This is a repository copy of *Deep convolutional neural networks for human action recognition using depth maps and postures*.

White Rose Research Online URL for this paper:
http://eprints.whiterose.ac.uk/151224/

Version: Accepted Version

Article:
Kamel, A., Sheng, B., Yang, P. orcid.org/0000-0002-8553-7127 et al. (3 more authors) (2019) Deep convolutional neural networks for human action recognition using depth maps and postures. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 49 (9). pp. 1806-1819. ISSN 2168-2216

https://doi.org/10.1109/tsmc.2018.2850149

© 2019 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other users, including reprinting/ republishing this material for advertising or promotional purposes, creating new collective works for resale or redistribution to servers or lists, or reuse of any copyrighted components of this work in other works. Reproduced in accordance with the publisher's self-archiving policy.

Reuse
Items deposited in White Rose Research Online are protected by copyright, with all rights reserved unless indicated otherwise. They may be downloaded and/or printed for private study, or other acts as permitted by national copyright laws. The publisher or other rights holders may allow further reproduction and re-use of the full text version. This is indicated by the licence information on the White Rose Research Online record for the item.

Takedown
If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk/

This is a repository copy of *Deep Convolutional Neural Networks for Human Action Recognition Using Depth Maps and Postures*.

White Rose Research Online URL for this paper:
http://eprints.whiterose.ac.uk/151224/

Version: Accepted Version

Article:
Kamel, A, Sheng, B, Yang, P orcid.org/0000-0002-8553-7127 et al. (3 more authors) (2019) Deep Convolutional Neural Networks for Human Action Recognition Using Depth Maps and Postures. IEEE TRANSACTIONS ON SYSTEMS MAN CYBERNETICS-SYSTEMS, 49 (9). pp. 1806-1819. ISSN 2168-2216

https://doi.org/10.1109/TSMC.2018.2850149

Reuse
Items deposited in White Rose Research Online are protected by copyright, with all rights reserved unless indicated otherwise. They may be downloaded and/or printed for private study, or other acts as permitted by national copyright laws. The publisher or other rights holders may allow further reproduction and re-use of the full text version. This is indicated by the licence information on the White Rose Research Online record for the item.

Takedown
If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk/

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

Aouaidjia Kamel, Bin Sheng, *Member, IEEE,* Yang Po, *Member, IEEE,* Ping Li, and Ruimin Shen

***Abstract*—In this paper, we present a method for human action recognition from depth images and posture data using convolutional neural networks (CNN). Two input descriptors are used for action representation, the first input is a depth motion image (DMI) that accumulates consecutive depth images of a human action, whilst the second input is a proposed moving joints descriptor (MJD) 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, the second channel is trained with both depth motion images and moving joint descriptors together, and the third channel is trained with moving joint descriptors only. The action predictions from the three CNN channels are fused together for the final action classification. The experiments show that the results of fusing the output of three channels are better than using one channel or fusing two channels only. The proposed method was evaluated on three public datasets: MSRAction3D, UTD-MAHD, and MAD dataset. The testing results indicate that the proposed approach outperforms most of existing state of the art methods such as HON4D and Actionlet on MSRAction3D. Although MAD dataset contains a high number of actions (35 actions) compared to existing action RGB-D datasets, the proposed method achieved 91.86% of accuracy.**

***Index Terms*—Action Recognition, Depth Motion Image, Moving Joints Descriptor, Convolutional neural network.**

I. Introduction

HUMAN action recognition is necessary for various computer vision applications that demand information of people's behavior, including surveillance for public safety, human-computer interaction applications and robotics [1]-[3]. However, action recognition in colored images is challenging task due to several factors, such as complex background, illumination variation, and clothing color, which make it difficult to segment the human body in every scene. The lack of depth cues in colored images has a negative impact on recognizing the action. Especially when it is performed in the camera direction. Depth sensors like Microsoft Kinect provide RGB-D images with illumination invariant, uniform color, and depth information that eases the ambiguity of human's motion. Additionally, depth sensors integrated real-time body skeleton estimation, providing relatively accurate posture information on the body joints in 3d coordinates system.

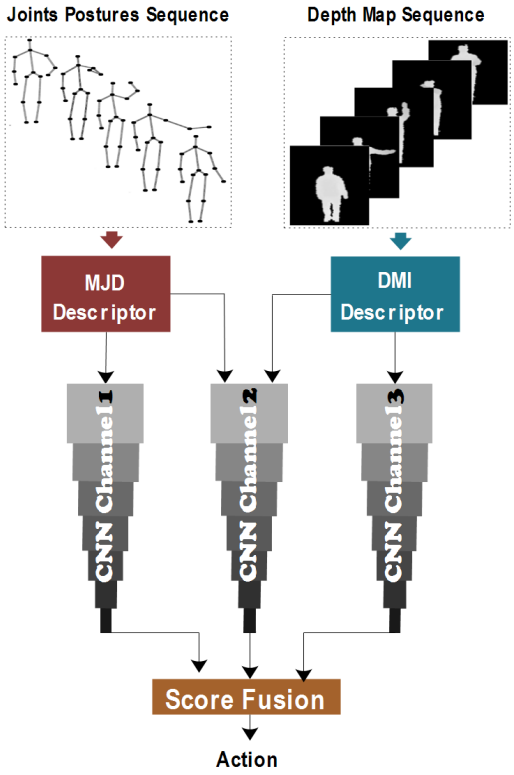

Fig. 1. The framework of the proposed action recognition method.

Recently, action recognition research has been directed toward using depth sensors due to the expressive features provided either from depth maps data or body posture data. The key success for an action recognition method lies on a good representation that provides distinctive features of each action for classification. Using depth map data from front view only for action recognition is still ambiguous for some actions, which leads to the wrong classification, because two actions may look similar from the front view, but they have a different appearance from side views. However, some existing methods such as [4] use feature extraction from different views in order to collect enough features about the action. On the other hand, using posture data for action representation is very sensitive to the movement of joints, which may reflect on recognizing two similar actions as different actions when they

2017 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, including reprinting/republishing this material for advertising or promotional purposes, collecting new collected works for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Aouaidjia Kamel, Bin Sheng, and Ruimin Shen are with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: KAMEL2015@sjtu.edu.cn, shengbin@cs.sjtu.edu.cn, and rmshen@cs.sjtu.edu.cn).

Yang Po is with the Department of Computer Science, Liverpool John Moores University, Byrom Stree, Liverpool, L3 3AF, UK (e-mail: poyangcn@gmail.com).

Ping Li is with the Department of Mathematics and Information Technology, The Education University of Hong Kong, (e-mail: pli@eduhk.hk).

are performed in a slightly different ways. A better approach for action recognition should be based on using the two types of data to overcome the weaknesses of using just one type. Regardless the data used for the action representation, feature extraction, and classification techniques play a major role in the recognition process. Basically, the approaches that are based on handcraft feature extraction such as [5] and [6], employ SVM as a classifier. However, in recent few years, deep learning and especially convolutional neural network (C-NN) which was inspired by the human visual cortex hierarchic processing, have made a huge success in image classification [7]. CNN is a powerful technique for both feature extraction and classification, it can automatically learn discriminative features from a training data.

