Deep Convolutional Neural Networks for Human Action Recognition Using Depth Maps and Postures

Aouaidjia Kamel, Bin Sheng, Po Yang, Ping Li, Ruimin Shen, and David Dagan Feng, *Fellow, IEEE*

***Abstract*—In this paper, we present a method (Action-Fusion) for human action recognition from depth maps and posture data using convolutional neural networks (CNNs). Two input descriptors are used for action representation. The first input is a depth motion image that accumulates consecutive depth maps of a human action, whilst the second input is a proposed moving joints descriptor which represents the motion of body joints over time. In order to maximize feature extraction for accurate action classification, three CNN channels are trained with different inputs. The first channel is trained with depth motion images (DMIs), the second channel is trained with both DMIs and moving joint descriptors together, and the third channel is trained with moving joint descriptors only. The action predictions generated from the three CNN channels are fused together for the final action classification. We propose several fusion score operations to maximize the score of the right action. The experiments show that the results of fusing the output of three channels are better than using one channel or fusing two channels only. Our proposed method was evaluated on three public datasets: 1) Microsoft action 3-D dataset (MSRAction3D); 2) University of Texas at Dallas-multimodal human action dataset; and 3) multimodal action dataset (MAD) dataset. The testing results indicate that the proposed approach outperforms most of existing state-of-the-art methods, such as histogram of oriented 4-D normals and Actionlet on MSRAction3D. Although MAD dataset contains a high number of actions (35 actions) compared to existing action RGB-D datasets, this paper surpasses a state-of-the-art method on the dataset by 6.84%.**

***Index Terms*—Action recognition, convolutional neural networks (CNNs), depth motion image (DMI), moving joints descriptor (MJD).**

Manuscript received January 30, 2018; accepted June 8, 2018. Date of publication July 11, 2018; date of current version August 16, 2019. This work was supported in part by the National Natural Science Foundation of China under Grant 61572316 and Grant 61671290, in part by the Research Grants Council of Hong Kong under Grant 28200215, in part by the National High-Tech Research and Development Program of China (863 Program) under Grant 2015AA015904, in part by the Key Program for International S&T Cooperation Project of China under Grant 2016YFE0129500, in part by the Science and Technology Commission of Shanghai Municipality under Grant 16DZ0501100 and Grant 17411952600, and in part by the Interdisciplinary Program of Shanghai Jiao Tong University under Grant 14JCY10. This paper was recommended by Associate Editor K. Panetta. *(Corresponding author: Bin Sheng.)*

A. Kamel, B. Sheng, and R. Shen are with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: shengbin@sjtu.edu.cn; rmshen@cs.sjtu.edu.cn).

P. Yang is with the Department of Computer Science, Liverpool John Moores University, Liverpool L3 3AF, U.K. (e-mail: p.yang@ljmu.ac.uk).

P. Li is with the Faculty of Information Technology, Macau University of Science and Technology, Macau 999078, China (e-mail: pli@must.edu.mo).

D. D. Feng is with the Biomedical and Multimedia Information Technology Research Group, School of Information Technologies, University of Sydney, Sydney, NSW 2006, Australia (e-mail: dagan.feng@sydney.edu.au).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMC.2018.2850149

I. Introduction

HUMAN action recognition is necessary for various computer vision applications that demand information about people's behavior, including surveillance for public safety, human–computer interaction applications, and robotics [1]–[6]. There are a variety of human action recognition systems, such as video-based human action recognition [7]–[10], wearable sensor-based human action recognition [11]–[15], wireless sensor network-based human action recognition [16], [17], etc. Among these studies, due to high recognition accuracy and easy deployment, video-based human action recognition techniques have got more research attention and been widely applied into lots of industrial applications.

Traditionally, video-based human action recognition methods are mostly based on processing sequences of two-dimension (2-D) RGB color images by utilizing classifiers like HMM [18], KNN [19], template matching [20], dynamic Bayesian network [21], SVM [7], etc., into global or local representations like blob feature, motion energy image, optical flow, etc. While these methods enable delivering up to 97% accuracy in recognizing simple human actions like running, bending, and hand waving on KTH dataset [22] for example with a simple background, they are quite sensitive to influencing factors on the quality of RGB images, such as complex background, illumination variation, and clothing color, which makes it difficult to segment the human body in every scene. Additionally, semantically equivalent actions can be performed in various ways of body movements by each individual. On the other hand, two different actions having a similar trajectory of motion make it more difficult to distinguish correctly. The lack of depth cues in colored images could lead to significant degradation of discriminating capability of an action recognizer and has a negative impact on recognizing the action, especially when it is performed in the camera direction. In order to overcome above limitations, recent human action recognition technologies [23]–[27] have considered involving depth cameras to provide three-dimension (3-D) depth data in a form of RGB-D images with illumination invariant, uniform color, and depth information that eases the ambiguity of human's motion. In order to accurately estimate the postures of the human body skeleton joints and further recognizing human actions, several human motion capture systems are built with multiple sensory data from depth cameras, RGB cameras, or wearable devices.

Among these motion capture systems, due to the availability of cost-effective devices like Kinect [28] and expressive

2168-2216 © 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

Authorized licensed use limited to: ShanghaiTech University. Downloaded on August 15,2026 at 17:52:04 UTC from IEEE Xplore. Restrictions apply.

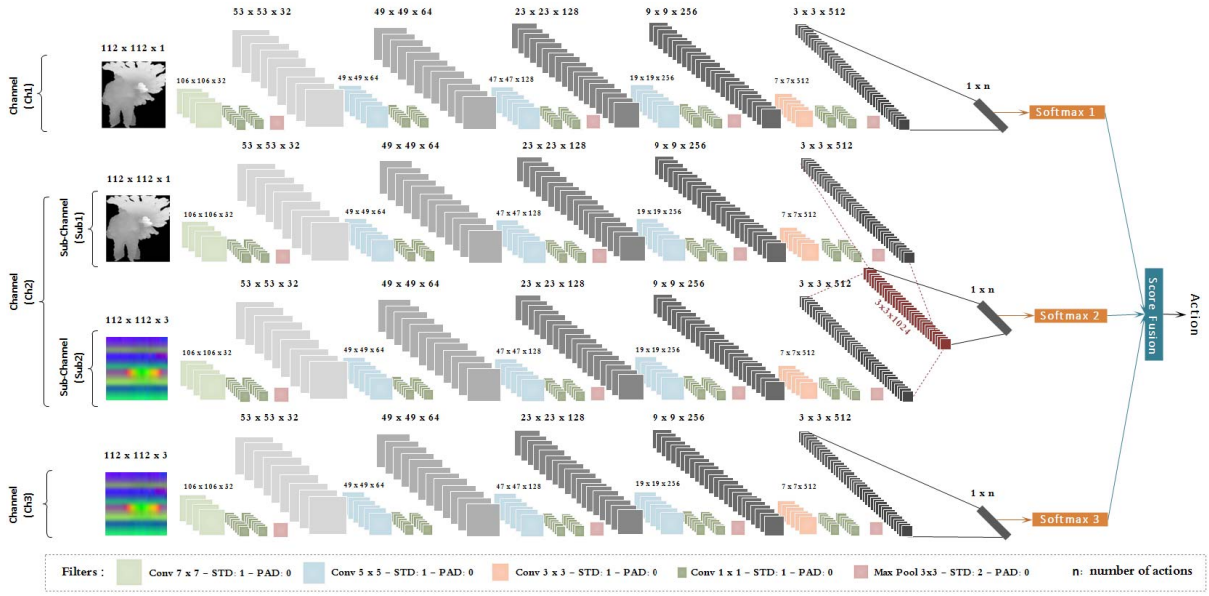


Fig. 1. Our proposed three channels CNN model for action recognition. Gray layers represent the feature maps outputs after applying convolutional and max-pooling operations. The size of the output feature maps is shown on the top of layers. STD: stride, PAD: padding.

features provided by depth maps and body postures, using depth maps or body postures to represent the human motion for action recognition became quite popular. But they also have some limitations on existing techniques. First of all, traditional depth map data-based human action recognition usually needs to build up multiview depth map dataset and extract a large volume of features in order to provide a distinctive representation of each human action for classification. For instance, two actions may look similar from the front view, but they have a different appearance from side views. While utilizing some feature extraction [29] from multiview, it might be possible to identify these two actions. The process of building up multiview camera and collecting sufficient features is quite time-consuming. Second, using human body posture data for human action representation is quite sensitive to the joints movement. It is very difficult to find two human actions that have similar joints coordinates during their motion, which may reflect on recognizing two semantically equivalent actions as different actions when they are performed in a slightly different way. Finally, existing methods of using depth maps or body postures data usually adapt traditional classifiers like SVM, which requires handcraft feature extraction. However, recently, deep learning and especially convolutional neural network (CNN) which was inspired by the human visual cortex hierarchic processing, have made a huge success in image classification [30], [31]. Consequently, regarding above considerations, this paper motivates to investigate a new method of fusing single depth map and body postures for achieving more cost-effective and accurate human activity recognition.

In this paper, we propose a new method (Action-Fusion) for human action recognition from depth maps and posture data using three channels of a deep CNN model. CNN is a powerful technique for both feature extraction and classification, which can automatically learn discriminative features from a training data. Using CNN as a tool for extracting features from action representations would be of a great advantage for action recognition. As a matter of fact, depth maps features are convenient to classify semantically equivalent actions when they are performed in a slightly different way. The previous analysis inspires as to think that a better approach for action recognition should be based on using the two types of data to balance the use of features by strengthen the weak part in each type by the strong part in the other, and come up with a robust action representation that can be used to classify actions accurately.

