell. All the gradient directions are divided into 9 bins (i.e., 9-d eigenvectors) as the horizontal axis of histogram, and the cumulative value of gradient value corresponding to the angle range is the vertical axis of histogram.

$$\begin{aligned} &\mathrm{Cell}_i(i = 1, 2, \ldots, N) \longleftarrow \nabla I(x, y), \\ &\mathrm{Cellhog}_i(i = 1, 2, \ldots, N) \longleftarrow \mathrm{BinCount}(\mathrm{cell}_i(i = 1, 2, \ldots, N)). \end{aligned} \tag{5}$$

Then, the original histogram vector value is modified. Suppose $\mathrm{Cell}_i$ is any cell and $M_1$ is any pixel in its neighborhood. The size of $\mathrm{Cell}_i$ area is $d \times d$, and the coordinate of the middle point is $(x_i, y_i)$. The coordinates of the pixel $M_1$ are $(x, y, \theta)$, where $\theta$ is the gradient direction value of $M_1$ and the gradient amplitude is $G(x, y)$. It is assumed that $\theta$ lies between the direction blocks $\theta_l$ and $\theta_r$ of $\mathrm{Cell}_i$. Let the correction coefficients of $M_1$ to the histogram of $\theta_l$ and $\theta_r$ direction block be $w_l$ and $w_r$, respectively. The original

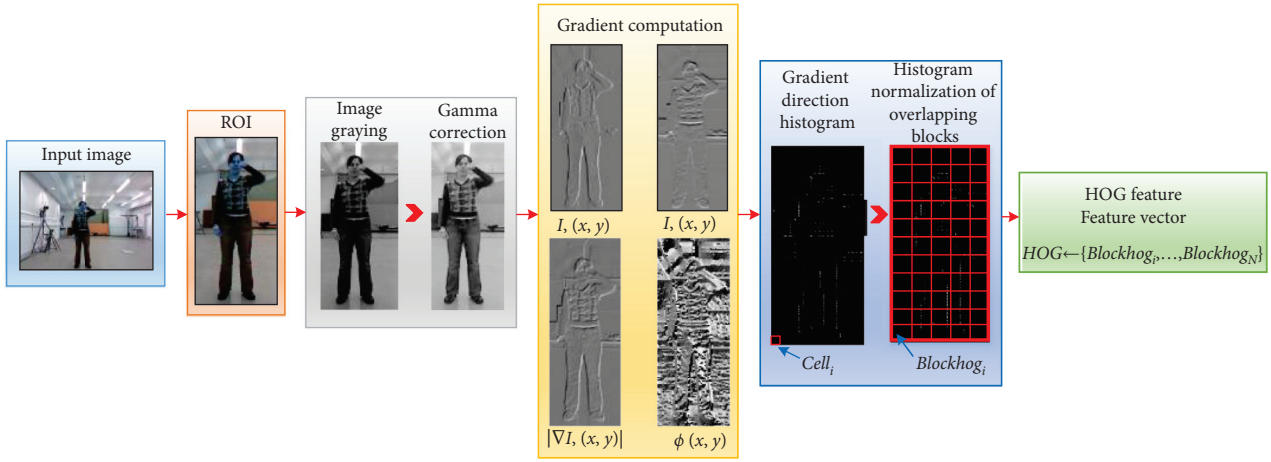


Figure 2: Frame diagram of HOG feature extraction.

histogram vector values of $\theta_l$ and $\theta_r$ are $h(x_i, y_i, \theta_l)$, $h(x_i, y_i, \theta_r)$. The trilinear interpolation method is used for correction, and the correction coefficients $w_l$ and $w_r$ are

$$\begin{cases} w_l = \left(1 - \dfrac{x - x_i}{d}\right)\left(1 - \dfrac{y - y_i}{d}\right)\left(1 - \dfrac{\theta(x, y) - \theta_l}{d_\theta}\right), \\ w_l = \left(1 - \dfrac{x - x_i}{d}\right)\left(1 - \dfrac{y - y_i}{d}\right)\left(1 - \dfrac{\theta(x, y) - \theta_r}{d_\theta}\right), \end{cases} \tag{6}$$

where $d_\theta$ is the angle difference between adjacent cell blocks.

After correction, the histogram vectors $h(x_i, y_i, \theta_l)$ and $h(x_i, y_i, \theta_r)$ of Cell$_i$histogram are as follows:

$$\begin{cases} h'(x_i, y_i, \theta_l) = h(x_i, y_i, \theta_l) + w_l G(x, y), \\ h'(x_i, y_i, \theta_r) = h(x_i, y_i, \theta_r) + w_r G(x, y). \end{cases} \tag{7}$$

According to formula (7), the histogram of Cell$_i$ is modified by using the gradient information of all pixels in Cell$_i$ neighborhood. In the same way, we modify the HOG of other cells of the original image to get the modified HOG vector

Step 5: histogram normalization of overlapping blocks.

If there is a large variety of illumination and backgrounds in the image, the range of the gradient value will be large, so good feature standardization is very important to improve the detection rate. There are many ways to standardize, most of which define a cell as a set of blocks and then standardize each block separately. Take the $2 \times 2$ cells adjacent to each other as a block. The $8 \times 8$ pixel is a cell, and the red, blue, yellow, pink, and green boxes are all blocks. That is, the $2 \times 2$ cells in each box form a block. Each block is $16 \times 16$ pixels. There are overlaps between adjacent blocks, so the information of adjacent pixels is effectively used, which is very helpful to the detection results.

$$\begin{aligned} &\text{Blockhog}_i\,(i = 1, 2, \ldots, M) \longleftarrow \text{BinCount}\left(\text{Block}_i\right. \\ &\qquad \left.\cdot (i = 1, 2, \ldots, M)\right), \\ &\text{Blockhog}_i = \{x_1, x_2, \ldots, x_{36}\}. \end{aligned} \tag{8}$$

Next, each block is standardized. There are four cells in a block. Each cell contains 9-dimensional feature vectors, so each block is represented by $4 \times 9 = 36$-dimensional feature vectors. In this paper, L2 norm is used for feature standardization. Let $\varepsilon$ be a very small normalized constant.

$$\begin{aligned} &\text{Blockhog}_i\,(i = 1, 2, \ldots, M) \longleftarrow \frac{\text{Blockhog}_i}{\sqrt{\left\|\text{Blockhog}_i\right\|_2^2 + \varepsilon^2}}, \\ &\left\|\text{Blockhog}_i\right\|_2 = \sqrt{\sum_i^n |x_i|^2}. \end{aligned} \tag{9}$$

After normalizing the histogram of overlapped blocks, the feature vectors of all blocks are combined to form the HOG feature.

Step 6: output HOG features.

$$\begin{aligned} &\mathrm{x}^{\text{RGB–HOG}} \longleftarrow \{\text{Blockhog}_1, \text{Blockhog}_2, \ldots, \text{Blockhog}_M\}, \\ &\mathrm{x}^{\text{RGB–HOG}} = [x_1, x_2, \ldots, x_s]. \end{aligned} \tag{10}$$

*4.2. D-STIP.* The action recognition method based on space-time interest points is one of the more popular action recognition methods at present. It describes the action by detecting the interest points whose pixel values have significant changes in the spatiotemporal neighborhood and extracts the underlying features from them.

Because the space-time interest points are extracted from local features, which are not easily affected by illumination, motion characteristics, or background changes, this method has improved robustness over less localized methods.

In this paper, we implement the representation of space-time interest points and space-time words based on depth image. This method first extracts the accurate space-time interest points from the samples and then extracts the local neighborhood features of the interest points. Next a space-time codebook based on the feature of the interest points is established, and a statistical histogram of the interest points based on the space-time codebook is obtained. The D-STIP extraction flowchart is shown in Figure 3.

Step 1: Dollar STIP detection.

Laptev extended the 2D Harris corner [36] to the 3D Harris corner [37] and used them as the significant changing points in the spatiotemporal domain. Firstly, the video sequence is represented in the linear space as

$$L\left(\cdot, \sigma_l^2, \tau_l^2\right) = g\left(\cdot, \sigma_l^2, \tau_l^2\right) * D(\cdot). \tag{11}$$

Then, the matrix can be obtained as

$$N = g\left(\cdot, \sigma_l^2, \tau_l^2\right) * \begin{vmatrix} L_x^2 & L_x L_y & L_x L_t \\ L_x L_y & L_y^2 & L_y L_t \\ L_x L_t & L_y L_t & L_t^2 \end{vmatrix}, \tag{12}$$

where $g(\cdot, \sigma_l^2, \tau_l^2)$ is the Gaussian kernel function, $\sigma_l^2$ is the spatial factor, $\tau_l^2$ is the temporal factor, and $D(\cdot)$ is the depth video image sequence. The three eigenvalues of the matrix $N$ $\lambda_1$, $\lambda_2$, and $\lambda_3$ correspond to changes in the depth video sequence $D(\cdot)$ in the two spatial directions $(x, y)$ and on temporal domain $t$, respectively. When these values are all large, it means that the video changes significantly along all three directions, and therefore this point is a space-time interest point.