In this paper, a new method is proposed for human action recognition from depth maps and posture data using three channels of a deep convolutional neural network model,

The contribution of the proposed work can be summarised in :

- In order to strengthen the weaknesses of using one type of data for action recognition, two action representations are used. Depth map representation and body joint representation. The proposed body joints representation is inspired by the way that the human body joints move to cover the joints direction in addition to the changing in joints position Fig. 5.
- A well designed CNN model is trained especially to extract features from the two types of action representation, taking the computation time in consideration by using "Network In Network" structure [26]. Three channels of the model are used to extract features from various input data.
- Fusion operations between prediction results of the three CNN channels are proposed in order to enhance the prediction accuracy. The proposed method offers a flexibility in choosing the way to classify the action by two types of data, three CNN channels, and many fusion operations.
- A large amount of training data is one of the key success of a CNN model prediction accuracy. Due to the lack of a large RGB-D action recognition dataset, using two action representations helps to reinforce the learning process on a small amount of data.

The first representation is a Depth Motion Image descriptor (DMI) similar to [4], but with a little difference in the computation method. It assembles the depth maps of an action in order to capture the changing in depth of human motion. The second representation is a proposed Moving Joints Descriptor (MJD) inspired from [8]. It represents the body joints movement over time using spherical coordinates instead of directly using Cartesian coordinates. The motivation behind choosing spherical coordinates is that the human body joints generally move around a fixed point of the body center in a circular manner. The changing in the angle provide further information about the joint direction, unlike 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

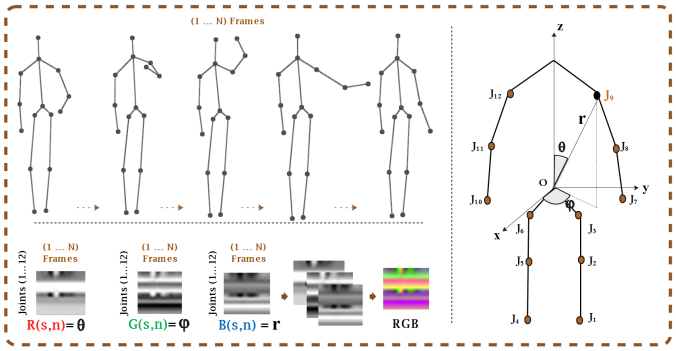

Fig. 2. Moving Joints Descriptor (MJD). Example of draw circle action from the MSRAction3D dataset, left-top: Skeleton sequence, left-bottom: Creation of RGB Moving Joints Descriptor Image, right: Skeleton model shows the three spherical coordinates of joint j_9. Where N: total number of frames, s: joint number, and n: frame number.

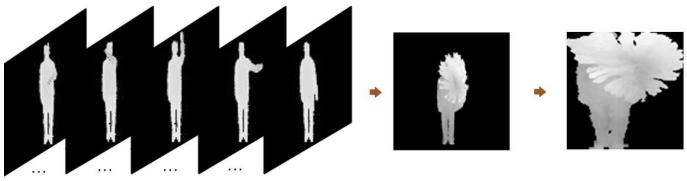

Fig. 3. Depth Motion Image (DMI). Example of draw circle action from MSRAction3D dataset, left: depth map sequence, middle: Depth Motion Image, right: Cropping ROI.

descriptors for feature extraction and classification. The first channel is trained with DMI, the second channel is a connection between two sub-channels, sub-channel is trained with DMI and the other is trained with MJD, while the third channel is trained with MJD only. The proposed approach generates three outputs from the CNN channels and nine other outputs produced from 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 can't be seen in the other channels which make combining them together produce better results. Many fusion operations are proposed in order to analyze and select the best operation that produces high accuracy prediction.

The experimental results of the proposed approach are compared with state of the art methods on three public datasets, MSRAction3D, UTD-MAHD, and MAD dataset. The comparison outcomes proved that the action recognition accuracy is better than most of existing methods and proved also that recognition accuracy is stable even with a large number of actions such as MAD dataset.

The remainder of this paper is organized as follows. A review of the related work is presented in Section II. After that, technical details of the proposed approach are given in Section III followed by the experiments and results in Section IV. A conclusion is reported in section V.

II. Related Work

Several recent depth-based approaches have been reported to improve human action recognition accuracy. An action graph based on a sampled 3D representation from a depth map to model the human motion is proposed in [9]. Several 4D descriptors have been used to represent the human action. In [5] a histogram of oriented 4D normals (HON4D) used in order to describe the action in 4D space covering spatial coordinates, depth and time. [10] also represents the depth sequence in 4D grids by dividing the space and time axis into multiple segments. Another 4D descriptor proposed by [11] called Random Occupancy Pattern (ROP) which deals with noise and occlusion combined with sparse coding approaches to increase robustness. Action recognition from different side views has been applied to gain more discriminative features. [4] generates side view from the front view of the depth map, both views are transformed to DMA (Depth Motion Appearance) descriptor and DMH (Depth Motion History) descriptor. Then, SVM is trained with the to descriptors to classify the action. Recently [12] generate top and side views by rotating 3D points from the front view. The three views are used as inputs to three convolutional neural network models for feature extraction and action classification.

In parallel to depth-based approaches, skeleton-based methods also have a huge contribution to the action recognition research area. In [13], each joint is associated with a Local Binary Pattern descriptor which is translation invariant and provide highly discriminative features. Additionally, a temporal motion representation called Fourier Temporal Pyramid is also proposed in order to model the joints movements. EigenJoints is a new type of features proposed in [14] to combine action information including static postures, motion and offset features. A framework based on sparse coding and temporal pyramid matching is proposed in [15] for better 3D joint features representation. A histogram of 3D joint location called HOJ3D in [16] represents the human joints locations. Then, a posture words are built from HOJ3D vectors and trained using a Hidden Markov Model to classify the actions. In [17] a framework is proposed for online human action recognition using a new Structured Channeling Skeletons feature (SSS) which can deal with intra-class variations including viewpoint, anthropometry, execution rate, and personal style. [18] proposed non-parametric 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 [19] based on using deep belief networks for feature extraction and encoding dynamic structure into a HMM-based model. [20] addresses 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. Moreover, data mining techniques have been applied to get spatial-temporal pose structures for action representation. [8] and [21] transform the joint coordinates to a 2D image descriptor. A convolutional neural network model is used for action classification from the descriptor. Very recent works: SOS [22] and Joint Trajectory Maps [12] propose a new approach which transforms the skeleton joints trajectories shapes from 3D space into three images that represent the front view, the top view and the side view of the joints' trajectory shapes. Three convolutional neural networks extract features from the three images to classify the action.

Convolutional neural network [23] 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. [24] summarise recent advances in convolutional neural network in term of regularisation, optimisation, Activation functions, loss functions, weight initialization and so on. Recent CNN-based action recognition methods are based on using multiple action representations that employ many CNN channels for the processing. In [25], many feature concatenation architectures are proposed in order to improve the classification accuracy using multiple sources of knowledge.