In this paper, two descriptors are used to represent the human actions, depth motion image (DMI) descriptor to represent the depth maps sequence and moving joints descriptor (MJD) to represent body postures sequence. The DMI descriptor is employed in a different way from the one in [29]. In their work, they calculated the descriptor from the front view and side view, then the results are used to calculate another two descriptors. However, in this paper, we prove that using the descriptor from the front view only with the help of MJD is enough to generate state-of-the-art results and hence, with less computation complexity. The DMI assembles depth maps of an action in order to capture the changing in depth of human motion. Our computation method is based on calculating the changing in depth of all the action frames at once rather than calculating it between two frames sequentially. MJD is inspired by [32]. In their work, they used Cartesian coordinates representation, however, in this paper, we propose a robust representation of the body joints movement over time using spherical coordinates instead of directly using Cartesian coordinates. The motivation behind choosing spherical coordinates for modeling the motion is that the human body joints generally move around a fixed point of the body hip center in a circular manner. The changing in the angles provides further information about the joints movement direction, unlike

Authorized licensed use limited to: ShanghaiTech University. Downloaded on August 15,2026 at 17:52:04 UTC from IEEE Xplore. Restrictions apply.

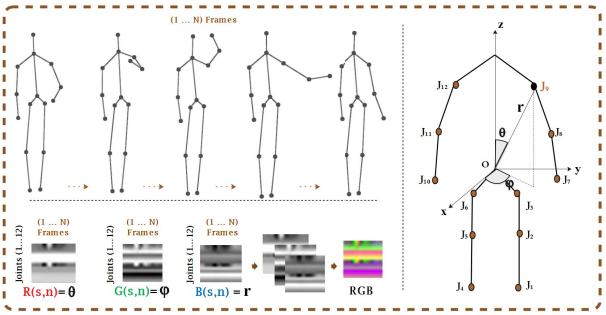


Fig. 2. MJD. An example of draw circle action from the MSRAction3D dataset. Left-Top: skeleton sequence. Left-Bottom: the creation of RGB MJD image, where N is the total number of frames, s is the joint number, and n is the frame number. Right: skeleton model shows the three spherical coordinates of joint j_9.

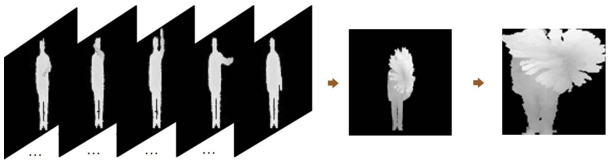


Fig. 3. DMI. An example of draw circle action from MSRAction3D dataset. Left: depth maps sequence. Middle: DMI. Right: cropped ROI and resized to 112×112.

Cartesian coordinates representation that provides only the changing in the joints position.

The action recognition process introduced in this paper involves three CNN channels trained with DMI and MJD descriptors for feature extraction and classification. The first channel is trained with DMI, the second channel is a connection between two subchannels. One subchannel is trained with DMI and the other subchannel is trained with MJD. The third channel is trained with MJD only. Each channel generates its own scores for the actions. Our experiments reported that taking the maximum score value of the three CNN channels leads to low accuracy prediction on the testing data. In order to maximize the score of the right action, five score operations are proposed and analyzed to select the best operation that can predict the right action accurately. In general, the proposed approach generates three outputs from the CNN channels and five other outputs produced by fusion operations between the three channels. The maximum action score value of all the outputs is considered as the final action prediction result. The results generated from fusing the three CNN channels are better than the ones generated using a single channel or two fused channels only. In fact, each channel learns features that cannot be seen in the other channels, which make combining them together produce better results. The experimental results of the proposed approach are compared with the state-of-the-art methods on three public datasets: 1) Microsoft Action 3-D dataset (MSRAction3D); 2) University of Texas at Dallas-multimodal human action dataset (UTD-MHAD); and 3) multimodal action dataset (MAD) dataset. The comparison outcomes proved that the action recognition accuracy is better than most of existing methods, and proved also that the recognition accuracy is stable even with a large number of actions, such as MAD dataset. The contributions of our proposed work can be summarized as follows.

1) An effective three channels deep CNNs method is proposed by using depth maps and posture data. This method is beneficial to strengthen the weaknesses of using one type of data for action recognition. A thorough performance evaluation of the proposed method with three general datasets has been carried out. The results suggest that the proposed method can effectively and efficiently recognize human actions with an improved accuracy over existing state-of-the-art methods, such as [23], [24], and [33].
2) A new MJD descriptor is proposed to represent joints movement in a form of spherical coordinates. The descriptor provides essential information on joints movement directions from the size of angles in addition to the changing in joints poses. A DMI descriptor is also used to represent the changing in action depth from the front view only rather than two views, such as [29]. The MJD representation can replace efficiently the missing side views with its informative representation that has a great influence on boosting whole accuracy. Fig. 1 shows how the two descriptors processed by the three CNN model channels. Figs. 2 and 3 illustrate the transformation of raw data to the MJD and DMI, respectively.
3) Score fusion operations are proposed for predicting the right action from three CNN channels of the trained model. Each of the three channels generates a score for each action. Usually, the highest score action represents the right action. It is possible to have two or three channels generate a highest score for different actions, and we cannot decide which channel must be considered. On the other hand, the right action may have a lower score than the ones generated from the CNN channels. The role of fusion operation is to maximize the score of the right action whatever the prediction of the three channels.
4) A large training data is one of the key success factors for a CNN model prediction accuracy. Due to the lack of a large RGB-D action recognition dataset, the two action representations help to reinforce the learning process on a small amount of data, which reflects less training computation time with a high learning accuracy.

II. Related Work

Recently, action recognition in the robotic domain has become popular because of the need for human–robot interaction. It requires using different types of features for action representation due to the diversity of human actions in the wild. The work in [34] investigated the problem of first-person interaction activity recognition using a dataset collected with a robot during the interaction with humans. Their method concatenated different types of descriptors to recognize human activities. Ryoo *et al.* [35] extended the dataset presented in [34] for early activity recognition to avoid harmful actions during the interaction with the robot. Gori *et al.* [36]

Authorized licensed use limited to: ShanghaiTech University. Downloaded on August 15,2026 at 17:52:04 UTC from IEEE Xplore. Restrictions apply.

applied state-of-the-art action recognition methods to classify challenging human actions of a dataset collected with a moving robot in the wild, which includes actions performed spontaneously in different scales, with occlusion, and with multiple people. Duckworth *et al.* [37] also used a mobile robot for activity recognition from long-term observation by encoding sequences of skeleton joints to a qualitative spatial representation.

Several recent depth-based approaches have been reported to improve human action recognition accuracy. An action graph based on a sampled 3-D representation from depth maps to model the human motion is proposed in [25]. Several four-dimension (4-D) descriptors have been used to represent human actions. In [23], a histogram of oriented 4-D normals (HON4D) is used in order to describe the action in 4-D space, including spatial coordinates, depth, and time. Vieira *et al.* [38] also represented the depth sequence in 4-D grids by dividing the space and time axis into multiple segments. Another 4-D descriptor proposed by [39] called random occupancy pattern, which deals with noise and occlusion to increase the robustness. Action recognition from different views has been applied to gain more discriminative features. Kim *et al.* [29] generated a side view from the front view of a depth map. Both views are transformed into depth motion appearance and depth motion history (DMH) descriptors, then SVM is trained with the two descriptors to classify the action. Recently [40] generates top and side views by rotating 3-D points from the front view. The three views are used as inputs to three CNN models for feature extraction and classification.

In parallel to depth-based approaches, skeleton-based methods also have a huge contribution to action recognition research. In [24], each joint is associated with a local occupancy pattern descriptor, which is translation invariant and provides highly discriminative features. They also proposed a temporal motion representation called Fourier temporal pyramid in order to model the joints movement. EigenJoints is a new type of features in [41] to combine action information, including static postures, motion, and offset features. A framework based on sparse coding and temporal pyramid matching is proposed in [42] for better 3-D joint features representation. A histogram of 3-D joint location called HOJ3D in [26] represents the human joints locations, then posture words are built from HOJ3D vectors and trained using a hidden Markov model to classify the actions. In [27], a framework is proposed for online human action recognition using a new structured streaming skeletons feature, which can deal with intraclass variations, including viewpoint, anthropometry, execution rate, and personal style. Zanfir *et al.* [43] proposed nonparametric moving pose (MP) for low-latency human action and activity recognition, the framework considers pose information, speed, and acceleration of the joints in the current frame within a time window. A hierarchical dynamic framework was reported in [44] based on using deep belief networks for feature extraction and encoding dynamic structure into an HMM-based model. Wang *et al.* [45] addressed the action recognition in videos by modeling the spatial–temporal structures of human poses. The method improves the pose estimation first, then groups the joints into five body parts and applies data mining techniques to get spatial–temporal pose structures for action representation. Du *et al.* [32] and Ke *et al.* [46] transformed the joint coordinates into a 2-D image descriptor to classify the actions using CNNs. Very recent works (SOS) [33] and joint trajectory maps [40] propose a new approach which transforms the skeleton joints trajectories shapes from 3-D space into three images that represent the front view, the top view, and the side view of the joints' trajectory shapes. Three CNNs are used to extract features from the three images to classify the actions.

CNN [47] is a powerful technique for feature extraction and classification. Recent action recognition approaches started to focus more on using CNN for action classification rather than using SVM. Researchers in deep learning try always to come up with new techniques to improve the CNN architectures and enhance the performance of feature extraction, classification, and computation speed. Gu *et al.* [48] summarized recent advances in CNNs in term of regularization, optimization, activation functions, loss functions, weight initialization, etc. Recent CNN-based action recognition methods are based on using multiple action representations that employ many CNN channels for the processing. In [49], many feature concatenation architectures are proposed in order to improve the classification accuracy using multiple sources of knowledge.

Although the previous approaches achieved good results, the problem of action recognition is still open and requires more robust action representation and feature extraction techniques to improve the accuracy and overcome the weaknesses of the previous methods. To this end, the proposed work in this paper investigates the use of both types of data: depth maps and postures to enhance the action recognition throw using CNN for feature extraction and classification.