Laptev defined the response function of interest points as

$$H = \det(N) - k^* \operatorname{trace}^3(N) = \lambda_1\lambda_2\lambda_3 - k^*\left(\lambda_1 + \lambda_2 + \lambda_3\right)^3, \tag{13}$$

where $\det(N)$ and $\operatorname{trace}(N)$ are determinants and traces of matrices, respectively, and $k$ is a coefficient and usually takes the value of 0.005. The function value $H$ obtains the local maximum at the point of interest.

3D Harris corner detection is very sensitive to movements that changes the direction of speed, such as walking, running, and waving, but for other movements, such as rotation and periodic movement, there is often no point of interest detected.

The interest points detected by the 3D Harris spatiotemporal corner are too sparse. Although we expect sparsity to some extent, if feature points are too sparse, it means that there are too few underlying features. This can negatively affect the recognition results. Dollar et al. [38] proposed a new method for interest point detection, which makes the extracted interest points more dense. The response function $H$ is calculated by the separable linear filter:

$$H = \left(D * g * h_{\text{ev}}\right)^2 + \left(D * g * h_{\text{od}}\right)^2, \tag{14}$$

where $g(x, y; \sigma)$ is a 2D Gaussian smoothing kernel function for spatial filtering:

$$g(x, y; \sigma) = \frac{1}{2\pi\sigma^2} \exp\left(\frac{-\left(x^2 + y^2\right)}{2\sigma^2}\right). \tag{15}$$

$h_{\text{ev}}$ and $h_{\text{od}}$ are orthogonal components of one-dimensional Gabor function, which are used for filtering in time domain:

$$\begin{aligned} h_{\text{ev}}(t; \tau, \omega) &= -\cos(2\pi t\omega) e^{-t^2/\tau^2}, \\ h_{\text{od}}(t; \tau, \omega) &= -\sin(2\pi t\omega) e^{-t^2/\tau^2}, \end{aligned} \tag{16}$$

where $\omega = \tau/4$ and the response function $H$ has only two parameters $\sigma$ and $\tau$, corresponding to space and time scales, respectively. The point whose response function H has the local maximum value is detected as the point of interest if it is greater than a certain threshold value. The number of interest points detected can be controlled by the threshold value. In order to solve the problem of scale changes, the method of multiscale combination can be used to detect interest points.

However, the noise points in the depth image will also have a greater response to the kernel function in the space-time domain, so they are mistakenly detected as points of interest. The wrong interest points will introduce a lot of errors to the subsequent feature description, which will seriously reduce the description ability of spatiotemporal interest points. In this paper, a correction filter is applied to the detected interest points to reduce noise interference.

The noise of depth image can be roughly divided into three categories: one is generated by depth sensing equipment, which appears randomly in the whole depth image. This kind of random noise generally appears less and has little influence on the detection of interest points. The second kind of noise appears at the edge of the scene object because of the nature of structured light imaging. The depth of noise often jumps between the foreground and background on both sides of the edge. The third is due to the problems of reflective material on the surface of the object, fast movement, and so on. The "holes" appear on the depth image, that is, the loss of the depth value on the image (the pixel value is zero). The second and third kind of noise will produce a lot of interference to the detection of interest points, and they are difficult to be removed by ordinary spatial smoothing filtering. Generally speaking, the disturbance frequency of noise signal is much faster than the motion frequency of human body, and it may appear in consecutive frames of human motion time segment. Based on this, we can calculate the average time of noise disturbance and then filter the obtained interest points. The correction function of interest point is shown in the following formula:

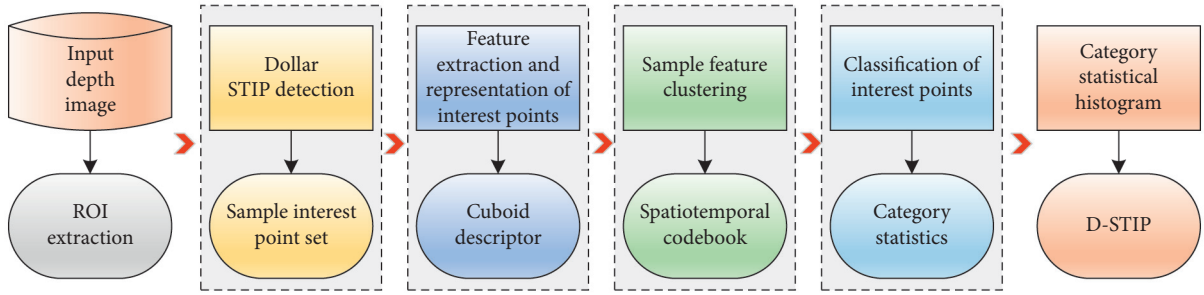


Figure 3: D-STIP extraction flowchart.

$$s(x, y, t_0 \mid \tau) = \frac{\sum_{i=1}^{n_{\text{fp}}} \delta t_i (x, y)}{n_{\text{fp}} (x, y)}, \tag{17}$$

where $n_{\text{fp}}(x, y)$ is the number of times the noise signal jumps in the whole movement period and $\delta t_i (x, y)$ is the duration of the ith jump. The interest point correction function is the ratio of noise signal in pixel $t$ to the whole time period. It gets higher values at the real moving pixels and lower values at the noise points. Therefore, pixels (noise points) with low ratio can be filtered out by setting a threshold value.

After detecting the interest points, it is necessary to select appropriate local feature descriptors to represent the interest points.

Step 2: feature description of interest points.

Dollar et al. [38] proposed the concept of the cuboid for the detection of interest points. The cuboid is a cuboid video block centered on interest points, whose edge length generally depends on the detection scale of the interest points. Using cuboid descriptors to represent interest points can represent interest points along with their neighborhood information.

Firstly, three kinds of transformations are performed with cuboid detection: (1) pixel value normalization; (2) for each pixel $(x, y, t)$, the gradient in different directions is calculated, and three cuboid matrices $(C_x, C_y, C_t)$ are obtained; (3) the Lucas–Kanade optical flow [39] is calculated for the adjacent frames, and two cuboid matrices $(V_x, V_y)$ are obtained. Therefore, for each point of interest $p_i$ in the set of extracted points of interest $P = \{p_1, p_2, \ldots, p_n\}$, we can calculate its feature description as $F_i = \{C_{ix}, C_{iy}, C_{it}, V_{ix}, V_{iy}\}$.

Step 3: the establishment of the space-time codebook.

Because of the difference of the performers' wearing, action mode, and amplitude, the same action will have different interest points in different videos. However, the features of these interest points are similar and provide the essential description of the temporal and spatial features of the action. After the feature representation of interest points, we need to use the feature vector to represent different actions, that is, to model the action. The most common modeling method to model the interest points is the bag of video words (BoVW) method.

A k-means clustering algorithm is used to cluster the feature set extracted from the training dataset. The number of clustering centers is selected in the experiment. The generated clustering centers are regarded as the spatiotemporal words $w_i = \{f_1, f_2, \ldots, f_m\}$, $m$ is the feature dimension, and $f_i$ is the $i^{\text{th}}$ feature component of the spatiotemporal words. The set of all spatiotemporal words is $v = \{w_1, w_2, \ldots, w_n\}$, where $n$ is the number of clustering centers. For different action videos, the spatiotemporal codebook corresponding to different action categories is trained according to the above steps in the training set. In the subsequent action recognition process, the interest points are classified by calculating the distance between the feature of interest points and spatiotemporal words.