In spite of the fact that the previous approaches achieved good results, the problem of action recognition is still open and require more robust action representations and feature extraction techniques to improve the accuracy and overcome the weakness of the previously mentioned 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 the power of CNN for feature extraction and classification.

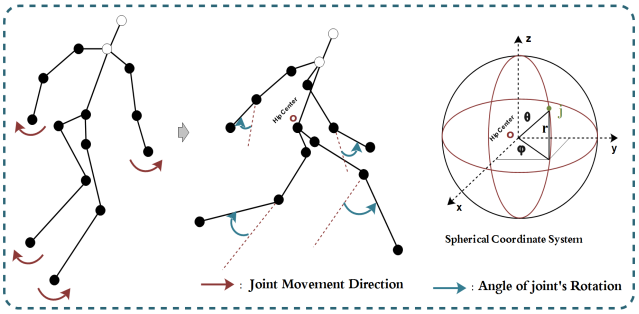

Fig. 4. Human body joints motion direction during a running action. The joints motion more subject to a rotation, which makes the spherical coordinate system more suitable to represent the joints movements.

III. Approach Overview

The framework of the proposed action recognition method is presented in Fig. 1.we use two types of data for human action representation, depth maps, and body postures. Each depth map frame is associated with the body postures. Each of the two inputs is transformed to a descriptor that assembles the input sequence in one image in order to provide an informative description of the action. Namely, DMI for depth maps and MJD for body joints. The DMI descriptor captures the changing in depth of the action during the body motion. The MJD descriptor which inspired from the nature of the human body joints movement around a fixed point to capture

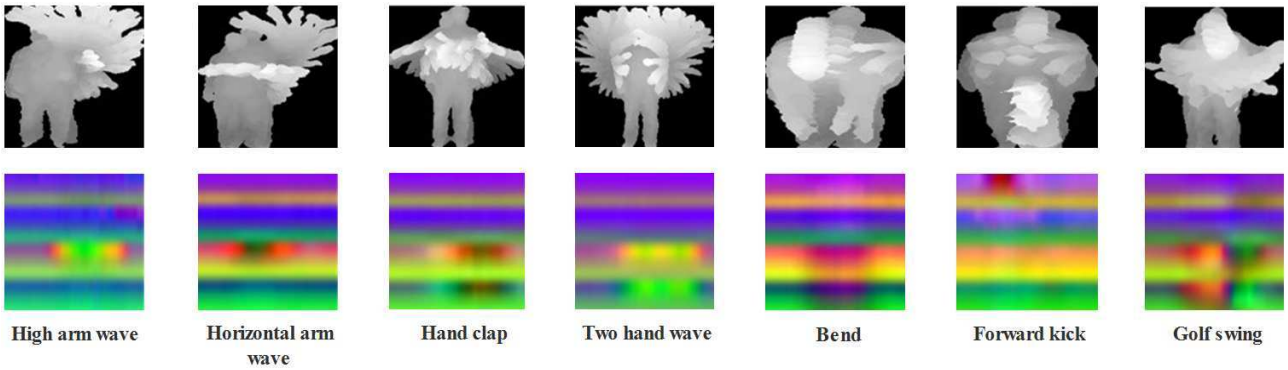

Fig. 5. Preprocessing results of seven actions samples from the MSRAction3D dataset.

the joints direction and the changing in the joint position. The MJD descriptor overcome the lack of side views in the DMI descriptor Three CNN models of the same structure are trained and tested with the two descriptors in a way that one model takes two descriptors as input and each of the two other models takes only one descriptor. The reason behind this assumption is to exploit the power of CNN for extracting features from the two descriptors in different ways with multiple channels for the sake of improving the classification accuracy.

We propose several score fusion operations to get a high score of the accuracy prediction by combining the outputs of the three models. The model training and testing are performed on three action datasets that contain both depth images and posture data.

IV. ACTION RECOGNITION METHOD

A. Data Preprocessing

1) Depth Motion Image (DMI): Depth motion image describes the overall action appearance by accumulating all depth maps of the action over time in order to make 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 provide distinctive features for each action which ease the feature extraction task for the CNN model.

The following equation illustrates the calculation of DMI.

$$\begin{aligned} \textbf{\textit{DMI(i,j)}} &= 255 - min(I(i,j,t)) \\ &\forall t \in [k...(k+N-1)] \end{aligned} \tag{1}$$

Where *I(i,j,t)* is the pixel position *(i,j)* of a frame *I* at time *t*, *DMI* is a grey image (8 bits) that represents the depth difference from frame *k* to *k+N-1*. The pixel value of DMI image is the min value of the same pixels position of the depth maps sequence of the action.

The resulting image is normalized by dividing each pixel value by the max value of image pixels, then the ROI (Region Of Interest) 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 (MJD): From the 20 joints of the skeleton model provided by the datasets, only 13 most informative joints including the hip center are selected. Fig. 2(right) shows the joints selected for the processing. The posture data provided by the datasets are presented in a form of Cartesian coordinates (x, y, z). However, Action representation using Cartesian coordinates is sensitive to joints movement, which may reflect on representing two similar actions as different actions. The movement of human body joints during the motion is subject to some restrictions. They can't move farther than a limited distance from the hip center joint. Furthermore, each body joint has a limited range of angle to move. Those restrictions can be modeled by spherical coordinates as presented in Fig. 4. The distance r represents how is the joint far from the hip center O. The angles θ and ϕ are useful to indicates 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 taking the hip center joint O as the origin of the system. The transformation is described in equations (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 changing in θ, ϕ and r. Three grey images R, G and B are constructed to represent the changing in the angles θ, ϕ and the radius r respectively over time. The rows number of the images represents the joints number, the columns number represents the frames number of the action and the pixel value is the coordinate of the joint s in the frame n as illustrated in equation(4). Finally, an RGB image is constructed by combining the three grey images together to produce the finale descriptor image. The representation proposed tries to extract the most informative features of the body motion by capturing the variation in the angle θ, the angle ϕ and the radius r over the frames sequence. Each of those three grey images provides an action representation, but using only one of them as an action descriptor is not enough, because two different actions may have the same angle θ which results in the wrong classification. However, we can't find two actions that have the same angles θ, ϕ and radius r. The combination of the three grey images provides more distinctive

representation. Fig. 2(left) illustrates the construction of MJD.

$$Joints = \{O, J_1, J_2, ..., J_{12}\}, \quad J_s = (\theta, \phi, r) \tag{2}$$

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

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

Where x, y and z are the Cartesian coordinates. θ, ϕ and r are the spherical coordinates. *R*,*G* and *B* are grey images, and *MJD* is the 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 112x112 and used as input to the CNN model. The model is composed of convolutional layers for feature extraction and pooling layers for dimensionality reduction. 32 convolutional filters of size 7x7 are used in the first convolutional layer, and three 5x5 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 3x3.