III. Action Recognition Method

Our action recognition framework is shown in Fig. 4. We use two types of data sequence for action representation: 1) depth maps and 2) body postures. Each of the two inputs is transformed into a descriptor that assembles input sequence in one image, namely DMI for depth maps and MJD for body postures. A model of three CNN channels of the same structure is trained and tested with the two descriptors. We propose several score fusion operations to get a high score of correct action by combining the prediction scores of the three CNN channels.

A. Data Preprocessing

1) Depth Motion Image: The DMI describes the overall action appearance by accumulating all depth maps of the action over time to generate a uniform representation that can define each action with its own specific appearance from the front view. It captures the changing in depth of the moving body parts. The DMI representation provides distinctive features for each action which ease the feature extraction task for the CNN model. The following equation illustrates

Authorized licensed use limited to: ShanghaiTech University. Downloaded on August 15,2026 at 17:52:04 UTC from IEEE Xplore. Restrictions apply.

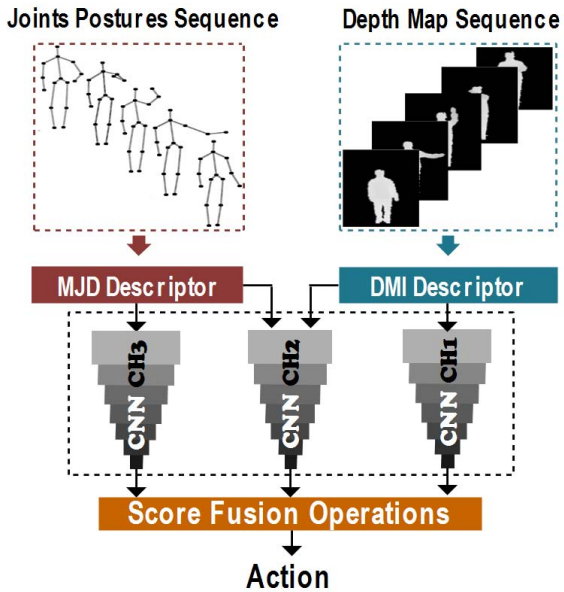


Fig. 4. Framework of our proposed action recognition method. Depth maps sequence is transformed into a DMI descriptor and postures sequence is transformed into an MJD. A model of three CNN channels (Ch1, Ch2, and Ch3) extracts features from the descriptors. Score fusion operations are used to maximize the score of the right action from the three CNN prediction outputs.

the calculation of DMI:

$$\mathrm{DMI}(i,j) = 255 - \min(I(i,j,t)) \quad \forall t \in [k...(k+N-1)] \tag{1}$$

where $I(i,\ j,\ t)$ is the pixel position $(i,\ j)$ of the frame I at time t, DMI is a gray image (8 bits) that represents the depth difference from frame k to $k + N-1$, and N represents the total number of frames. The pixel value of DMI image is the minimum value of the same pixels position of the depth maps sequence. The resulting image is normalized by dividing each pixel value by the maximum value of all pixels in the image, then the region of interest (ROI) is cropped to get rid of uninformative black pixels. Fig. 3 shows a draw circle action sequence with its DMI and Fig. 5(top) shows seven DMI actions samples created from the MSRAction3D dataset.

2) Moving Joints Descriptor: From the 20 joints of the skeleton model provided by the datasets, only 12 most informative joints are selected. Fig. 2(right) shows the joints selected for the processing. In order to make the hip center joint O the origin of the system, we subtract its coordinates from each of the 12 joints coordinates. The posture data provided by the datasets are presented in a form of Cartesian coordinates (x, y, z). However, the action representation using Cartesian coordinates is sensitive to joints movement, which may reflect on representing two semantically equivalent actions as different actions. The movement of human body joints during the motion is subject to some restrictions. They cannot move farther than a limited distance from the hip center joint. Furthermore, each joint has a limited range of angle to move. Those restrictions can be modeled by spherical coordinates as presented in Fig. 6, which shows an example of joints movement during a running action and its representation in spherical coordinates. The distance r represents how is the joint far from the hip center O. The angles θ and ϕ are useful to indicate the movement direction of the joint.

In order to construct the MJD from spherical coordinates, the Cartesian coordinates of joints are transformed to spherical coordinates with taking into consideration the hip center joint O as the origin of the system. The transformation is described in (2) and (3). In spherical coordinates system, the joint motion is subject to three metrics, the angle θ represents the vertical angle of the joint with the z-axis, the angle ϕ represents the horizontal angle with the x-axis, and the radius r represents the distance between the origin and the joint. For the sake of capturing the change in the spherical coordinates over time, three gray images R, G, and B are constructed to represent the motion of θ, ϕ, and r, respectively. The rows number of each image represents the joints number, the columns number represents the frames number of the action, and the pixel value is the coordinate of the joint J_k in the frame n as illustrated in (4). Finally, an RGB image is constructed by combining the three gray images together to produce the finale descriptor image. Fig. 2(left) illustrates the construction of MJD and Fig. 5(bottom) shows seven MJD samples created from the MSRAction3D dataset

$$\text{Joints} = \{J_1, \ldots, J_k, \ldots, J_{12}\}, \quad J_k = (\theta, \phi, r) \tag{2}$$

$$r = \sqrt{x^2 + y^2 + z^2}, \quad \theta = \arccos\frac{z}{r}, \quad \phi = \arctan\frac{y}{x} \tag{3}$$

$$\begin{aligned} R(J_k, n) &= \{\theta : \theta \text{ of the joint } J_k \text{ in frame } n\} \\ G(J_k, n) &= \{\phi : \phi \text{ of the joint } J_k \text{ in frame } n\} \\ B(J_k, n) &= \{r : r \text{ of the joint } J_k \text{ in frame } n\} \\ \mathrm{MJD} &= R + G + B \end{aligned} \tag{4}$$

where x, y, and z are the Cartesian coordinates. θ, ϕ, and r are the spherical coordinates. $k = \{1, 2, \ldots, 12\}$ is the joint number. R, G, and B are gray images, and MJD is the RGB moving joints descriptor image.

B. Convolutional Neural Network Model

1) Model Description: After the data preprocessing task, the two descriptors DMI and MJD are resized to 112×112 and used as inputs to the CNN model. The model is composed of convolutional layers for feature extraction and pooling layers for dimensionality reduction. Thirty two convolutional filters of size 7×7 are used in the first convolutional layer and three 5×5 convolutional filters are used in the second, third and the forth convolutional layers with 64, 128, and 256 filters number, respectively. The last convolutional layer applies 512 filters with a size of 3×3. Each of the convolutional layers mentioned before is followed by a "network in network" structure proposed by [50], which is based on using 1×1 convolutional filters with a larger number than the previous layer's filters. This structure makes the model deeper and has more parameters without completely changing the network structure, and with cheap computation cost. However, in our CNN model, we use the same number of 1×1 convolutional filters as the previous layer. During the training experiments, we

Authorized licensed use limited to: ShanghaiTech University. Downloaded on August 15,2026 at 17:52:04 UTC from IEEE Xplore. Restrictions apply.

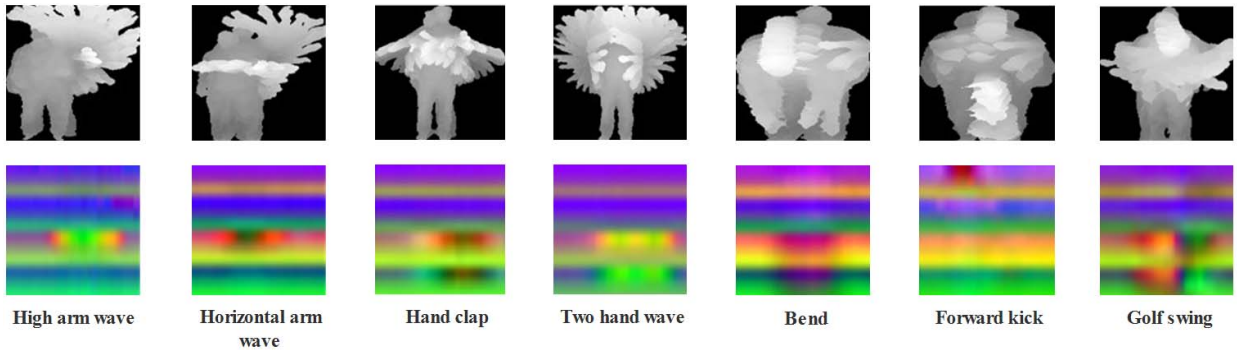


Fig. 5. Preprocessing results of seven actions samples from the MSRAction3D dataset. Top: DMI descriptors. Bottom: MJD descriptors.

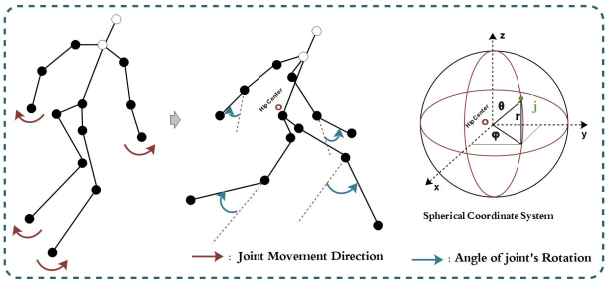


Fig. 6. Human body joints motion direction during a running action. The joints motion is more subject to a rotation, which makes the spherical coordinate system suitable to represent the joints movement.

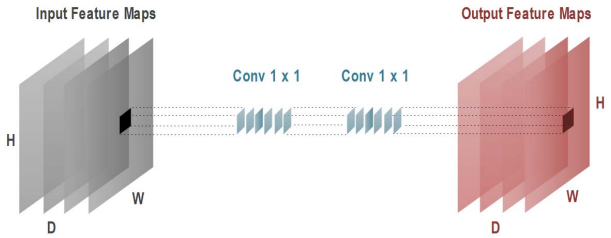


Fig. 7. Network block structure used to improve the CNN model performance accuracy with less computation cost. H, W, and D refer to height, width, and depth of the feature maps.

found that using 1×1 convolutional filters without increasing the number, improves the accuracy without a noticeable influence on the computation time. Fig. 7 shows how the two 1×1 convolutional layers are used. The size of the output feature map after using two 1×1 convolution layers is the same as the input size.