The statistical histogram of interest points $H = \{h_1, h_2, \ldots, h_n\}$ based on the spatiotemporal codebook is obtained by counting the categories of all interest points in the video, where $n^{]}$ is equal to the dimension of spatiotemporal codebook and $h_i$ is the frequency of the $i^{\text{th}}$ spatiotemporal word in the video. Finally, the histogram is used as the video descriptor:

$$\begin{aligned} \text{y}^{D\text{–STIP}} &\longleftarrow \{h_1, h_2, \ldots, h_n\}, \\ \text{y}^{D\text{–STIP}} &= [y_1, y_2, \ldots, y_N]. \end{aligned} \tag{18}$$

*4.3. S-JRPF.* Skeleton joint point is the visual salient point of human body, and its movement in 4D space can reflect the semantic information of action. The research of joint-based motion recognition can be traced back to Johansson's early work [40]. Their experiments show that most of the movements can be identified only according to the position of joint points. This idea has been adopted by a large number of subsequent researchers and has gradually formed an important branch of human motion recognition methods.

With the release of the Microsoft Kinect sensor, it is convenient to get the depth map of the scene and the 3D skeleton of the human body. Compared with the feature of deep image extraction, the 3D skeleton data provided by Kinect has only 20 joint points as information of the human body. After feature extraction, the feature dimension will be

lower and thus computations are smaller, which is beneficial to real-time performance of action recognition algorithms. For three-dimensional skeleton motion data, we first need to express the motion through the feature expression before we can correctly identify the motion. We do so by using the Kinect. By using the coordinate information of the 20 joint points from the Kinect, we can find a good representation of the human body.

Based on the joints' modal data, this paper presents the spatial distribution feature of joint projection to represent human motion. Firstly, the 3D skeleton data of each frame are collected and projected in three planes (*XOY* plane, *YOZ* plane, and *XOZ* plane) to obtain the position distribution of projection points of single frame 3D skeleton joint data on different projection planes. The projection of the joint points of the human body is shown in Figure 4.

Then, the joint points on the three projection planes are represented in polar coordinates:

$$\begin{cases} \rho_i^{\mathrm{XOY}} = \sqrt{x_i^2 + y_i^2}, \theta_i^{\mathrm{XOY}} = \arctan\left(\dfrac{y_i}{x_i}\right), \\ \rho_i^{\mathrm{YOZ}} = \sqrt{y_i^2 + z_i^2}, \theta_i^{\mathrm{YOZ}} = \arctan\left(\dfrac{z_i}{y_i}\right), i = 1, 2, \ldots, 20, \\ \rho_i^{\mathrm{XOZ}} = \sqrt{x_i^2 + z_i^2}, \theta_i^{\mathrm{XOZ}} = \arctan\left(\dfrac{z_i}{x_i}\right). \end{cases} \tag{19}$$

Finally, the polar coordinates of the projection points on the three projection planes are spliced as the feature vectors of the frame. In order to make the feature data fall in [0, 1], the joints' relative position feature can be obtained by using minimax normalization since the skeleton modal information is invariant under translation transformations, scale transformations, and rotation transformations. Therefore, the feature view in joints' mode can be expressed as

$$\begin{aligned} \mathbf{z}^{\mathrm{S-JRPF}} \longleftarrow & \left\{\rho_1^{\mathrm{XOY}}, \ldots, \rho_{20}^{\mathrm{XOY}}, \rho_1^{\mathrm{YOZ}}, \ldots, \rho_{20}^{\mathrm{YOZ}}, \rho_1^{\mathrm{XOZ}}, \ldots,\right. \\ & \left. \rho_{20}^{\mathrm{XOZ}}, \theta_1^{\mathrm{XOY}}, \ldots, \theta_{20}^{\mathrm{XOY}} \theta_1^{\mathrm{YOZ}}, \ldots \theta_{20}^{\mathrm{YOZ}}, \theta_1^{\mathrm{XOZ}}, \ldots, \theta_{20}^{\mathrm{XOZ}}\right\}, \\ \mathbf{z}^{\mathrm{S-JRPF}} = & \left[z_1, z_2, \ldots, z_Q\right]. \end{aligned} \tag{20}$$

## 5. Recognition Algorithm

Experiments show that the classification performance of the learning system is better than that of each basic classifier, so the effectiveness of ensemble learning is proved. Dietterich [41] listed ensemble learning as the top four research directions of machine learning. Integrated learning is to build a strong classifier with excellent classification performance and generalization ability. In the traditional classification algorithm, SVM classification algorithm and KNN classification algorithm have better classification effect than other traditional classification algorithms.

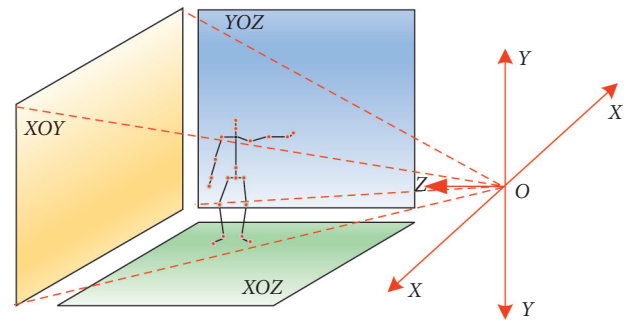


Figure 4: The projection of the joint points of the human body.

However, the classification effect of the base classifier is not stable. Simply using the base classifier to classify the data, it is easy to make the classification result overfit. Combining the base classifier according to the combination strategy produces a strong classifier, and the classification performance of the strong classifier is better than that of each base classifier. In order to generate a better classification method, this paper will build an ensemble KNN multiclassifier model.

The KNN method is based on analogical learning, which is a nonparametric classification technology. It is very effective in pattern recognition based on statistics. It can obtain high classification accuracy for unknown and non-normal distribution and has the advantages of robustness and clear concept.

The basic ideas are as follows: feed in new data without a class label, extract the feature from the new data, and compare the new feature to the feature of each sample in the training set; then select the class labels of the $k$ nearest (most similar) samples and count the number of the label occurrences. The class with the highest occurrence count is determined to be the class of the new data.

Now, we expect to use KNN classification rules to complete the correct classification of test data point $\mathbf{x}$. By finding $k$ nearest neighbors near the test sample point $\mathbf{x}$, the test sample point is predicted to be the category with the most $k$ nearest neighbors. Among the $N$ training samples, $N_1$ training samples belong to category $\omega_1$, $N_2$ training samples belong to category $\omega_1$, ..., $N_c$ training samples belong to category $\omega_c$. If $k_1, k_2, \ldots, k_c$ belong to categories $\omega_1, \omega_2, \ldots, \omega_c$, respectively, then the discriminant function can be defined as

$$g_i(\mathbf{x}) = k_i, \quad i = 1, 2, \ldots, c. \tag{21}$$

The decision rule is if

$$g_j(\mathbf{x}) = \max_i k_i, \tag{22}$$

then

$$\mathbf{x} \in \omega_j. \tag{23}$$

To classify specific actions, we can search the training set for the $K$ actions that are nearest to the new action and determine the class of the new action based on the classes of these $K$ actions. This paper proposes an integrated classification method using multilearners based on a training set

of multimodal features, which is more effective to identify the new action. It fully utilizes the biasing effects from different learners and therefore enhances the generalizing capability of the learning. The implementation sequence of the algorithm is as follows:

Step 1: describe the training sets of action features with different modal information separately.

$$\begin{aligned} T_{\text{RGB–HOG}} &= \left\{\mathbf{x}_1^{\text{RGB–HOG}}, \mathbf{x}_2^{\text{RGB–HOG}}, \ldots, \mathbf{x}_N^{\text{RGB–HOG}}\right\}, \\ T_{D\text{–STIP}} &= \left\{\mathbf{y}_1^{D\text{–STIP}}, \mathbf{y}_1^{D\text{–STIP}}, \ldots, \mathbf{y}_N^{D\text{–STIP}}\right\}, \\ T_{S\text{–JRPF}} &= \left\{\mathbf{z}_1^{S\text{–JRPF}}, \mathbf{z}_2^{S\text{–JRPF}}, \ldots, \mathbf{z}_N^{S\text{–JRPF}}\right\}, \end{aligned} \tag{24}$$

where $T_{\text{RGB–HOG}}$, $T_{D\text{–STIP}}$, and $T_{S\text{–JRPF}}$ are training sample sets and the number of samples in each training set is $N$.