Each of the convolutional layers mentioned before is followed by a "Network In Network" structure proposed by [26]. It is based on using convolution filters with 1x1 size and larger numbers than the previous layer, which makes the model deeper and have more parameters without completely changing the network structure, and with cheap computation cost. However in our CNN model, the number of 1x1 convolutional filters is the same as the previous layer. During the training experiments, we found that using 1x1 convolutional without increasing the depth size improve the accuracy without a noticeable influence on the computation time. Fig 7. shows how the two 1x1 convolutional layers are used. The size of the output feature map after using two 1x1 convolution layers is the same as the input.

Three max-pooling layers of filter size 3x3 are used for dimensionality reduction. Each convolutional layer in the model is followed by ReLu (Rectified Linear Units) activation function for increasing non-linearity. A fully connected layer with a size equal to the number of actions is used as the result of feature extraction. Fig. 6 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 images either DMI or MJD make it difficult to capture distinctive features when the convolutional operation is applied with small filter size. For example, the application of 3x3 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 3x3 region, which is the reason behind using 7x7 filters and 5x5 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 the proposed model and according to the training experiments, we found that using only one fully connected layer as a classifier after the pooling layer preserve features and generates better results. At the testing phase, softmax regression layer is used to generate a score for each class based on the trained weights. The class which has the highest score is considered as the correct class.

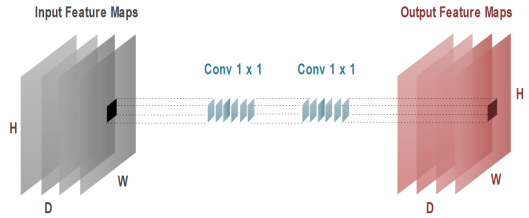

Fig. 7. The network block structure used to improve the CNN model performance accuracy with less computation cost. H, W, and D refer to height, width, and depth of the feature maps.

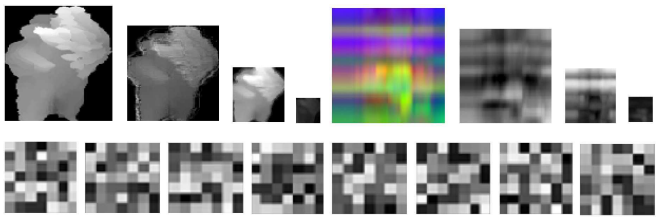

Fig. 8. Top: samples of feature maps from the second channel Ch2 of the CNN model (DMI features from sub-channel Sub1 and MJD features from sub-channel Sub2). Bottom: samples of trained 7x7 filters.

2) Model training: The CNN model described previously was involved in three different training channels. We denote channel 1 as $Ch1$, channel 2 as $Ch2$ and channel 3 as $Ch3$. The channel $Ch1$ was trained with DMI descriptors, the channel $Ch2$ was trained with DMI and MJD descriptors together, and channel $Ch3$ was trained with MJD only. The Channel $Ch2$ is a composition of two others sub-channels: $Sub1$ and $Sub2$. Each of the two sub-channels was trained with one kind of descriptors, namely $Sub1$ was trained with DMI descriptors and $Sub2$ was trained with MJD descriptors. The two sub-channels are concatenated after the last pooling layer, which results in a new layer of depth size equal to the sum of the output of the two pooling layers of sub-channels. The concatenation operation was inspired by [25] which propose different concatenation methods based on fusing the last fully connected layers. However in our case, and according to our experiments, we found that the concatenation between the outputs of pooling layers is more efficient than the concatenation of fully connected layers.

The three channels mentioned before were trained together at the same time with the same parameters. The appropriate learning rate for the network to converge is 0.0008 with a weight decay of 0.0005 and a momentum of 0.9. The batch size selected for the training is 50 images for the three channels with all the datasets. The weight initialization was performed using Xavier method [24]. The number of iterations required for each channel to reach the minimum value of the loss

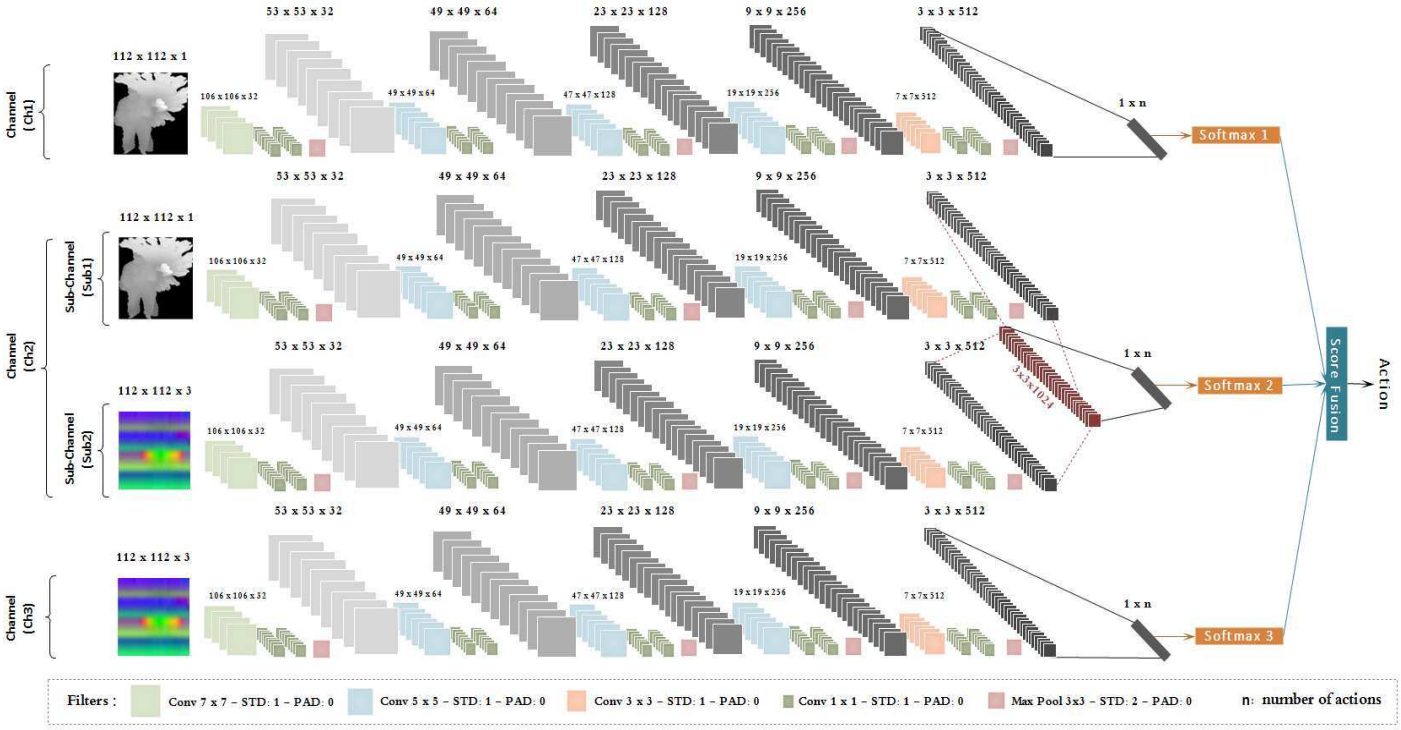

Fig. 6. Proposed three channels convolutional neural network model for action recognition.

function differs from one channel to another depending on the data input type and the dataset size. The network in Fig. 6 is designed, trained and tested using caffe deep learning framework [27].