Three max-pooling layers of 3×3 filter size are used for dimensionality reduction. Each convolutional layer in the model is followed by rectified linear units activation function. A fully connected layer with a size equals the number of actions is used as the result of feature extraction. Fig. 1 describes the network architecture, including layers output sizes, and filters. A multinomial logistic loss function is applied with stochastic gradient descent algorithm to update the weights during the training process. The textures of the two input descriptors make it difficult to capture distinctive features when the convolutional operation is applied with small filter size. For example, the application of 3×3 filters on the input image at the very beginning is not efficient because two images that represent different actions may have similar features in a 3×3 region, which is the reason behind using 7×7 filters and 5×5 filters in the first convolutional layers. Usually, CNN architectures end up with one or two fully connected layers before the last classifier layer. However, in our model, and according to the training experiments, we found that using only one fully connected layer as a classifier after the pooling layer generates better results. At the testing phase, softmax regression layer is used to generate a score for each class based on the trained weights.

2) Model Training: The CNN model described previously is employed in three different training channels. We denote channel 1 by Ch1, channel 2 by Ch2, and channel 3 by Ch3. The channel Ch1 is trained with DMI descriptors, the channel Ch2 is trained with DMI and MJD descriptors together, and channel Ch3 is trained with MJD only. The Channel Ch2 is a composition of two others subchannels, Sub1 and Sub2. Each of the two subchannels is trained with one kind of descriptors, namely Sub1 is trained with DMI descriptors and Sub2 is trained with MJD descriptors. The two subchannels are concatenated after the last pooling layer, which results in a new layer of depth size equals the sum of the two previous pooling layers outputs of the subchannels. The concatenation operation was inspired by [49], which propose different concatenation methods based on fusing the last fully connected layers. However, we found that the concatenation of pooling layers outputs is more efficient in term of accuracy. The three channels mentioned before are trained together at the same time with the same parameters.

We initialized the learning rate with 0.01, the weight decay with 0.0005 and the momentum with 0.9, which are the same parameters used in AlexNet [30]. However, this initialization caused unstable behavior in the loss function. Since the learning rate has a big impact on the training process, we fixed the weight decay and the momentum, then we decreased the learning rate several times until 0.0008, which generates a stable decrease in the loss function. A lower learning rate value makes the loss still stable but with a slower decrease. We initialized the weights with [51], and we trained the model with a batch size of 50 images for each of the two descriptors in all the datasets. The number of iterations required for each channel to reach the minimum loss function value differs from one channel to another depending on the input data features and the dataset size. The network in Fig. 1 is designed, trained, and

Authorized licensed use limited to: ShanghaiTech University. Downloaded on August 15,2026 at 17:52:04 UTC from IEEE Xplore. Restrictions apply.

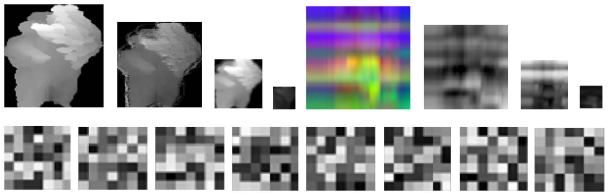


Fig. 8. Top: samples of feature maps generated from the layers of the channel Ch2 of the CNN model (DMI features from subchannel Sub1 and MJD features from subchannel Sub2). Bottom: samples of trained 7 × 7 filters.

TABLE I
SCORE FUSION OPERATIONS ON THE THREE CNN CHANNELS

Fusion	Operation
Fus1	$Prod(Sfm1, Sfm3)$
Fus2	$Prod(Sfm1, Sfm2)$
Fus3	$Max(Sfm2, Sfm3)$
Fus4	$Prod(Sfm1, Sfm2, Sfm3)$
Fus5	$Prod(Prod(Sfm1, Sfm2, Sfm3), Max(Sfm1, Sfm3))$

tested using caffe deep learning framework [52]. Fig. 8(top) shows feature maps samples of an input pair of DMI and MJD descriptors generated from the layers of the channel CH2, and Fig. 8(bottom) shows samples of 7 × 7 trained filters of the first convolutional layer.

C. Score Fusion

The output of softmax layer is a vector of length equals the number of actions (5), where each element represents the probability of the input image to be a specific action. When we tested our trained model on the softmax output of each CNN channel separately, we found that in most cases, the maximum value corresponds to the correct action. However, for some test samples, the maximum value does not represent the correct action. A lower probability value than the maximum may correspond to the correct prediction. In order to improve the prediction accuracy of the data samples that generate wrong classification results, the softmax outputs of the three CNN channels are fused. In the testing experiments, many fusion alternatives have been tried, such as element-wise averaging, maximum, addition, and product, but the maximum and product operations which we denote Max and Prod generate better results than the other operations.

As we will see in the experimental results section, the classification accuracy does not only depends on the operation Max or Prod, but it also depends on the channels involved in the computation. For example, the result of Prod operation between softmax output Sfm1 of channel Ch1 and Sfm2 of channel Ch2 is different when it is performed between Sfm1 of channel Ch1 and Sfm3 of channel Ch3, or between the three channels outputs, Sfm1, Sfm2, and Sfm3. While the accuracy varies according to operation types and channel types, different fusion operations are proposed and summarized in Table I. In total, we have eight possible predictions in our proposed approach: three from the CNN channels (Sfm1, Sfm2, Sfm3) and five from the fused channels (Fus1, Fus2, . . . , Fus5). The final classification result is the maximum value of the eight outputs as shown in (6).

The motivation behind the model fusion architecture described in Fig. 1 is that the channel Ch1 provides features related to the overall action appearance, which is useful to recognize the action even when it is performed slightly in a different way. While the channel Ch3 features are sensitive to the joints movement, it is rare when we find two actions have similar features, even when they are semantically equivalent. The channel Ch2 provide features that balance between the two representations. Additionally, in the case when some of the joints are missed because they are not captured by the system, which will be reflected on missing features in the MJD, the fusion of MJD and DMI features could help the network to recognize the actions even though some MJD features are missed. In fact, the DMI features compensate for the missing features of MJD

$$\begin{aligned} \text{Sfm1} &= \{p_{11}, \ldots, p_{c_1 a}, \ldots, p_{1A}\} \\ \text{Sfm2} &= \{p_{21}, \ldots, p_{c_2 a}, \ldots, p_{2A}\} \\ \text{Sfm3} &= \{p_{31}, \ldots, p_{c_3 a}, \ldots, p_{3A}\} \end{aligned} \tag{5}$$

where Sfm1, Sfm2, and Sfm3 are the softmax layer outputs of channel Ch1, Ch2, and Ch3, respectively. $p_{c_1 a}$, $p_{c_2 a}$, and $p_{c_3 a}$ represent the probability of an action $\boldsymbol{a}$ to be the correct class in channel Ch1, Ch2, and Ch3, respectively. A is the total number of actions

$$\text{Action} = \text{Max}(\text{Sfm1}, \text{Sfm2}, \text{Sfm3}, \text{Fus1}, \ldots, \text{Fus5}) \tag{6}$$

$$\tag{7}$$

where Action represents the action of the highest score, which is the final classification result.

IV. EXPERIMENTAL RESULTS

As most commonly used RGB-D human action recognition datasets [53], we chose three datasets to evaluate the performance of our proposed method, MSRAction3D [25], UTD-MHAD [54], and MAD [55]. The datasets provide depth maps and posture data which are suitable to construct the DMI and MJD descriptors. We follow the same testing settings of the state-of-the-art methods to compare our proposed approach with the previous ones. A set of testing experiments were conducted on the three CNN channels, including the evaluation of each channel separately and the combination of the channels together based on the fusion operations. Although the results of the fused channels vary from a dataset to another. Generally, the classification results of using MJD in channel Ch3 are better than using DMI in channel Ch1 on the three datasets, which reflects the high performance of using posture representation over depth representation. However, the performance of channel Ch2 using both representations DMI and MJD is better than Ch1 or Ch3.

Table II shows a recapitulation of the classification accuracy of each CNN channel and the fusion operations on the three datasets. In most cases, the Prod operation performs better than the Max operation because it multiplies the scores together,

Authorized licensed use limited to: ShanghaiTech University. Downloaded on August 15,2026 at 17:52:04 UTC from IEEE Xplore. Restrictions apply.

TABLE II
TESTING RESULTS OF THE THREE CNN OUTPUTS AND THE FUSION OPERATIONS ON THE THREE DATASETS

Channels	MSRAction3D (Cross-subject)	UTD-MHAD (Cross-subject)	MAD (Cross-validation : 5-fold)					
			fold-1	fold-2	fold-3	fold-4	fold-5	Average
Sfm1	82.42%	50,00%	70.36%	65.71%	70,00%	68.21%	64.29%	67.71%
Sfm2	87.91%	82.79%	86.10%	86.79%	87.50%	86.10%	91.79%	87.66%
Sfm3	84.99%	82.09%	86.79%	85,00%	82.50%	87.14%	83.93%	85.07%
Fus1	92.31%	85.17%	90.71%	87.14%	85.71%	88.57%	90.36%	88.50%
Fus2	87.91%	85.12%	83.21%	85.71%	87.14%	88.50%	91.07%	87.13%
Fus3	91.21%	85.34%	90.36%	90,00%	91.07%	88.93%	95,00%	91.07%
Fus4	93.41%	88.14%	89.64%	88.57%	92.14%	89.64%	95.35%	91.07%
Fus5	94.51%	87.67%	91.10%	90,00%	92.14%	90.71%	95.36%	91.86%
Max	**94.51%**	**88.14%**	91.10%	90,00%	92.14%	90.71%	95.36%	**91.86%**

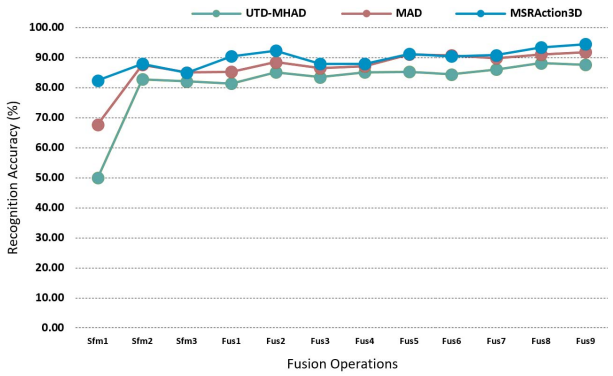


Fig. 9. Performance comparison of the fusion operations on the three datasets.