Step 2: determine the vector representation of the action in three kinds of modal description.

$$\begin{aligned} \mathbf{x}_*^{\text{RGB–HOG}} &= [x_1^*, x_2^*, \ldots, x_S^*], \\ \mathbf{y}_*^{D\text{–STIP}} &= [y_1^*, y_2^*, \ldots, y_N^*], \\ \mathbf{z}_*^{S\text{–JRPF}} &= [z_1^*, z_2^*, \ldots, z_Q^*], \end{aligned} \tag{25}$$

where $\mathbf{x}_*^{\text{RGB–HOG}}, \mathbf{y}_*^{D\text{–STIP}}$, and $\mathbf{z}_*^{S\text{–JRPF}}$ are the three feature vector representations of the action $\Theta$ to be predicted. The sample dimensions of $\mathbf{x}_*^{\text{RGB–HOG}}, \mathbf{y}_*^{D\text{–STIP}}$, and $\mathbf{z}_*^{S\text{–JRPF}}$ are $S, N$, and $Q$, respectively.

Step 3: select the $Topk_1$, $Topk_2$, and $Topk_3$ actions that are nearest to the action to be predicted from the three training sets using different distance measurement formulas, separately. The equations to compute the similarity for various models are

$$\begin{aligned} D_1\left(\mathbf{x}_*^{\text{RGB–HOG}}, \mathbf{x}_j\right) &= \left[\sum_{i=1}^{S}\left(x_i^* - x_{ji}\right)\right]^{1/2}, \\ D_2\left(\mathbf{y}_*^{D\text{–STIP}}, \mathbf{y}_j\right) &= \sum_{i=1}^{N}\left|y_i^* - y_{ji}\right|, \\ D_3\left(\mathbf{z}_*^{S\text{–JRPF}}, \mathbf{z}_j\right) &= \left(\mathbf{z}_*^{S\text{–JRPF}} \text{-} \mathbf{z}_j\right)' V^{-1}\left(\mathbf{z}_*^{S\text{–JRPF}} \text{-} \mathbf{z}_j\right), \end{aligned} \tag{26}$$

where $D_1(\mathbf{x}_*^{\text{RGB–HOG}}, \mathrm{x}_j)$ is the Euclidean distance metric, $D_2(\mathbf{y}_*^{D\text{–STIP}}, \mathrm{y}_j)$ is the Manhattan distance metric, $D_3(\mathbf{z}_*^{S\text{–JRPF}}, \mathrm{z}_j)$ is the Mahalanobis distance, and $V^{-1}$ is the covariance function.

Step 4: compute the weight of each class of the $Topk_1 + Topk_2 + Topk_3$ actions that are nearest to the action to be predicted:

$$\begin{aligned} p\left(\Theta, C_j\right) = &\sum_{\mathrm{x}_j \in \text{KNN}} \omega_{1j} D_1\left(\mathbf{x}_*^{\text{RGB–HOG}}, \mathbf{x}_j\right) \text{Attribute}\left(\mathbf{x}_j, C_j\right) \\ &+ \sum_{\mathrm{y}_j \in \text{KNN}} \omega_{2j} D_2\left(\mathbf{y}_*^{D\text{–STIP}}, \mathbf{y}_j\right) \text{Attribute}\left(\mathbf{x}_j, C_j\right) \\ &+ \sum_{\mathrm{z}_j \in \text{KNN}} \omega_{3j} D_3\left(\mathbf{z}_*^{S\text{–JRPF}}, \mathbf{z}_j\right) \text{Attribute}\left(\mathbf{z}_j, C_j\right), \end{aligned}$$

$$\omega_j = \frac{D_j^{-1}}{\sum_{j=1}^{K} D_j^{-1} + \varepsilon}, \tag{27}$$

where $\mathbf{x}_*^{\text{RGB–HOG}}, \mathbf{y}_*^{D\text{–STIP}}$, and $\mathbf{z}_*^{S\text{–JRPF}}$ are the feature vectors of action $\Theta$ described in various models. $(\cdot)$ is the attribute function of the class. If $\mathbf{x}_*^{\text{RGB–HOG}}$ belongs to class $C_j$, $(\cdot)$ takes 1; otherwise, it takes 0. $\omega_j$ is the weight coefficient of the nearest neighbor of the sample, $D_j^{-1}$ is the reciprocal of the distance, and $\varepsilon$ is the smaller positive number which is not 0.

Step 5: compare the class weights and assign the action to be predicted to the class with the largest weight.

$$y \longleftarrow \underset{C_j}{\text{argmax}}\left(p\left(\Theta, C_j\right)\right). \tag{28}$$

## 6. Experiments and Results

This section provides the experimental results and analysis of our algorithm as applied to the G3D dataset and Cornell Activity Dataset 60.

*6.1. Figure of Merit.* Cross-validation is adopted in the experiments to train the classification model and to test its performance. In addition, the *precision*, *recall*, and *F-measure* are used to evaluate the effectiveness of the algorithm, as illustrated in the following equations:

$$\text{precsion} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{29}$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{30}$$

$$F - \text{measure} = \frac{2\text{RP}}{\text{R} + \text{P}}. \tag{31}$$

In a biclassification, $TP$ is the number of positive samples that are correctly predicted by the classification model, $FP$ is the number of negative samples that are classified as positive

(a) (b) (c)

Figure 5: Sample images from the G3D dataset: (a) RGB image; (b) depth image; (c) skeleton image.

Figure 6: Sample images from the Cornell Activity Dataset 60.

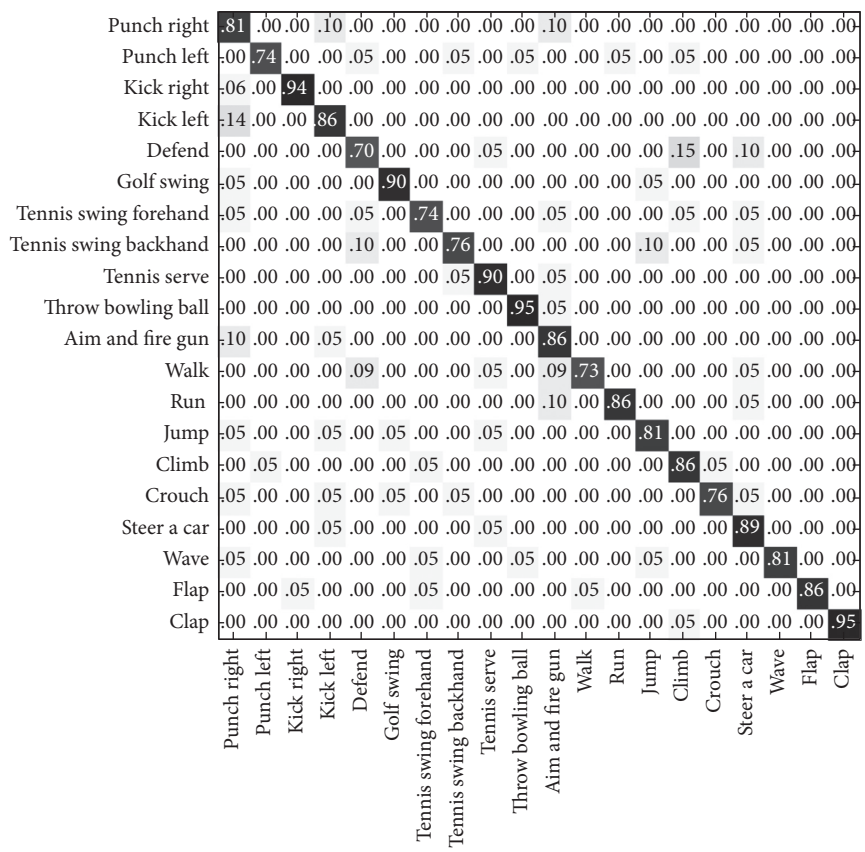


Figure 7: Confusion matrix results using RGB-HOG features in G3D dataset.