C. Score Fusion

The output of softmax layer is a vector of length equal to the number of actions (equation(5)), where each element represents the probability of the input image to be a specific action. In most cases, the maximum value corresponds to the correct action. However, for some test samples, the maximum value doesn't represent the correct action. The correct action may correspond to a probability value lower than the maximum value. In order to improve the prediction accuracy of the samples data that generate the wrong classification, 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 other operations. In most cases, the $Prod$ operation performs better than the Max operation as will be discussed in the results section.

The classification accuracy not only depends on the operation Max or $Prod$, but it depends also on the channels involved in the computation. For example, the result of Max operation between softmax output $Sfm1$ of channel $Ch1$ and $Sfm2$ of channel $Ch2$ is different when it is performed between $Sfm2$ of channel $Ch2$ and $Sfm3$ of channel $Ch3$ or between the three channels together Sfm1, Sfm2 and Sfm3. While the accuracy varies according to the operation type and the channel type, different fusion operations are proposed and summarized in Table I. In total we have twelve possible predictions of the proposed methods, three from the CNN channels ($Sfm1$,$Sfm2$,$Sfm3$) and nine from the fused channels ($Fus1$,$Fus2$,...,$Fus9$). The final classification result is the maximum values of the twelve outputs as cited in equation(6).

The motivation behind the model fusion architecture described in Fig. 6, 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 rarely when we find two actions have similar features even when they represent the same action. The channel $Ch2$ provide features that balance between the two representations, which reflects its good results over channels $Ch1$ and $Ch2$ (Results Section). Additionally, the fusion operations try to generate the correct action class throw combining the three channels predictions.

$$\begin{aligned} Sfm1 &= \{p_{11}, ..., p_{1n}\} \\ Sfm2 &= \{p_{21}, ..., p_{2n}\} \\ Sfm3 &= \{p_{31}, ..., p_{3n}\} \end{aligned} \tag{5}$$

where $Sfm\boldsymbol{i}$: is the softmax layer output of channel $Ch\boldsymbol{i}$ and $p_{\boldsymbol{ij}}$: represents the probability of an action $\boldsymbol{j}$ to be the correct class in channel $Ch\boldsymbol{i}$.

Where Max calculates element wise maximum value between the softmax's vectors output from the three CNN channels, Whilst $Prod$ calculates the dot product value between

TABLE I
SCORE FUSION OPERATIONS ON THE THREE CNN CHANNELS.

Fusion	Operation
Fus1	$Max(Sfm1, Sfm3)$
Fus2	$Prod(Sfm1, Sfm3)$
Fus3	$Max(Sfm1, Sfm2)$
Fus4	$Prod(Sfm1, Sfm2)$
Fus5	$Max(Sfm2, Sfm3)$
Fus6	$Prod(Sfm2, Sfm3)$
Fus7	$Max(Sfm1, Sfm2, Sfm3)$
Fus8	$Prod(Sfm1, Sfm2, Sfm3)$
Fus9	$Prod(Prod(Sfm1, Sfm2, Sfm3), Max(Sfm1, Sfm3))$

the softmax's vectors.

$$\boldsymbol{Action} = \boldsymbol{Max}(Sfm1, Sfm2, Sfm3, Fus1, ..., Fus9) \quad (6)$$

Where $Action$ represents the action of the highest score which represent the final class prediction.

V. RESULTS

Referring to [28] which provides a survey of most commonly used RGB-D human action recognition datasets, three datasets have been chosen in order to evaluate the performance of the proposed method: MSRAction3D (Microsoft Action 3D dataset) [9], UTD-MHAD (University of Texas at Dallas - Multimodal Human Action Dataset) [29] and MAD (Multimodal Action Dataset) [30]. Each of those datasets provides depth map data and posture data that are suitable to construct the DMI and MJD descriptors. Each dataset has a common testing settings that are used by the state of the art methods. We follow the same testing settings to compare the proposed method 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 combinations of channels scores together based on the fusion operations. Nonetheless the results of the fused channels vary from a dataset to another, generally, the classification results of using MJD in channel $Ch1$ are better than using DMI in channel $Ch3$ on the three datasets, which reflects the performance of using posture representation over depth representation. However, the classification results of channel $Ch2$ using both representations DMI and MJD are better than both $Ch1$ and $Ch3$. On the other hand, the $Prod$ operation generally generates better results than the Max operation. Furthermore, the fusion results between the three channels are better than fusing just two channels. The comparison with existing methods is based on taking the maximum accuracy obtained from different fusion operations and the three channels outputs. Table II shows recapitulation of the classification accuracy of each CNN channel and the fusion operations on the three datasets.

A. MSRAction3D

MSRAction3D dataset is captured by Microsoft Kinect v1 depth camera, the dataset contains twenty 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 [13] are followed to evaluate the proposed 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.

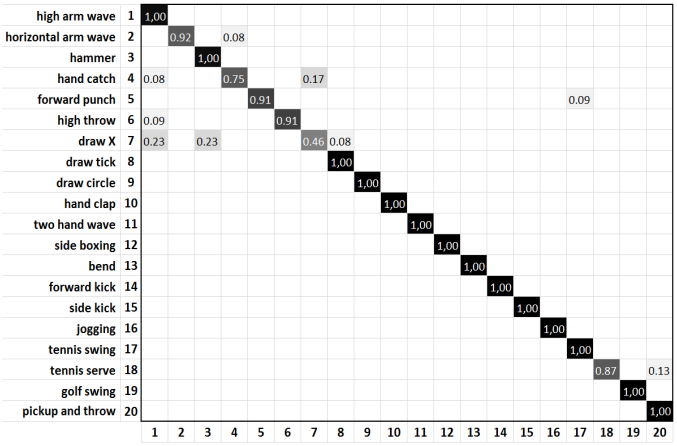

Fig. 9. Confusion matrix of the proposed method for the MSRAction3D dataset.