TABLE III
COMPARISON OF OUR PROPOSED METHOD WITH EXISTING DEPTH-BASED METHODS ON MSRACTION3D DATASET

Method	Accuracy
HON4D [23]	88.89%
SNV [57]	93.45%
Range-Sample Feature [56]	**95.62**%
Random Occupacy Pattern [39]	86.50%
Bag-of-3D-Points [25]	74.70%
STOP [38]	84.80%
DSTIP [58]	89.30%
Proposed (Action-Fusion)	**94.51%**

which not only maximizes the score of the correct class, but it also maximizes the score of the class which is correct but its score has a lower value than the maximum. Additionally, the Prod operation between three channels (Fus4) generates better results than the Prod between two channels (Fus1 and Fus2) because of the features diversity. Although the Max operation does not generate the best results, the correct action often corresponds to the maximum of the channels scores. The fusion operation Fus5 generates the best results in most cases because it is calculated based on both operations Max and Prod, and it involves all the three channels scores in the calculation. The comparison with existing methods is based on taking the maximum accuracy of the fusion operations and the three channels outputs. Fig. 9 shows the stability of the recognition behavior of the fusion operations. If a fusion operation accuracy is higher on one dataset, it is also higher on the two other datasets as well.

A. MSRAction3D

MSRAction3D dataset is captured by Microsoft Kinect v1 depth camera, the dataset contains 20 actions, "high arm wave, horizontal arm wave, hammer, hand catch, forward punch, high throw, draw x, draw tick, draw circle, hand clap, two hand wave, side-boxing, bend, forward kick, side kick, jogging, tennis swing, tennis serve, golf swing, pick up, and throw" performed by ten subjects, each subject repeated the action two or three times. In order to have a fair comparison, the testing settings used by [24] are followed to evaluate our method on the MSRAction3D dataset. Precisely, the cross-subject protocol, odd subjects are used for training (1, 3, 5, 7, and 9) and even subjects (2, 4, 6, 8, and 10) are used for testing. Table II (Row 2: MSRAction3D) shows the classification accuracy results of each CNN channel and the fusion operations. The fusion score Fus5 achieved the best classification accuracy on this dataset, followed by Fus4. The classification results of the second channel Ch2 are better than the ones of Ch1 or Ch3. However, the fusion operations results are better than the three CNN channels results. The maximum value of the results obtained from the fusion operations and the three CNN channels is Fus5 by 94,51%, which we consider for the comparison with existing methods.

Table III shows the comparison results with existing state-of-the-art methods that are based on using depth map data only. The accuracy of our proposed method is better than most existing depth-based approaches except [56]. In spite of the fact that the experiments setting of [56] on MSRAction3D dataset are not mentioned, we also compared our results with their results. Table IV shows the comparison results with existing state-of-the-art methods that are based on using posture data only. The proposed method accuracy is also better than existing skeleton-based methods except [42] which is based on sparse coding and temporal pyramid matching. Generally,

Authorized licensed use limited to: ShanghaiTech University. Downloaded on August 15,2026 at 17:52:04 UTC from IEEE Xplore. Restrictions apply.

TABLE IV
COMPARISON OF OUR PROPOSED METHOD WITH EXISTING SKELETON-BASED METHODS ON MSRACTION3D DATASET

Method	Accuracy
EigenJoints [41]	81.40%
Actionlet Ensemble [24]	88.20%
DL-GSGC [42]	**96.70%**
HOJ3D [26]	78.97%
SSS Feature [27]	81.70%
MP Descriptor [43]	91.70%
High-level Skeleton Feature [44]	82.00%
Pose Set [45]	90.00%
Proposed (Action-Fusion)	**94.51%**

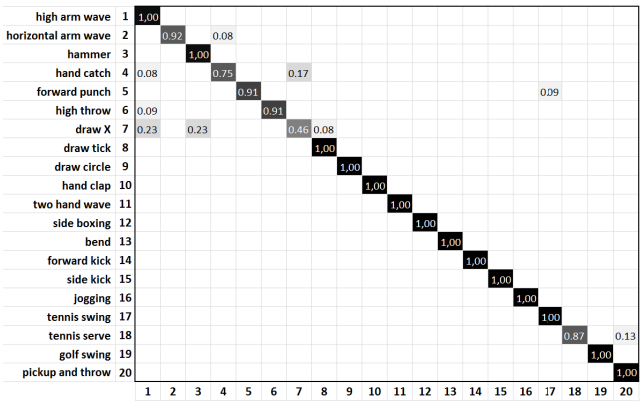


Fig. 10. Confusion matrix of our approach for the MSRAction3D dataset.

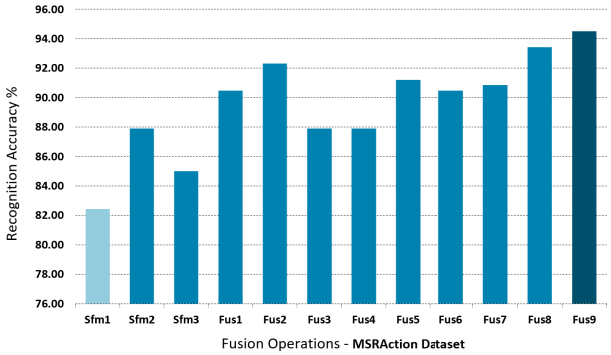


Fig. 11. Fusion operations accuracies of the MSRAction3D dataset.

the performance of our method over skeleton-based and depth based methods is due to the incorporation of depth features and posture features. Fig. 10 shows the confusion matrix of the proposed method for the MSRAction3D dataset and Fig. 11 shows the difference between the fusion operations accuracies for the MSRAction3D dataset. In Fig. 12, we present the DMI and MJD of three actions: 1) high arm wave; 2) horizontal arm wave; and 3) hammer associated with the classification accuracy shown in the confusion matrix (Fig. 10). In spite of the fact that the DMI appearances are almost equivalent, the MJD has different features which helps to recognize the actions even when they are performed in almost similar ways.

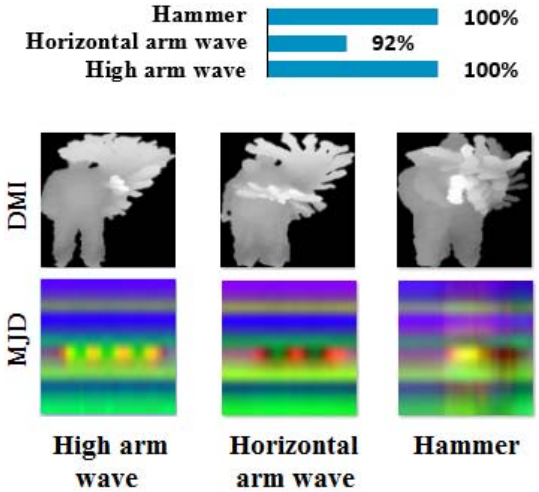


Fig. 12. Classification accuracy of three different actions presented in the confusion matrix of MSRAction3D dataset (Fig. 10), which look semantically equivalent in appearance.

TABLE V
COMPARISON OF OUR PROPOSED METHOD WITH EXISTING METHODS ON UTD-MHAD DATASET

Method	Accuracy
Kinect and Inertial [54]	79.10%
SOS [33]	86.97%
Joint Trajectory Maps [40]	87.90%
Proposed (Action-Fusion)	**88.14%**

B. UTD-MHAD

UTD-MHAD was captured using a fusion of depth and inertial sensor data, it consists of 27 actions performed by 8 subjects. Each subject repeats the action four times. The actions are "right arm swipe to the left, right arm swipe to the right, right hand wave, two hand front clap, right arm throw, cross arms in the chest, basketball shoot, right hand draw x, right hand draw circle (clockwise), right hand draw circle (counter clockwise), draw triangle, bowling (right hand), front boxing, baseball swing from right, tennis right hand forehand swing, arm curl (two arms), tennis serve, two hand push, right hand knock on door, right hand catch an object, right hand pick up and throw, jogging in place, walking in place, sit to stand, stand to sit, forward lunge (left foot forward) and squat (two arms stretch out)." The evaluation settings used for this dataset follow the cross-subject protocol, odd subjects for training and even subjects for testing, same as settings of [54].

Table II (Row 3: UTD-MHAD) shows the classification results of the three CNN channels and the fusion operations. In this dataset, the Fus4 achieved the highest classification accuracy by 88.14%. Similar to the MSRAction3D dataset, the classification results of the second channel Ch2 are better than the classification results of Ch1 or Ch3. As the maximum accuracy value is generated from the fusion operation Fus4, it is considered for the comparison with existing methods that have been tested on the UTD-MHAD dataset. Table V shows the comparison results with the state-of-the-art methods and Fig. 13 shows the confusion matrix evaluation of each

Authorized licensed use limited to: ShanghaiTech University. Downloaded on August 15,2026 at 17:52:04 UTC from IEEE Xplore. Restrictions apply.