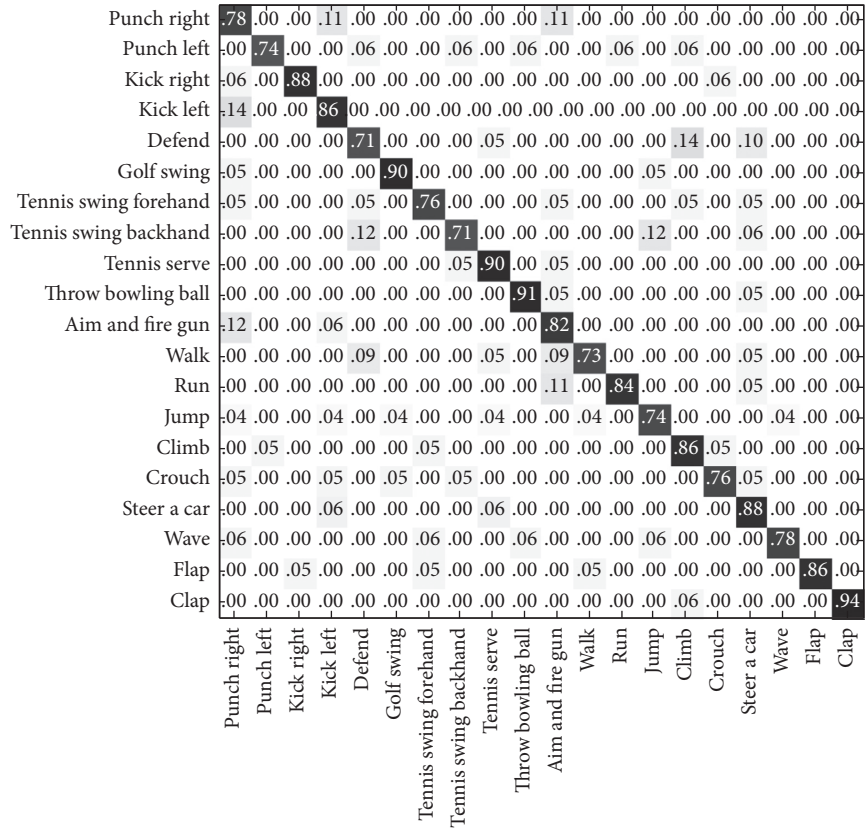


Figure 8: Confusion matrix results using D-STIP features in G3D dataset.

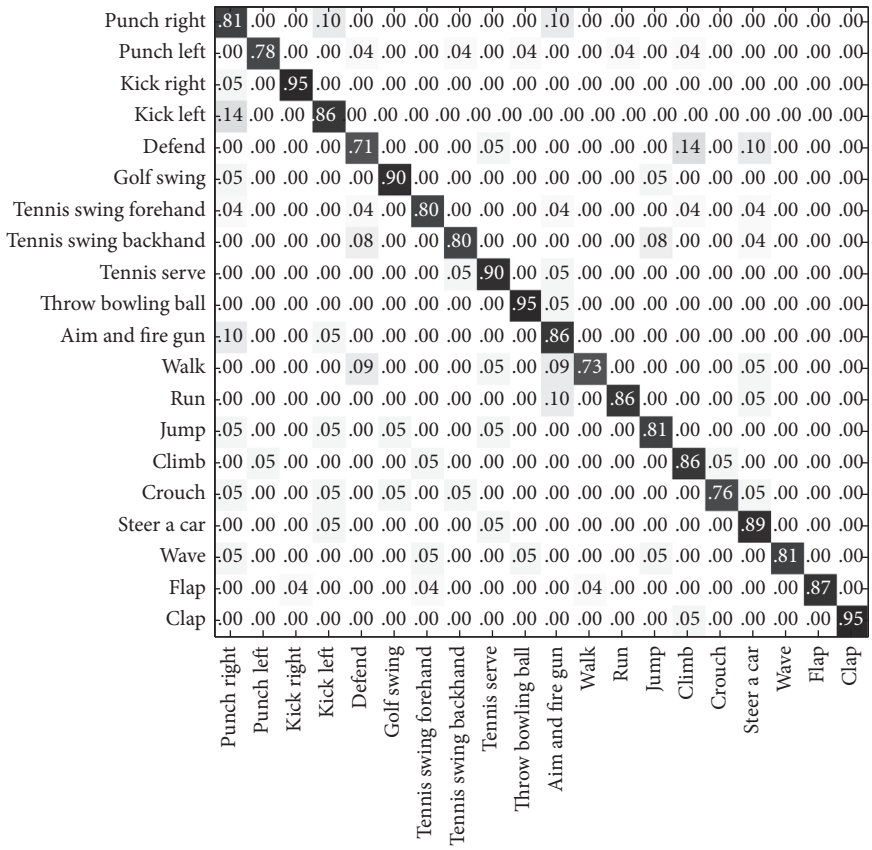


Figure 9: Confusion matrix results using S-JRPF features in G3D dataset.

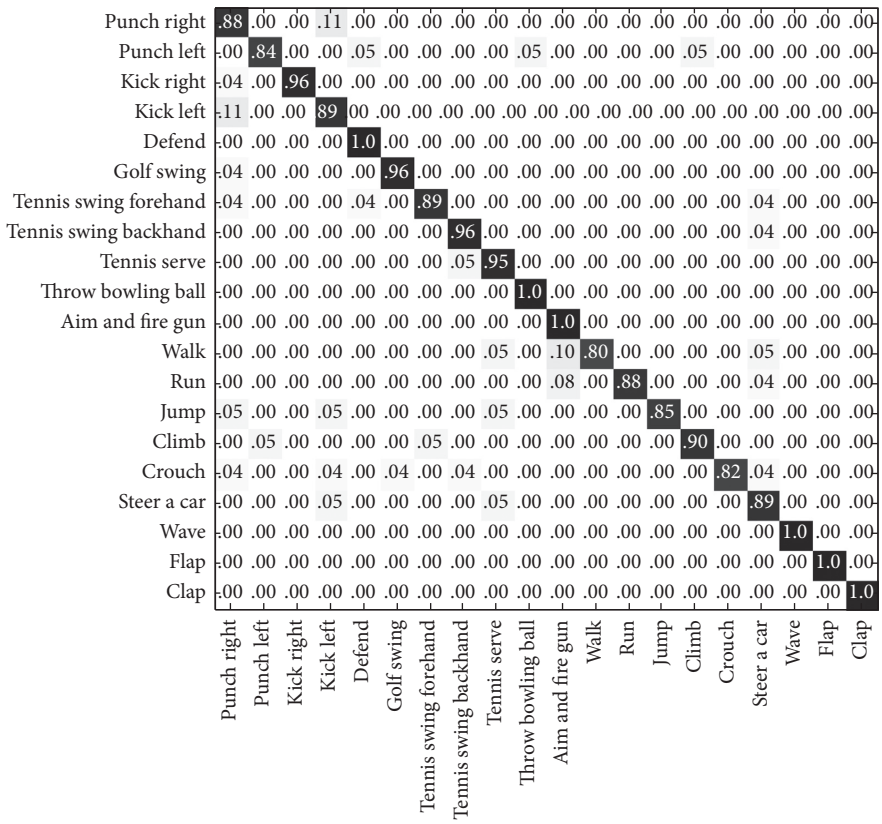


Figure 10: Confusion matrix results using the proposed method in G3D dataset.

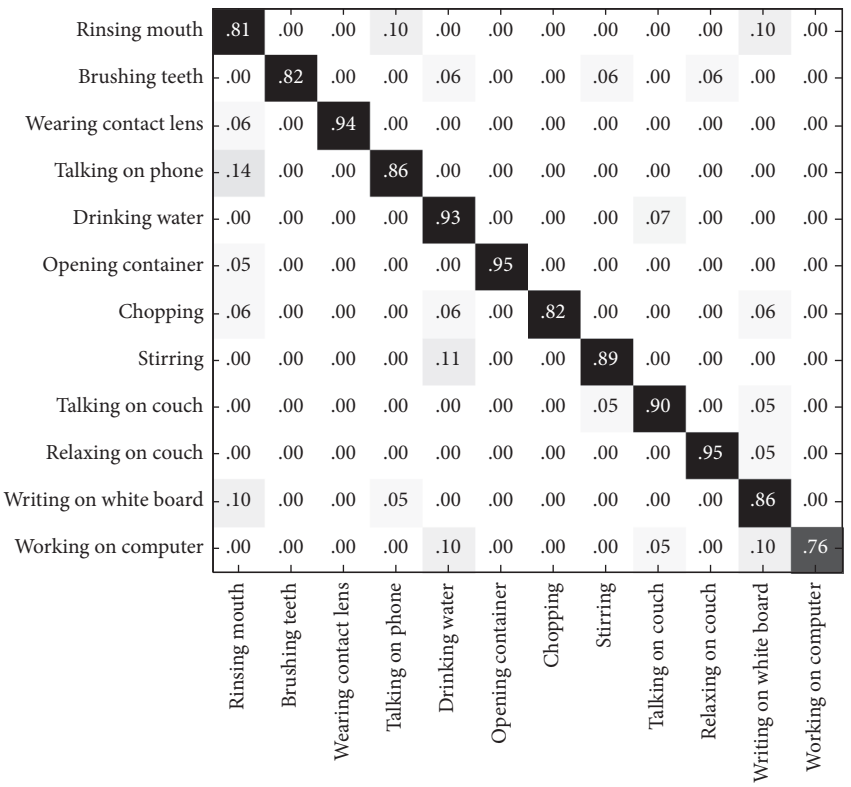


Figure 11: Confusion matrix based on RGB-HOG feature on the CAD60 dataset.