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

Method	Accuracy
HON4D [5]	88.89%
SNV [6]	93.09%
Range-Sample Feature [31]	**95.62%**
Random Occupacy Pattern [11]	86.50%
Bag-of-3D-Points [9]	74.70%
STOP [10]	84.80%
DSTIP [32]	89.30%
Proposed	**94.50%**

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

Method	Accuracy
EigenJoints [14]	81.40%
Actionlet Ensemble [13]	88.20%
DL-GSGC [15]	**96.70%**
HOJ3D [16]	78.97%
SSS Feature [17]	81.70%
MP Descriptor [18]	91.70%
High-level Skeleton Feature [19]	82.00%
Pose Set [20]	90.00%
Proposed	**94.50%**

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	90.48%	81.40%	88.21%	80.71%	82.86%	86.43%	88.21%	85.28%
Fus2	92.31%	85.17%	90.71%	87.14%	85.71%	88.57%	90.36%	88.50%
Fus3	87.91%	83.49%	83.21%	84.64%	88.21%	86.43%	90.36%	86.57%
Fus4	87.91%	85.12%	83.21%	85.71%	87.14%	88.50%	91.07%	87.13%
Fus5	91.21%	85.34%	90.36%	90,00%	91.07%	88.93%	95,00%	91.07%
Fus6	90.48%	84.42%	90.36%	88.57%	91.79%	88.93%	94.29%	90.79%
Fus7	90.84%	86.05%	88.93%	86.79%	91.43%	88.21%	93.57%	89.79%
Fus8	93.41%	88.14%	89.64%	88.57%	92.14%	89.64%	95.35%	91.07%
Fus9	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%**

Table II(Row 2: MSRAction3D) shows the classification accuracy results of each CNN channel and the fusion operations. The fusion score $Fus9$ achieved the best classification accuracy on this dataset, followed by $Fus8$. The classification result of the second channel $Ch2$ is better than both of $Ch1$ and $Ch3$. However, the fusion operations results are equal or better than the three CNN channels results. The maximum value of the results obtained from the fusion operations and the three CNN channels is $Fus9$ 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 proposed method accuracy is better than most existing depth-based approaches except [31]. In spite of the fact that the experiments setting of [31] 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 [15] which is based on sparse coding and temporal pyramid matching.

Generally, the proposed method performance over skeleton-based and depth based methods is due to the incorporation of depth features and posture features. Fig. 7 shows the difference between the fusion operations accuracies for the MSRAction3D dataset, and Fig. 11 shows the DMI and MJD of three actions, high arm wave, horizontal arm wave and hammer, associated with the classification accuracy shown in the confusion matrix (Fig. 9). In spite of the fact that the DMI appearance is mostly similar, the MJD is different features are different which helps in recognizing the actions even when they are performed in mostly similar ways.

B. UTD-MHAD

UTD-MHAD was captured using a fusion of depth and inertial sensor data, it consists of 27 actions performed by

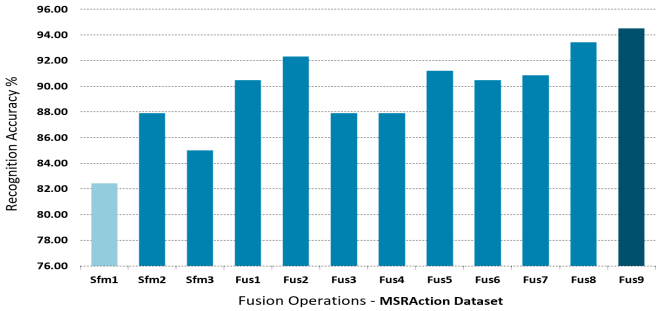

Fig. 10. The difference between the fusion operations accuracies of the MSRAction3D dataset.

8 subjects. Each subject repeated each action 4 times. The actions are represented in a form of depth and 3D poses frame sequences. The actions represented in this dataset 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 [29].

Table II(Row 3: UTD-MHAD) shows the classification results of the three CNN channels and the fusion operations. In this dataset, the $Fus8$ achieved the highest classification accuracy by 88.16%. Similar to MSRAction3D dataset, the classification result of the second channel $Ch2$ is better than both classification results of $Ch1$ and $Ch3$. As the maximum

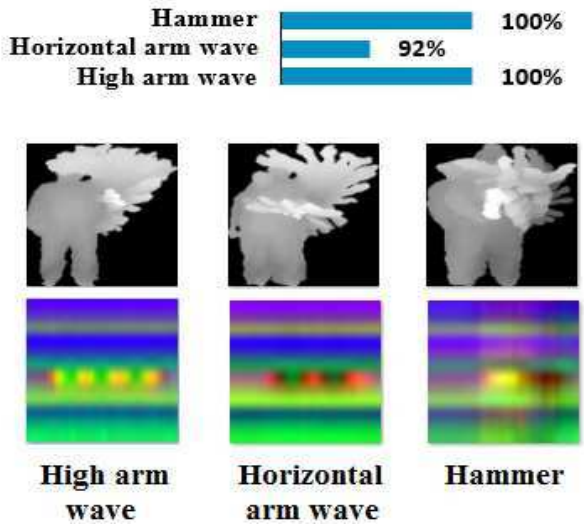

Fig. 11. Classification accuracy of three similar actions in appearance from MSRAction3D dataset that shown in the confusion matrix (Fig. 9).

accuracy value generated from the fusion operation $Fus8$, it is considered for the comparison with the results of existing methods that have been tested on the UTD-MHAD dataset. Table V shows the comparison results. Although there is no many works have been tested on this dataset like MSRAction3D, the proposed method achieved better results than the best recent method [12].

Fig. 13 shows the difference between the fusion operations accuracies on the UTD-MHAD dataset and Fig. 14 presents three very similar actions, clap, arms cross, and boxing. As it is shown in the confusion matrix (Fig. 12), 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 in cases where there is a very small difference between in the motion. However, the arms cross action is fully recognized because it is relatively different from clap and boxing actions.

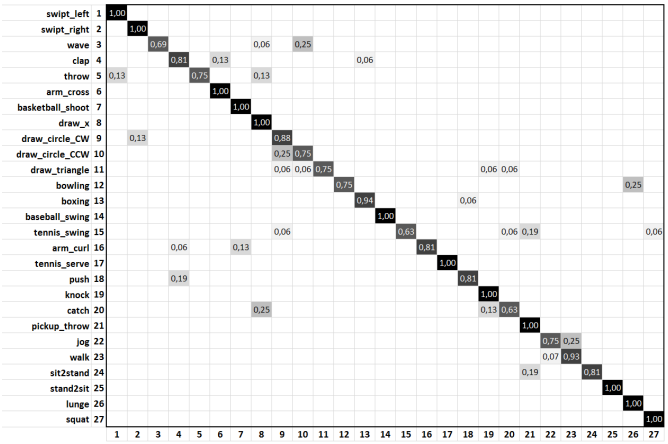

Fig. 12. Confusion matrix of the proposed method for the UTD-MHAD dataset.

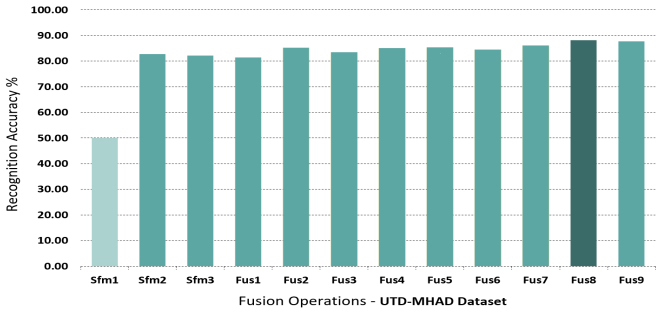

Fig. 13. The difference between the fusion operations accuracies of the UTD-MHAD dataset.