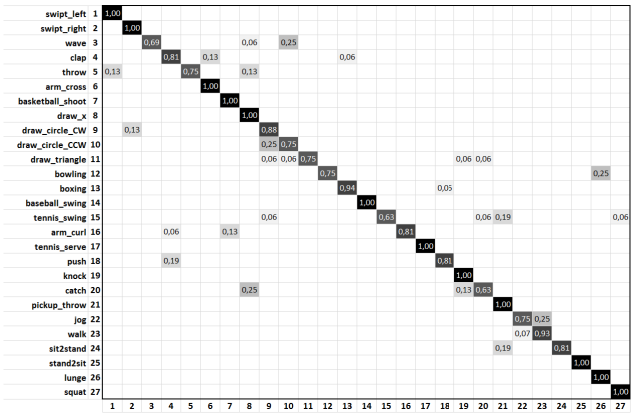


Fig. 13. Confusion matrix of our method for the UTD-MHAD dataset.

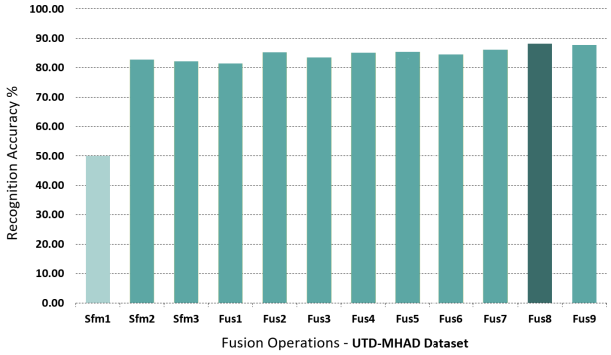


Fig. 14. Fusion operations accuracies of the UTD-MHAD dataset.

action. Although there is no many works have been tested on this dataset like MSRAction3D, the proposed method achieved better results than one of the recent methods [40]. Fig. 14 shows the difference between the fusion operations accuracies on the UTD-MHAD dataset and Fig. 15 presents three different actions which look semantically equivalent, clap, arms cross, and boxing. As it is shown in the confusion matrix (Fig. 13), the clap action is 13% recognized as arm cross and 6% as boxing due to its similar appearance to the two other actions. However the recognition accuracy still 81%, it proves the performance of the proposed method to classify actions even when there is a very small difference in their motions. The arms cross action is fully recognized because it is relatively different from clap and boxing actions.

C. MAD

The MAD dataset is one of largest RGB-D action recognition datasets in term of actions number. It contains 35 actions performed by 20 subjects, each subject performs the action twice. The actions are "running, crouching, jumping, walking, jump and side-kick, left arm swipe to the left, left arm swipe to the right, left arm wave, left arm punch, left arm dribble, left arm pointing to the ceiling, left arm throw, swing from left (baseball swing), left arm receive, left arm back receive, left leg kick to the front, left leg kick to the left, right arm swipe to the left, right arm swipe to the right, right arm wave, right arm punch, right arm dribble, right arm, pointing to the

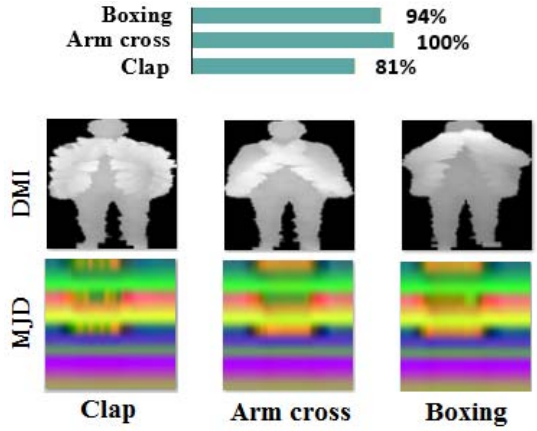


Fig. 15. Classification accuracy of three different actions presented in the confusion matrix of UTD-MHAD dataset (Fig. 13), which look semantically equivalent in appearance.

TABLE VI
COMPARISON OF OUR PROPOSED METHOD WITH EXISTING METHODS ON MAD DATASET

Method	Accuracy
Event Transition [59]	85.02%
Proposed (Action-Fusion)	**91.86%**

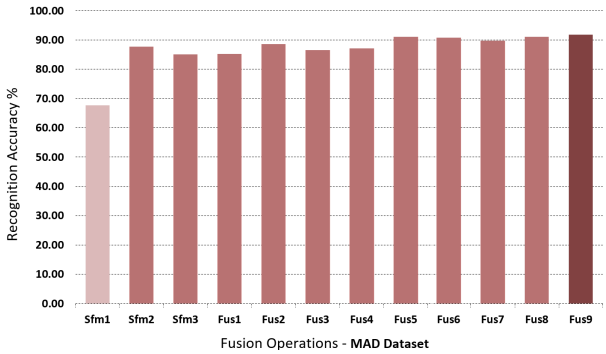


Fig. 16. Fusion operations accuracies of the MAD dataset.

ceiling, right arm throw, swing from right (baseball swing), right arm receive, right arm back receive, right leg kick to the front, right leg kick to the right, cross arms in the chest, basketball shooting, both arms pointing to the screen, both arms pointing to both sides, both arms pointing to right side, both arms pointing to left side."

Unlike the two previous datasets, MAD dataset requires background removing to construct the DMI descriptor. Since the subjects are standing far from the background, we removed the background based on a depth threshold. The evaluation protocol used for this dataset is fivefold cross-validation, which is the same as protocol described in [59]. It is based on using 4/5 of subjects for training and 1/5 for testing, then another new 4/5 of subjects are chosen for training (including 1/5 that previously used for testing) and the rest 1/5 is used for testing. This process should be performed five times to involve all the data in the training and the testing process. The final recognition accuracy is the average of the fivefold results.

Authorized licensed use limited to: ShanghaiTech University. Downloaded on August 15,2026 at 17:52:04 UTC from IEEE Xplore. Restrictions apply.

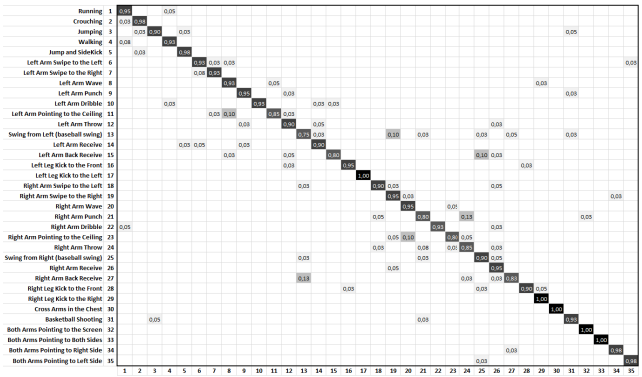


Fig. 17. Confusion matrix of our proposed method for the MAD dataset.

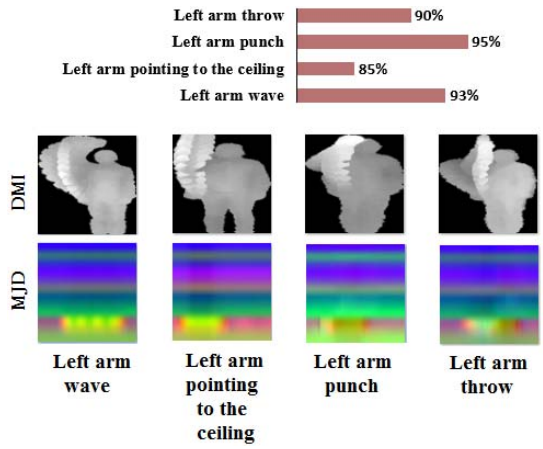


Fig. 18. Classification accuracy of four different actions presented in the confusion matrix of MAD dataset (Fig. 17), which look semantically equivalent in appearance.

Table II (rows 4–9: MAD) shows detailed classification results of the three CNN channels outputs and the fusion operations for each fold and for the average of the fivefold. The maximum accuracy value of the results is generated from the fusion operation Fus5 by 91.86%. To our knowledge, there is only one work tested on the MAD dataset [59]. Table VI shows the comparison results with [59], and Fig. 16 shows the difference between the fusion operations accuracies on the MAD dataset. The confusion matrix of the proposed method on the dataset is shown in Fig. 17. Four different actions are shown in Fig. 18, which look almost semantically equivalent, left arm wave, left arm pointing to the ceiling, left arm punch, and left arm throw. While the four actions performed with a left hand to the top, the DMI descriptors look relatively similar. However, the MJD carries different features. The classification results of the four actions vary from 85% to 95% as presented in the confusion matrix (Fig. 17), which reflects the efficiency of combining depth and posture data for action recognition.

D. Computation Complexity

1) Preprocessing Time: The preprocessing time includes the computation of DMI and MJD descriptors. The calculation of DMI descriptor requires assembling a sequence of raw depth maps of size equals 320 × 240 in one 112 × 112 image. However, the MJD calculation requires transformation of a sequence of 12 × 3 (2-D array) of joints coordinates into an RGB image of size equals 112 × 112. The difference in the input data size and the complexity of preprocessing steps influence widely on computation time, as clearly shown in Fig. 19. For example, an action of 65 frames needs 0.835 s to calculate the DMI descriptor and 0.029 s to calculate the MJD descriptor. More frames involved in the action means more computation time required. The computation time of DMI and MJD with 73 frames are 0.985 and 0.032, respectively. However, with 80 frames, the duration is 1.084 and 0.041, respectively. It is also noticed that the changing rate of the DMI descriptor is larger than the MJD descriptor. If an action includes more than 15 frames (from 65 to 80 frames), the computation increases with 0.249 for DMI and 0.012 for MJD. The results of Fig. 19 have been calculated on CPU with a machine of Intel Core i7-6700 @ 3.40 GHz, 8 GB of RAM and 64 bits operating system. As an example of the preprocessing comparison, we discuss the computation of descriptors in [29]. They calculated a depth descriptor similar to ours. However, they involved an extra parameter v which indicates the action view. They use the descriptor from the front and the side view, and that requires twice computation time than ours. On the other hand, in their method, they use the descriptor to compute two other descriptors from two views as well, which is really high computationally demanding. In our method, we use the descriptor from the front view only with MJD descriptor in which the whole computation of both of them requires less time than [29].