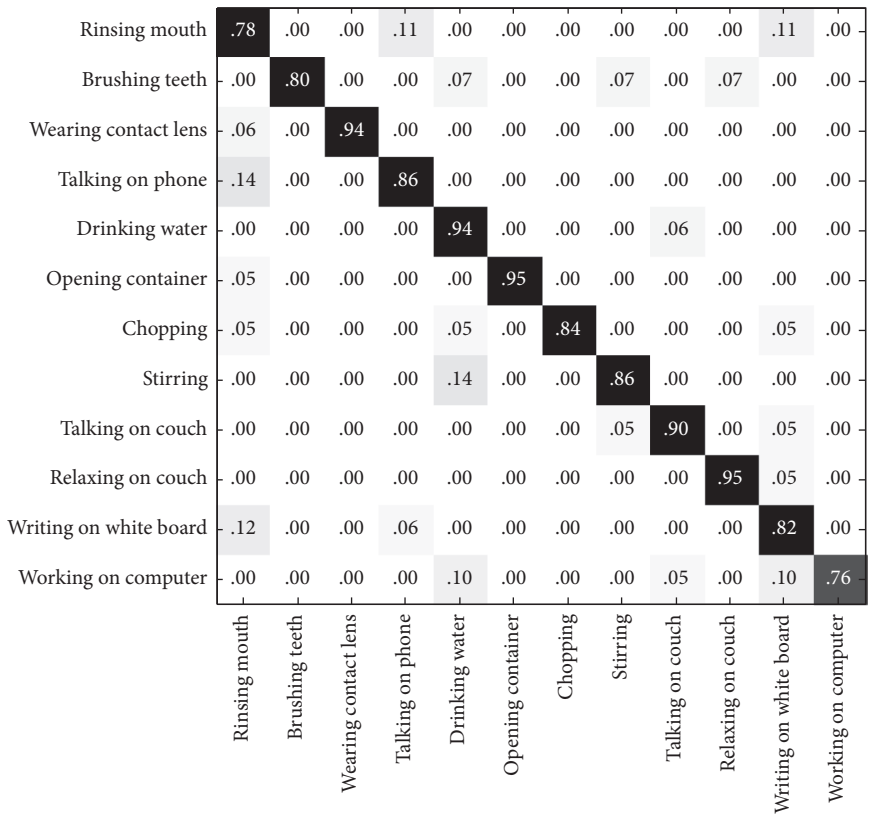


Figure 12: Confusion matrix based on D-STIP feature on the CAD60 dataset.

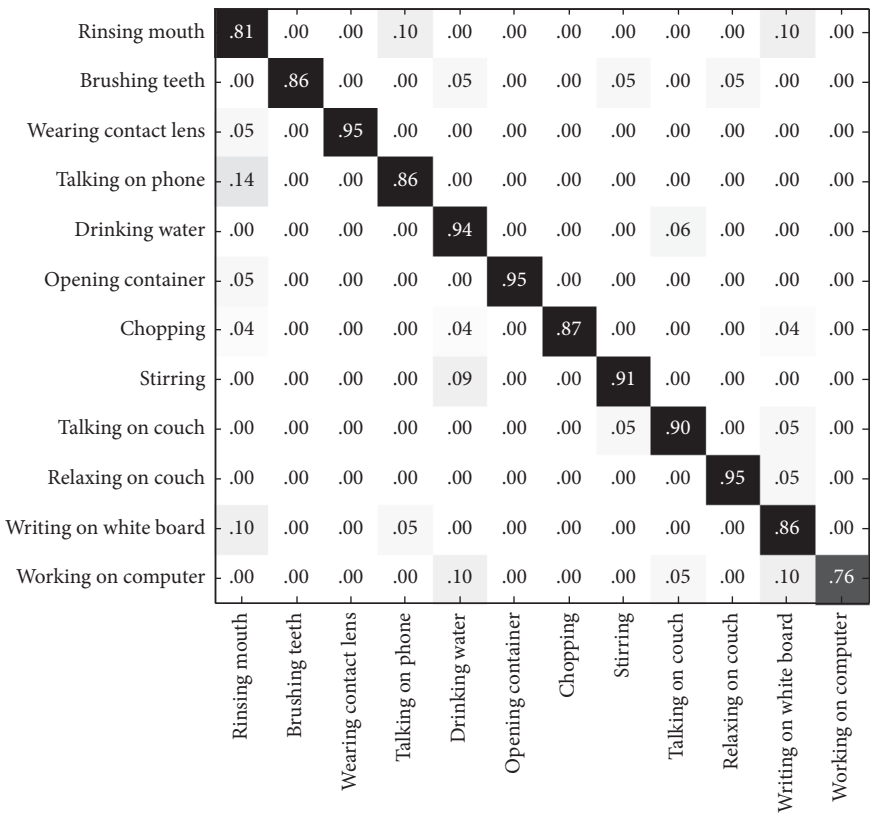


Figure 13: Confusion matrix based on S-JRPF feature on the CAD60 dataset.

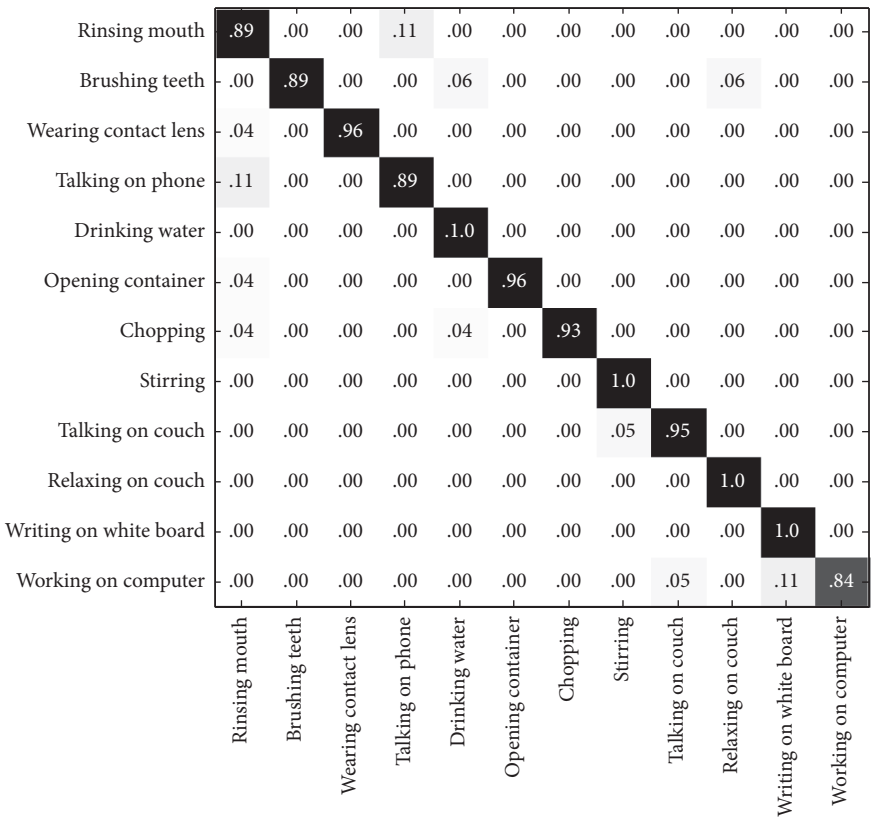


Figure 14: Confusion matrix based on this paper's method on the CAD60 dataset.

by the model, and *FN* is the number of positive samples that are classified as negative by the model. These formulas can be extended to multiclass classifications.

*6.2. Datasets.* The G3D dataset contains 20 categories of human actions, each performed by 10 persons. The 20 category actions are punch right, punch left, kick right, kick left, defend, golf swing, tennis swing forehand, tennis swing backhand, tennis serve, throw bowling ball, aim and fire gun, walk, run, jump, climb, crouch, steer a car, wave, flap, and clap. The Cornell Activity Dataset 60 (CAD60) contains 12 actions, which are performed by 4 persons in 5 different environments. These actions are rinsing mouth, brushing teeth, wearing contact lens, talking on phone, drinking water, opening container, chopping, stirring, talking on couch, relaxing on couch, writing on white board, and working on computer. The actions in the G3D and CAD60 datasets contain image information in three different kinds of modals: RGB image, depth image, and skeleton joint data, as illustrated in Figures 5 and 6. In the experiment, we randomly divide all videos into training data and test data according to the ratio of 7 : 3. The final test result is the average of 10 test results. In the experiment, we set $Topk_1$, $Topk_2$, and $Topk_3$ to 5.