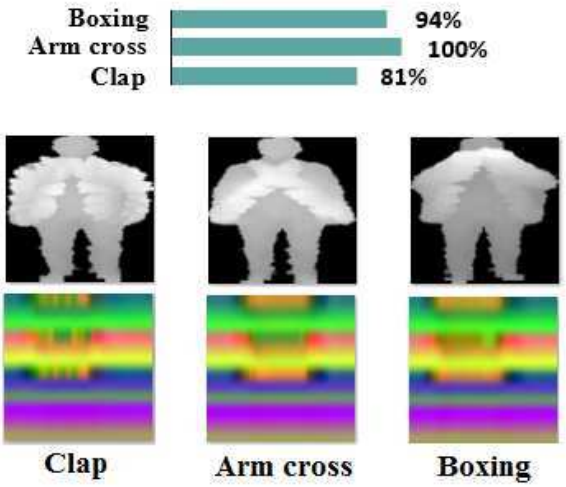

Fig. 14. Classification accuracy of three similar actions in appearance from UTD-MHAD dataset that shown in the confusion matrix (Fig. 12).

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

Method	Accuracy
Kinect and Inertial [29]	79.10%
SOS [22]	86.97%
Joint Trajectory Maps [12]	87.90%
Proposed	**88.14%**

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 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 were standing far from the background, we removed the background based on a threshold depth. The testing evaluation protocol used for this dataset is 5-folds cross-validation, the same as protocol described in [33]. Namely, 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 are used for testing. This process should be performed five times involve all the data in training and testing process. The final accuracy precision is the average of the five testing results.

Table II(Rows 4-9: MAD) represents detailed classification results of the three CNN outputs and the fusion operations from each fold of 5-fold cross-validation test and the average of the five tests. The maximum accuracy value of the results is generated from the fusion operation $Fus9$ by 91.86%. The proposed method achieved better results than the only existing method that jointly analyses video events with precise temporal localization and classification by modeling arbitrary transition patterns between events [33].Table VI shows the comparison results and Fig. 13 shows the difference between the fusion operations accuracies for MAD dataset.

Fig. 15 shows the confusion matrix of the proposed method on the MAD dataset, and Fig. 17 shows four mostly similar actions in appearance, left arm wave, left arm pointing to the ceiling, left arm punch and left arm throw. While the four actions performed with left hand to the top, the DMI descriptors look relatively similar. However, the Moving Joint Descriptors (MJD) carries different features which support the feature similarity of the depth appearance. The classification results of the four actions vary from 85% to 95% as presented in the confusion matrix (Fig. 15), which reflects the efficiency of combining depth and posture data for action recognition.

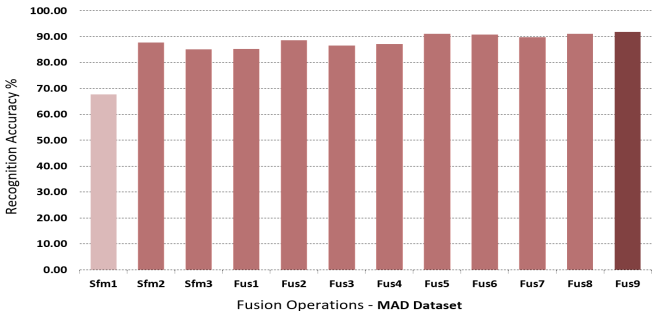

Fig. 16. The difference between the fusion operations accuracies of the MAD dataset.

D. Computation Complexity

1) Preprocessing Time: The preprocessing time includes the computation of DMI and MJD descriptors. The input of DMI descriptor is a grey image of size 112x112 pixels and the input of MJD descriptor is a matrix of size 15x3 of joints coordinates. The difference in the input size influences widely on the computation time as clearly shown in Fig. 19. For example, an action of 65 frames needs 0.835 seconds for calculating the DMI descriptor and 0.029 seconds for calculating 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 15 frames (from 65 to 80 frames) the computation increases with 0.249 for the DMI and 0.012 for the MJD. The results are shown in Fig. 19 is calculated on CPU with a machine of Intel(R) Core(TM) i7-6700 @ 3.40GHZ 3.40GHZ, 8 GB of RAM and 64 operating system.

2) Training and testing Time: The training time differs from a dataset to another, depending on the number of the 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 two other datasets that have more training data. From Table VII we notice that the training time and the number of iterations required for the model to converge are subject to the 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 accuracy average, the computation training time for this dataset is the sum of the five training durations. After training the model with the two descriptors, the trained weights are used to predict the action of a new data which is unseen in the training. While

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

Method	Accuracy
Event Transition [33]	85.02%
Proposed	**91.86%**

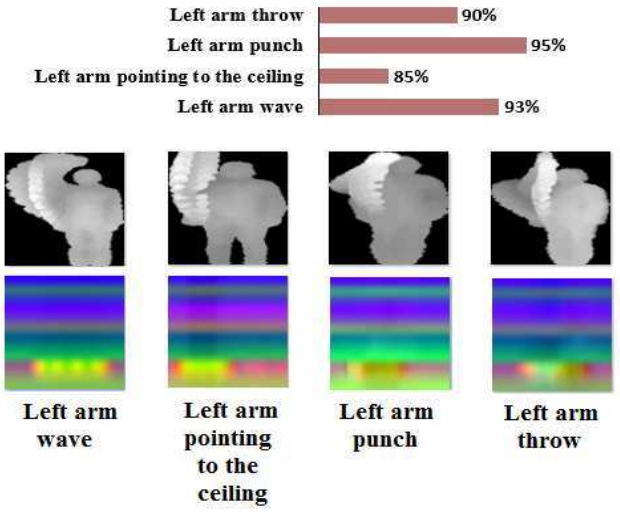

Fig. 17. Classification accuracy of four similar actions in appearance from MAD dataset that shown in the confusion matrix (Fig. 15).