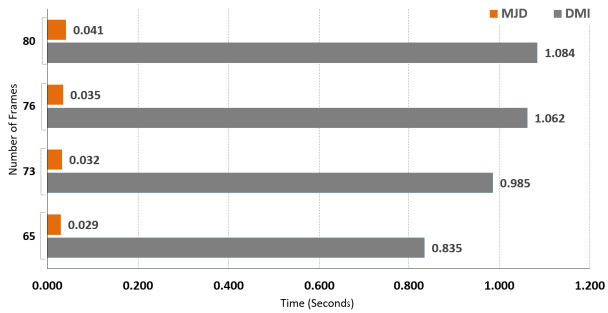


Fig. 19. Computation time of the DMI and MJD descriptors according to the action frames number.

2) Training and Testing Time: The training time differs from a dataset to another, depending on the number of descriptors that are used for training. While MSRAction3D dataset has the lowest number of training data, the training time is also smaller compared to the other two datasets that have more training data. From Table VII, we notice that the training time and number of iterations required for the model to converge are subject to number of training data. The case of the MAD dataset is a little different from the other two datasets. As the evaluation protocol of this dataset demands five training steps to calculate the average of the fivefold results, the computation training time for this dataset is the sum of the five training durations. Even though the fivefold have the same number of training data, the number of iterations required to get the minimum loss differs from onefold to another because each data

Authorized licensed use limited to: ShanghaiTech University. Downloaded on August 15,2026 at 17:52:04 UTC from IEEE Xplore. Restrictions apply.

TABLE VII
TRAINING AND TESTING TIME OF THE THREE DATASETS

Datasets	Number of Training Data	Number of Testing Data	Number of Iterations	Train (min)	Test (sec): One input
MSRAction3D	284	273	441	7.35	0.07
UTD-MHAD	431	430	720	12	0.07
- fold-1			2260	37.67	
- fold-2			1750	29.17	
MAD - fold-3	1120	280	1950	32.5	0.35
- fold-4			4370	72.83	
- fold-5			1470	24.5	

fold has different combination of actions, and hence different types of features to be learned.

While the structure of the model used for training is the same for the three datasets as well as the type of training data, the processing time of action prediction of an input pair for any dataset is the same (0.07 s), but for the MAD dataset, the testing for one input requires averaging the prediction accuracies from the five trained models of the fivefold, which results in 0.07×5 s computation time. The hardware material used for testing and training is different from the one used for the preprocessing. The training process has been performed on a GPU of 12.2 GB memory of a server with Intel Xeon CPU E5-2630 v4 @ 2.20 GHz (10 cores), 64 GB of RAM and 64 bits operating system. Generally, CNN models require a large number of training examples, such as thousands of images with hours of training to reach a high prediction accuracy. Existing RGB-D action datasets like the ones used in this paper have a limited number of training data. The key success of the learning process from a large amount of data is to extract enough features to recognize each action. In our proposed work, two types of descriptors and three CNN channels provide a variety of feature extraction ways that can replace the lack of training data, and with less computation complexity as presented in Table VII.

3) Discussions: Although the recognition accuracy of the proposed work is better than most of existing state-of-the-art results, the computation time from the raw input data to the final action prediction depends on the hardware used for computation. If we want to compare the computation time of the proposed work with the existing works, we should take two aspects into consideration, the descriptor computation time and the classification algorithm complexity. Some existing methods, such as [23] and [60] use only one type of input data, either depth maps or posture data to create a descriptor. However, other methods, such as [40] use three descriptors to cover the human action from different views. In our case two input descriptors are computed, and one of them requires very less computation than the other. According to this analysis, we can classify the proposed method in the middle rank among existing methods in term of descriptors calculation.

Most of the approaches mentioned in the related work section employ SVM as a classifier, such as [23]. Generally, SVM computation time is less than neural networks, but it also depends on how the neural networks model is deep. The approaches which are based on using CNN like ours, require more computation than using simple feed-forward neural networks due to the 2-D processing. Even the CNN approaches differ by the number of layers for the processing. Additionally, one CNN channel is less computationally demanding than three channels. In this case, we can rank the processing time of the proposed method in term of classification among computationally demanding methods. As previously highlighted in the introduction and in the score fusion section, the proposed approach offers many possibilities on how to use the data and the model. For example, using MJD descriptor only with channel Ch3 is not the best choice to produce accurate classification results, but it is still better than some of the existing approaches and with less computation time. More descriptors and channels involved in the action recognition process means high accuracy with low computation speed.

V. CONCLUSION

A method for human action recognition from depth maps and posture data using deep CNNs has been proposed. Two action representations and three CNNs channels are used to maximize feature extraction. The posture data descriptor influence greatly on the whole recognition process by providing features to support the front view of the depth maps representation. Fusion operations between the output predictions of CNN channels are proposed to maximize the score of the right action. Since RGB-D datasets have a small number of training samples, two action representations are helpful to learn the model with a variety of feature and replace the lack of data. The results of our proposed method outperform most of the state-of-the-art methods on three public datasets. Although the proposed method showed high accuracy in action recognition and surpasses most of existing state-of-the-art methods, it was only evaluated using testing samples of humans performing actions in an environment with still cameras from a predefined distance. Future work concerns on collecting a dataset of depth maps and posture data of actions with a moving wearable camera or a robot to record actions performed spontaneously by humans from different views and distances [61], then we train the CNN model on the dataset samples and test the effectiveness of the proposed approach in real environment.

Authorized licensed use limited to: ShanghaiTech University. Downloaded on August 15,2026 at 17:52:04 UTC from IEEE Xplore. Restrictions apply.

REFERENCES

[1] C. Chen, R. Jafari, and N. Kehtarnavaz, "Action recognition from depth sequences using depth motion maps-based local binary patterns," in *Proc. IEEE Win. Conf. Appl. Comput. Vis.*, 2015, pp. 1092–1099.

[2] J. Yu and J. Sun, "Multiactivity 3-D human pose tracking in incorporated motion model with transition bridges," *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published.

[3] W. Chi, J. Wang, and M. Q.-H. Meng, "A gait recognition method for human following in service robots," *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published.

[4] G. Liang, X. Lan, J. Wang, J. Wang, and N. Zheng, "A limb-based graphical model for human pose estimation," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 7, pp. 1080–1092, Jul. 2018.

[5] Y. Guo, D. Tao, W. Liu, and J. Cheng, "Multiview Cauchy estimator feature embedding for depth and inertial sensor-based human action recognition," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 4, pp. 617–627, Apr. 2017.

[6] S. Zhang, C. Gao, F. Chen, S. Luo, and N. Sang, "Group sparse-based mid-level representation for action recognition," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 4, pp. 660–672, Apr. 2017.

[7] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Action recognition by dense trajectories," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 3169–3176.

[8] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 3551–3558.

[9] X. Peng, C. Zou, Y. Qiao, and Q. Peng, "Action recognition with stacked Fisher vectors," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 581–595.

[10] X. Peng, L. Wang, X. Wang, and Y. Qiao, "Bag of visual words and fusion methods for action recognition," *Comput. Vis. Image Understand.*, vol. 150, pp. 109–125, Sep. 2016.

[11] H. Xu, J. Liu, H. Hu, and Y. Zhang, "Wearable sensor-based human activity recognition method with multi-features extracted from Hilbert–Huang transform," *Sensors*, vol. 16, no. 12, pp. 1–26, 2016.

[12] H. Ponce, M. D. L. Martínez-Villaseñor, and L. Miralles-Pechuán, "A novel wearable sensor-based human activity recognition approach using artificial hydrocarbon networks," *Sensors*, vol. 16, no. 7, pp. 1–28, 2016.

[13] J. Qi, P. Yang, M. Hanneghan, S. Tang, and B. Zhou, "A hybrid hierarchical framework for gym physical activity recognition and measurement using wearable sensors," *IEEE Internet Things J.*, to be published.

[14] J. Qi, P. Yang, M. Hanneghan, and S. Tang, "Multiple density maps information fusion for effectively assessing intensity pattern of lifelogging physical activity," *Neurocomputing*, vol. 220, pp. 199–209, Jan. 2017.

[15] P. Yang *et al.*, "Lifelogging data validation model for Internet of Things enabled personalized healthcare," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 1, pp. 50–64, Jan. 2018.

[16] J. Sriwan and W. Suntiamorntut, "Human activity monitoring system based on WSNs," in *Proc. Int. Joint Conf. Comput. Sci. Softw. Eng.*, 2015, pp. 247–250.

[17] G. Chetty and M. White, "Body sensor networks for human activity recognition," in *Proc. Int. Conf. Signal Process. Integr. Netw.*, 2016, pp. 660–665.

[18] M. H. Kolekar and D. P. Dash, "Hidden Markov model based human activity recognition using shape and optical flow based features," in *Proc. IEEE Region 10 Conf.*, 2016, pp. 393–397.

[19] S. Al-Ali, M. Milanova, A. Manolova, and V. Fox, "Human action recognition using combined contour-based and Silhouette-based features and employing KNN or SVM classifier," *Int. J. Comput.*, vol. 9, p. 37, 2015.

[20] A. K. S. Kushwaha, O. Prakash, A. Khare, and M. H. Kolekar, "Rule based human activity recognition for surveillance system," in *Proc. Int. Conf. Intell. Human Comput. Interact.*, 2012, pp. 1–6.

[21] V. K. Singh and R. Nevatia, "Human action recognition using a dynamic Bayesian action network with 2D part models," in *Proc. Indian Conf. Comput. Vis. Graph. Image Process.*, 2010, pp. 17–24.

[22] S. Zhu and D. Song, "Human action recognition based on multiple instance learning," *J. Appl. Sci.*, vol. 14, no. 19, pp. 2276–2284, 2014.

[23] O. Oreifej and Z. Liu, "HON4D: Histogram of oriented 4D normals for activity recognition from depth sequences," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 716–723.