*6.3. Experiments and Results.* In this section, we validate the feasibility and efficiency of this paper's method in two experiments. In the first, we test the recognition rate and the *precision*, *recall*, and *F-measure* on the G3D and CAD60 datasets based on a single feature and this paper's algorithm. In the second experiment, we compare our method to other algorithms.

We present the result of this paper's method in Experiment 1 with the confusion matrix. The $(i, j)$ element of the matrix is the percentage of action of class $i$ that are classified as the action of class $j$. Therefore, the greater the diagonal elements, the better the classification result.

Figures 7–10 illustrate the recognition rates using the single modal feature on the G3D dataset with confusion matrices. Figure 10 shows the recognition rate resulting from this paper's method using multimodal information. From the above figures, it can be observed that the 20 categories of action recognition rates based on multimodal features are all higher than those using the single modal feature. For the six actions of defend, throw bowling ball, aim and fire gun, wave, flap, and clap, the accuracy is 100%. Figures 11–14 illustrate the recognition rate using the single modal feature on the CAD60 dataset with confusion matrices. Figure 14 shows the recognition rate of this paper's method using multimodal features on the CAD60 dataset. Through comparison, it can be found that this paper's method achieves a good recognition rate of 94% on the CAD60 dataset, with 100% accuracy for the actions of drinking water, stirring, relaxing on couch, and writing on white board. The results of experiments show that the integrated KNN modal based on multimodal data is better than single KNN model based on single modal data. Single

Table 1: Recognition rate using the single model feature and multimodal features.

| Dataset | Feature | Precision (%) | Recall (%) | *F*-measure (%) |
|---|---|---|---|---|
| G3D | RGB-HOG | 82 | 81 | 81 |
| | D-STIP | 81 | 82 | 81 |
| | S-JRPF | 84 | 84 | 84 |
| | Ours | 93 | 92 | 92 |
| CAD60 | RGB-HOG | 88 | 88 | 88 |
| | D-STIP | 87 | 86 | 86 |
| | S-JRPF | 88 | 88 | 88 |
| | Ours | 94 | 94 | 94 |

Table 2: Comparison of various action recognition methods.

| Dataset | Descriptor | This paper's method (%) | Boosting (%) | Bagging (%) | SVM (%) | ANNs (%) |
|---|---|---|---|---|---|---|
| G3D | RGB-HOG | **82** | 79 | 75 | 79 | 74 |
| | D-STIP | **81** | 80 | 79 | 80 | 80 |
| | S-JRPF | **84** | 81 | 80 | 80 | 74 |
| Ours | **93** | — | — | — | — | |
| CAD60 | RGB-HOG | **88** | 84 | 80 | 81 | 76 |
| | D-STIP | **87** | 80 | 79 | 80 | 80 |
| | S-JRPF | **88** | 81 | 80 | 80 | 74 |
| | Ours | **94** | — | — | — | — |

Table 3: Comparison of various action recognition methods.

| Method | Descriptor | Recognition Methods | Accuracy | |
|---|---|---|---|---|
| | | | G3D (%) | CAD60 (%) |
| Dollar's method [38] | Sparse spatiotemporal features | SVM | 78 | 83 |
| Liu's method [42] | PMI spatiotemporal features | SVM | 82 | 86 |
| Laptev's method [43] | Spatiotemporal corner | SVM | 87 | 84 |
| Rapantzikos's method [44] | Dense saliency spatiotemporal features | KNN | 88 | 89 |
| Rodriguez's method [45] | Spatiotemporal template | Template matching | 88 | 89 |
| Dropout-based CNN [46] | — | Random dropout-based CNN | 89 | 90 |
| Our Approach | Mixed features | SE-SVM | 93 | 94 |

KNN model is difficult to meet the needs of human behavior prediction.

In addition, we present, in terms of *precision*, *recall*, and *F-measure*, the recognition rates using the single modal feature and multimodal features in Table1. The recognition rates of this paper's method using multimodal features are higher than those of the methods using the single modal feature.

In the second experiment, we compare this paper's method to other classical machine learning methods. Table 2 shows the comparison of this paper's algorithm to boosting, bagging, support vector machine (SVM), and artificial neural networks (ANNs). From the results in Table 2, it can be observed that the integrated multilearner recognition algorithm based on multimodal features achieves the highest recognition rate of 94%. It can be seen from the table that the combined nearest neighbor classifier based on multimodal features has better classification accuracy, mainly because our proposed algorithm is a behavior recognition algorithm based on multimodal feature fusion, which can make full use of the complementarity between different models. In general, the accuracy of the combined nearest neighbor classifier based on multimodal data is higher than that of the original single nearest neighbor classifier.

Table 3 compares the average class accuracy of our method with results reported by other researchers. Compared with the existing traditional machine learning approaches, our method shows much better performance, outperforming the state-of-the-art approaches. Note that a precise comparison between the approaches is difficult, since experimental setups, e.g., different strategies in training, slightly differ with each approach. In addition, compared with random dropout-based CNN method only using RGB data, our method also achieves better results. Dropout method is to set the weights of some hidden layer nodes of neural network to 0 during training, which is used to solve the model overfitting problem caused by too few training samples. On the basis of dropout, we further improve it and add a layer of randomization process to realize random dropout, so as to further prevent the overfitting phenomenon of the model. Therefore, when the training sample data are small, the multilearner recognition

method based on multimodal features is better than the deep learning method.

## 7. Conclusion

A human action recognition method based on multimodal features is proposed in this paper. Through the Kinect sensor, three modal information is acquired for each image, and the RGB-HOG feature, D-STIP feature, and S-JRPE feature are extracted. An integrated learning strategy with multilearners is adopted, which fully utilizes the biasing effects from different learners. The method achieves good recognition rates on standard public datasets and is robust in real time. Although the method presented herein achieved good experimental results on public datasets, there still remain many issues in action recognition, calling for deeper investigations. Generally, a large amount of tagged video training samples are necessary for the classifier to achieve a good generalizing capability. This requires a lot of manual tagging work, and thus practical modeling can be difficult. It is thus a very valuable direction to investigate how to enhance the learning system's performance utilizing the abundant untagged video samples at hand in the public data.

## Data Availability

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

## Disclosure

The abstract of the manuscript was already presented as conference proceedings in Global Intelligence Industry Conference (GIIC 2018).

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This research was funded by the National Natural Science Foundation of China (grant nos. 61673249, 61672204, U1805263, and 61662025), the Natural Science Foundation of Anhui Province (grant no. 2008085MF202), the Guidance Project of Science and Technology of Xiamen (grant no. 3502Z20179038), the Research Foundation of Education Bureau of Hunan Province (grant no. 16C1311), the Natural Science Foundation of Zhejiang Province (grant nos. LY20F030006 and LY20F020011), the Natural Science Research Project of Universities of Anhui Province (grant no. KJ2019A1121), the Research Development Project Fund of Hefei University (grant no. 18zr19zda), the Key R&D Program of Shanxi Province (grant no. 201903D421050), and the Key Teaching and Research Project of Hefei University (grant no. 2018hfjyxm09).

## References

[1] M. Kulbacki, J. Segen, S. Wojciechowski et al., "Intelligent video monitoring system with the functionality of online recognition of people's behavior and interac between people," in *Proceedings of the Intelligent Information and Database Systems*, N. T. Nguyen, D. H. Hoang, T. P. Hong, H. Pham, and B. Trawinski, Eds., pp. 492–501, New York, NY, USA, 2018.

[2] Y. Gao, X. Xiang, N. Xiong et al., "Human action monitoring for healthcare based on deep learning," *Ieee Access*, vol. 6, pp. 52277–52285, 2018.

[3] H. Kim, S. Lee, Y. Kim et al., "Weighted joint-based human behavior recognition algorithm using only depth information for low-cost intelligent video-surveillance system," *Expert Systems with Applications*, vol. 45, pp. 131–141, 2016.

[4] A. Chaaraoui, J. Padilla-López, F. Ferrández-Pastor, M. Nieto-Hidalgo, and F. Flórez-Revuelta, "A vision-based system for intelligent monitoring: human behaviour analysis and privacy by context," *Sensors*, vol. 14, no. 5, pp. 8895–8925, 2014.

[5] H. Sang and Q. Tian, "Rapid action recognition system for human-computer interaction," *Computer Engineering and Applications*, vol. 55, no. 6, pp. 101–107, 2019.

[6] M. Lou, J. Li, G. Wang, and G. He, "Action recognition accelerator for human-computer interaction on FPGA," *2019 Ieee International Symposium on Circuits and Systems*, vol. 14, 2019.

[7] H. D. Mehr and H. Polat, "Human activity recognition in smart home with deep learning approach," 2019.

[8] M. G. Al Zamil, "Multimodal daily activity recognition in smart homes," 2019.

[9] Y. A. Malkani, W. A. Memon, and L. Das Dhomeja, "A low-cost activity recognition system for smart homes," 2018.

[10] A. S. Fangbemi, B. Liu, N. H. Yu, and Y. Zhang, "Efficien-thuman action recognition interface for augmented and virtualreality applications based on binary Descriptor," in *Computer Engineering and Applications*, L. T. DePaolis and P. Bourdot, Eds., vol. 19, pp. 252–260, 2018.

[11] H.-B. Zhang, Y.-X. Zhang, B. Zhong et al., "A comprehensive survey of vision-based human action recognition methods," *Sensors*, vol. 19, no. 5, 2019.

[12] S. A. R. Abu-Bakar, "Advances in human action recognition: an updated survey," *Iet Image Processing*, vol. 13, no. 13, pp. 2381–2394, 2019.

[13] B. Sun, D. Kong, W. Zhang, and W. Jia, "Survey on human action recognition from depth maps," *Journal of Beijing University of Technology*, vol. 44, no. 10, pp. 1353–1368, 2018.

[14] M. Koohzadi and N. M. Charkari, "Survey on deep learning methods in human action recognition," *Iet Computer Vision*, vol. 11, no. 8, pp. 623–632, 2017.

[15] C. Chen, R. Jafari, and N. Kehtarnavaz, "A survey of depth and inertial sensor fusion for human action recognition," *Multimedia Tools and Applications*, vol. 76, no. 3, pp. 4405–4425, 2017.

[16] B. Liu, H. Cai, Z. Ju, and H. Liu, "RGB-D sensing based human action and interaction analysis: a survey," *Pattern Recognition*, vol. 94, pp. 1–12, 2019.

[17] J. Zhang, W. Li, P. O. Ogunbona, P. Wang, and C. Tang, "RGB-D-based action recognition datasets: a survey," *Pattern Recognition*, vol. 60, pp. 86–105, 2016.

[18] S. P. Sahoo and S. Ari, "Depth estimated history image based appearance representation for human action recognition," in *Proceedings of the 2019 IEEE Region 10 Conference: Technology, Knowledge, and Society, TENCON 2019*, Institute of Electrical and Electronics Engineers Inc., Kerala, India, pp. 965–969, 2019.

[19] Y. L. Li, G. J. Wang, X. G. Lin, G. Cheng, and L. He, "Real-time human action recognition system using depth map

sequences," in *Proceedings of the 2nd International Conference on Opto-Electronics Engineering and Materials Research, OEMR 2013*, Trans Tech Publications Ltd, Zhengzhou, Henan, China, pp. 1647–1651, 2013.

[20] W. Xu, M. Wu, M. Zhao, Y. Liu, B. Lv, and T. Xia, "Human action recognition using multilevel depth motion maps," *IEEE Access*, vol. 7, pp. 41811–41822, 2019.

[21] C. Chen, K. Liu, and N. Kehtarnavaz, "Real-time human action recognition based on depth motion maps," *Journal of Real-Time Image Processing*, vol. 12, no. 1, pp. 155–163, 2016.

[22] X. Wan, T. Xing, Y. Ji, S. Gong, and C. Liu, "3D human action recognition with skeleton orientation vectors and stacked residual Bi-LSTM," in *Proceedings of the 4th Asian Conference on Pattern Recognition, ACPR 2017*, Institute of Electrical and Electronics Engineers Inc., Nanjing, China, pp. 577–582, 2017.

[23] J. Liu, G. Wang, L.-Y. Duan, K. Abdiyeva, and A. C. Kot, "Skeleton-based human action recognition with global context-aware attention LSTM networks," *IEEE Transactions on Image Processing*, vol. 27, no. 4, pp. 1586–1599, 2018.

[24] T. Liu, J. Wang, S. Hutchinson, and M. Q.-H. Meng, "Skeleton-based human action recognition by pose specificity and weighted voting," *International Journal of Social Robotics*, vol. 11, no. 2, pp. 219–234, 2019.

[25] C. N. Phyo, T. T. Zin, and P. Tin, "Skeleton motion history based human action recognition using deep learning," in *Proceedings of the 6th IEEE Global Conference on Consumer Electronics, GCCE 2017*, Institute of Electrical and Electronics Engineers Inc., Nagoya, Japan, pp. 1-2, 2017.

[26] H. Zhang, P. Zhong, J. He, and C. Xia, "Combining depth-skeleton feature with sparse coding for action recognition," *Neurocomputing*, vol. 230, pp. 417–426, 2017.

[27] N. El Din El Madany, Y. He, and L. Guan, "Human action recognition by fusing deep features with globality locality preserving canonical correlation analysis," in *Proceedings of the 24th IEEE International Conference on Image Processing*, IEEE Computer Society, Beijing, China, pp. 2871–2875, 2017.

[28] Y. Guo, D. Tao, W. Liu, and J. Cheng, "Multiview cauchy estimator feature embedding for depth and inertial sensor-based human action recognition," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 4, pp. 617–627, 2017.

[29] S. Asteriadis and P. Daras, "Landmark-based multimodal human action recognition," *Multimedia Tools and Applications*, vol. 76, no. 3, pp. 4505–4521, 2017.

[30] Z. Gao, J.-M. Song, H. Zhang, A.-A. Liu, Y.-B. Xue, and G.-P. Xu, "Human action recognition via multi-modality information," *Journal of Electrical Engineering and Technology*, vol. 9, no. 2, pp. 739–748, 2014.

[31] B. Seddik, S. Gazzah, and N. Essoukri Ben Amara, "Human-action recognition using a multi-layered fusion scheme of Kinect modalities," *Iet Computer Vision*, vol. 11, no. 7, pp. 530–540, 2017.

[32] F. Malawski and B. Kwolek, "Improving multimodal action representation with joint motion history context," *Journal of Visual Communication and Image Representation*, vol. 61, pp. 198–208, 2019.

[33] J. Imran and B. Raman, "Evaluating fusion of RGB-D and inertial sensors for multimodal human action recognition," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 1, pp. 189–208, 2020.

[34] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, IEEE, New York, NY, USA, pp. 886–893, 2005.

[35] Y. Pang, Y. Yuan, X. Li, and J. Pan, "Efficient HOG human detection," *Signal Processing*, vol. 91, no. 4, pp. 773–781, 2011.

[36] C. Harris, "A combined corner and edge detector," in *Proceedings of the 4th Alvey Vision conference*, pp. 147–151, New York, NY, USA, 1988.

[37] I. Laptev and T. Lindeberg, "Space-time interest points," in *Proceedings of the 9th IEEE International Conference on Computer Vision*, IEEE, Nice, France, pp. 432–439, 2003.

[38] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *Proceedings of the 2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, IEEE, Beijing, China, pp. 65–72, 2005.

[39] B. Lucas and T. Kanade, "An iterative image registration technique with an application toStereo vision," in *Proceedings of the $7th$ International Joint Conference on ArtificialIntelligence*, Beijing, China, 1997.

[40] G. Johansson, "Visual motion perception," *Scientific American*, vol. 232, no. 6, pp. 76–88, 1975.

[41] T. G. Dietterich, "Machine-learning research," *AI Magazine*, vol. 18, no. 4, pp. 97–136, 1997.

[42] J. Liu, J. Luo, and M. Shah, "Recognizing realistic actions from videos "in the wild"" in *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Beijing, China, pp. 1996–2003, 2009.

[43] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Beijing, China, pp. 1–8, 2008.

[44] K. Rapantzikos, Y. Avrithis, and S. Kollias, "Dense saliency-based spatiotemporal feature points for action recognition," in *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Beijing, China, pp. 1454–1461, 2009.

[45] M. D. Rodriguez, J. Ahmed, and M. Shah, "Action mach a spatio-temporal maximum average correlation height filter for action recognition," in *Proceedings of the 2008 IEEE conference on computer vision and pattern recognition*, IEEE, Beijing, China, pp. 1–8, 2008.

[46] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.