[24] J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Mining actionlet ensemble for action recognition with depth cameras," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 1290–1297.

[25] W. Li, Z. Zhang, and Z. Liu, "Action recognition based on a bag of 3D points," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2010, pp. 9–14.

[26] L. Xia, C.-C. Chen, and J. K. Aggarwal, "View invariant human action recognition using histograms of 3D joints," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2012, pp. 20–27.

[27] X. Zhao, X. Li, C. Pang, X. Zhu, and Q. Z. Sheng, "Online human gesture recognition from motion data streams," in *Proc. ACM Multimedia*, 2013, pp. 23–32.

[28] J. Han, L. Shao, D. Xu, and J. Shotton, "Enhanced computer vision with microsoft kinect sensor: A review," *IEEE Trans. Cybern.*, vol. 43, no. 5, pp. 1318–1334, Oct. 2013.

[29] D. Kim, W.-H. Yun, H.-S. Yoon, and J. Kim, "Action recognition with depth maps using HOG descriptors of multi-view motion appearance and history," in *Proc. Int. Conf. Mobile Ubiquitous Comput. Syst. Services Technol.*, 2014, pp. 126–130.

[30] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[31] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[32] Y. Du, Y. Fu, and L. Wang, "Skeleton based action recognition with convolutional neural network," in *Proc. IAPR Asian Conf. Pattern Recognit.*, 2015, pp. 579–583.

[33] Y. Hou, Z. Li, P. Wang, and W. Li, "Skeleton optical spectra-based action recognition using convolutional neural networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 3, pp. 807–811, Mar. 2018.

[34] M. S. Ryoo and L. Matthies, "First-person activity recognition: What are they doing to me?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 2730–2737.

[35] M. S. Ryoo, T. J. Fuchs, L. Xia, J. K. Aggarwal, and L. Matthies, "Robot-centric activity prediction from first-person videos: What will they do to me?" in *Proc. ACM/IEEE Int. Conf. Human–Robot Interact.*, 2015, pp. 295–302.

[36] I. Gori, J. Sinapov, P. Khante, P. Stone, and J. K. Aggarwal, "Robot-centric activity recognition 'in the wild,'" in *Social Robotics*. Cham, Switzerland: Springer, 2015, pp. 224–234.

[37] P. Duckworth, M. Alomari, Y. Gatsoulis, D. C. Hogg, and A. G. Cohn, "Unsupervised activity recognition using latent semantic analysis on a mobile robot," in *Proc. Eur. Conf. Artif. Intell.*, 2016, pp. 1062–1070.

[38] A. W. Vieira, E. R. Nascimento, G. L. Oliveira, Z. Liu, and M. F. M. Campos, "STOP: Space-time occupancy patterns for 3D action recognition from depth map sequences," in *Proc. Progr. Pattern Recognit. Image Anal. Comput. Vis. Appl.*, 2012, pp. 252–259.

[39] J. Wang, Z. Liu, J. Chorowski, Z. Chen, and Y. Wu, "Robust 3D action recognition with random occupancy patterns," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 872–885.

[40] P. Wang, W. Li, C. Li, and Y. Hou, "Action recognition based on joint trajectory maps with convolutional neural networks," *CoRR*, vol. abs/1612.09401, pp. 1–11, Dec. 2016.

[41] X. Yang and Y. L. Tian, "EigenJoints-based action recognition using Naïve–Bayes-Nearest-Neighbor," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2012, pp. 14–19.

[42] J. Luo, W. Wang, and H. Qi, "Group sparsity and geometry constrained dictionary learning for action recognition from depth maps," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 1809–1816.

[43] M. Zanfir, M. Leordeanu, and C. Sminchisescu, "The moving pose: An efficient 3D kinematics descriptor for low-latency action recognition and detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 2752–2759.

[44] D. Wu and L. Shao, "Leveraging hierarchical parametric networks for skeletal joints based action segmentation and recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 724–731.

[45] C. Wang, Y. Wang, and A. L. Yuille, "An approach to pose-based action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 915–922.

[46] Q. Ke, M. Bennamoun, S. An, F. Sohel, and F. Boussaid, "A new representation of skeleton sequences for 3D action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4570–4579.

[47] J. Koushik, "Understanding convolutional neural networks," *CoRR*, vol. abs/1605.09081, pp. 1–6, May 2016.

[48] J. Gu *et al.*, "Recent advances in convolutional neural networks," *Pattern Recognit.*, vol. 77, pp. 354–377, May 2018.

[49] E. Park, X. Han, T. L. Berg, and A. C. Berg, "Combining multiple sources of knowledge in deep CNNs for action recognition," in *Proc. IEEE Win. Conf. Appl. Comput. Vis.*, 2016, pp. 1–8.

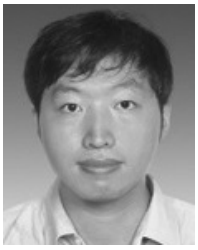
[50] M. Lin, Q. Chen, and S. Yan, "Network in network," *CoRR*, vol. abs/1312.4400, pp. 1–10, Dec. 2013.

Authorized licensed use limited to: ShanghaiTech University. Downloaded on August 15,2026 at 17:52:04 UTC from IEEE Xplore. Restrictions apply.

[51] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. Int. Conf. Artif. Intell. Stat.*, vol. 9, 2010, pp. 249–256.
[52] Y. Jia *et al.*, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. ACM Multimedia*, 2014, pp. 675–678.
[53] J. Zhang, W. Li, P. O. Ogunbona, P. Wang, and C. Tang, "RGB-D-based action recognition datasets: A survey," *Pattern Recognit.*, vol. 60, pp. 86–105, 2016.
[54] C. Chen, R. Jafari, and N. Kehtarnavaz, "UTD-MHAD: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor," in *Proc. IEEE Int. Conf. Image Process.*, 2015, pp. 168–172.
[55] D. Huang, S. Yao, Y. Wang, and F. De La Torre, "Sequential max-margin event detectors," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 410–424.
[56] C. Lu, J. Jia, and C.-K. Tang, "Range-sample depth feature for action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 772–779.
[57] X. Yang and Y. Tian, "Super normal vector for human activity recognition with depth cameras," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 5, pp. 1028–1039, May 2017.
[58] L. Xia and J. K. Aggarwal, "Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 2834–2841.
[59] Y. Kim *et al.*, "Modeling transition patterns between events for temporal human action segmentation and classification," in *Proc. IEEE Int. Conf. Workshops Autom. Face Gesture Recognit.*, 2015, pp. 1–8.
[60] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy, "Sequence of the most informative joints (SMIJ): A new representation for human skeletal action recognition," *J. Vis. Commun. Image Represent.*, vol. 25, no. 1, pp. 24–38, 2014.
[61] Y. Nie, C. Xiao, H. Sun, and P. Li, "Compact video synopsis via global spatiotemporal optimization," *IEEE Trans. Vis. Comput. Graphics*, vol. 19, no. 10, pp. 1664–1676, Oct. 2013.

Aouaidjia Kamel received the M.Eng. degree in computer science from the Abbes Laghrour University of Khenchela, Khenchela, Algeria, in 2009. He is currently pursuing the Ph.D. degree in computer science with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China.

His current research interests include undestanding human behavior, human–machine interaction, machine learning, and deep neural networks.



Bin Sheng received the B.A. degree in English and the B.Eng. degree in computer science from the Huazhong University of Science and Technology, Wuhan, China, the M.Sc. degree in software engineering from the University of Macau, Macau, China, and the Ph.D. degree in computer science from the Chinese University of Hong Kong, Hong Kong.

He is currently an Associate Professor with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China. His current research interests include virtual reality and computer graphics.

Po Yang received the B.Sc. degree in computer science from Wuhan University, Wuhan, China, in 2004, the M.Sc. degree in computer science from the University of Bristol, Bristol, U.K., in 2006, and the Ph.D. degree in electronic engineering from the University of Staffordshire, Stoke-on-Trent, U.K., in 2010.

He is currently a Senior Lecturer with the Department of Computing Science, Liverpool John Moores University, Liverpool, U.K. He holds a strong tracking of high-quality publications and research experiences. He has published over 40 papers. His current research interests include Internet of Things, RFID and indoor localization, pervasive health, image processing, GPU, and parallel computing.

Ping Li received the Ph.D. degree in computer science and engineering from the Chinese University of Hong Kong, Hong Kong.

He is currently an Assistant Professor with the Macau University of Science and Technology. He has one image/video processing national invention patent, and has excellent research project reported worldwide by *ACM TechNews*. His current research interests include image/video stylization, big data visualization, GPU acceleration, and creative media.

Ruimin Shen received the B.Sc. and M.Sc. degrees in computer science from Tsinghua University, Beijing, China, and the Ph.D. degree in computer science from Hagen University, Hagen, Germany.

He is currently a Full Professor with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China. His current research interests include virtual reality, computer graphics, e-learning technologies, knowledge discovery, and data mining.

David Dagan Feng (F'03) received the Ph.D. degree in computer science from the University of California at Los Angeles, Los Angeles, CA, USA, in 1988.

He is currently the Head with the School of Information Technologies, the Director of Biomedical and Multimedia Information Technology Research Group, and the Research Director with the Institute of Biomedical Engineering and Technology, University of Sydney, Sydney, NSW, Australia. He has published over 700 research papers, pioneered several new research directions, and made a number of landmark contributions in his field.

Dr. Feng was a recipient of the Crump Prize for Excellence in Medical Engineering at the University of California at Los Angeles. He has served as the Chair of the International Federation of Automatic Control Technical Committee on Biological and Medical Systems. He has organized/chaired over 100 major international conferences/symposia/workshops, and has been invited to give over 100 keynote presentations in 23 countries/regions. He is a fellow of the Australian Academy of Technological Sciences and Engineering.

Authorized licensed use limited to: ShanghaiTech University. Downloaded on August 15,2026 at 17:52:04 UTC from IEEE Xplore. Restrictions apply.