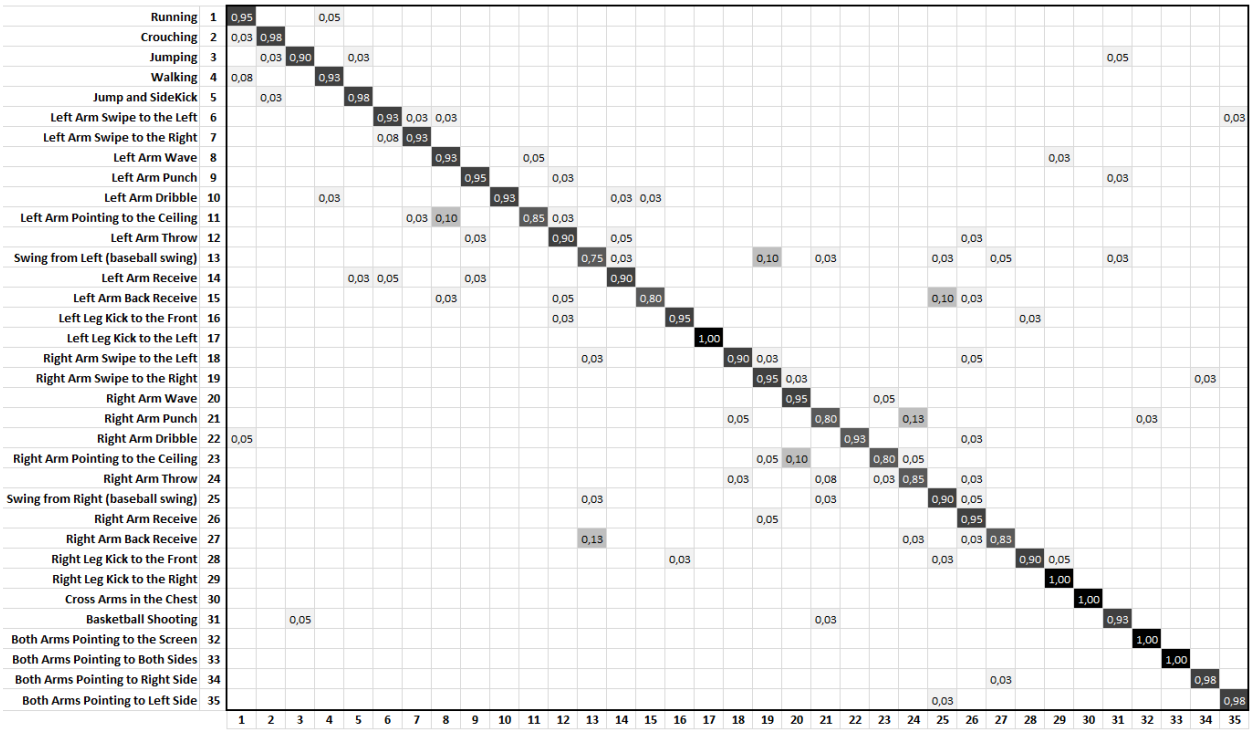

Fig. 15. Confusion matrix of the proposed method for the MAD dataset.

TABLE VII
TRAINING AND TESTING TIME OF THE THREE DATASETS.

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

the structure of the model used for training is the same for the three dataset as well as the type of training data, the processing time of action prediction of an input pair from any dataset is the same (0.07 seconds), but for MAD dataset, the testing for one input require averaging the prediction accuracies from the five trained models of the 5-folds, which results in 0.07x5 seconds computation time. The hardware material used for testing and training is different from the one used for the preprocessing. A server with GPU and Intel(R) Xeon(R) CPU E5-2630 v4 @ 2.20GHz 16 GB of RAM.

3) Discussion : 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 material performance used for computation, which makes it difficult to compare the proposed work with existing approaches in term of computation complexity. If we want to compare the computation time of the proposed work with the existing works, we must take two aspects into consideration, The descriptor computation time and the classification algorithm complexity. Some existing methods such as [5] and [34] use only one type of input data, either depth maps or posture data to create a descriptor. However other methods such as [12] use three descriptors, to cover the human from different views. In our case two input descriptors are computed and one of them requires less computation than the other, to this end we can classify the proposed method as in the middle rank of the existing methods in term of descriptors creation.

Most of the mentioned methods in the related work section use SVM as a classifier, such as [5]. Generally, SVM computation time is less then Neural Networks, but it also depends on how the Neural Networks model is deep. The methods based on deep neural network for classification generally use deep models to improve the accuracy. However, the approaches

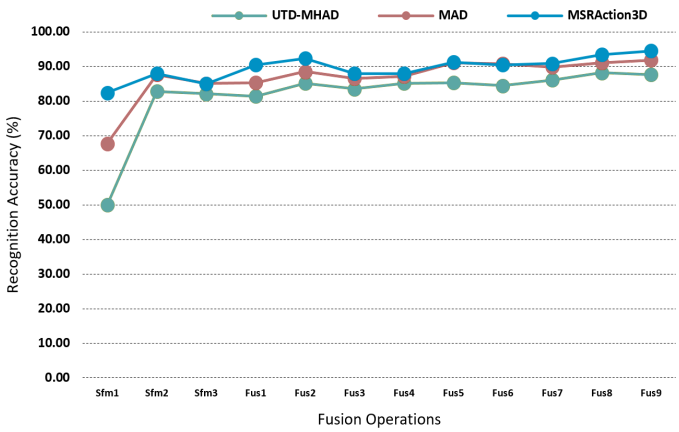

Fig. 18. Comparison of the fusion operations performances on the three datasets.

based on using CNN like ours, require more computation then using feed forward neural network due to the 2D processing. Even the CNN approaches differs by the number of layers for the processing. Additionally, one CNN channel is less computationally demanding then three channels. In this case, we can rank the processing time of the proposed method in term of classification among the high computationally demanding methods.

As previously highlighted in the introduction, the proposed approach offers many possibilities on how to use the data and the model with the fusion operations. 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. In this case, the proposed method can be classified among the most efficient methods in term of computation.

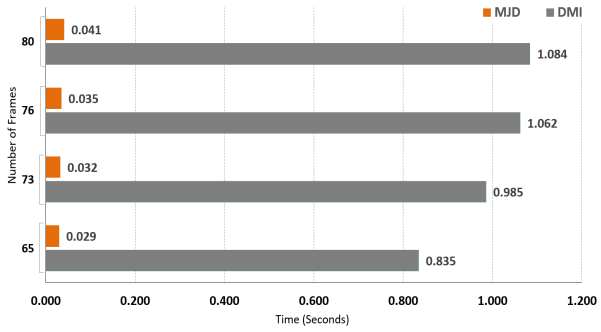

Fig. 19. The computation time of the DMI and MJD descriptors according to the number of frames action.

VI. Conclusion

A method for human action recognition from depth map and posture data using deep convolutional neural networks has been proposed. Two action representations and three convolutional neural networks channels were used to maximize feature extraction by fusing the results of the three CNN channels together. The method has been evaluated on three public benchmark datasets. The classification accuracy of the three datasets are better than most existing state of the art methods that are based on either depth data or posture data. This work claims that different action representations provide different cues. One representation carries action features that are absent in the other representation. In spite of the fact that CNN proved its power for feature extraction and classification in many computer vision problems, even a good CNN model can't classify the action correctly when the input doesn't provide discriminative features, which is the motivation behind the proposed framework. Fig. 18 shows the stability of the proposed method on the three datasets. The recognition behavior of the fusion operations is mostly the same. If a fusion operation accuracy is good on one dataset, it is also good on the two other datasets as well.

References

[1] W. Chi, J. Wang, and M. Q.-H. Meng, "A gait recognition method for human following in service robots," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2017.

[2] J. Yu and J. Sun, "Multiactivity 3-d human pose tracking in incorporated motion model with transition bridges," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2017.

[3] G. Liang, X. Lan, J. Wang, J. Wang, and N. Zheng, "A limb-based graphical model for human pose estimation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2017.

[4] D. Kim, W.-h. Yun, H.-S. Yoon, and H.-S. Jaehong, "Action recogntion with depth maps using hog descriptors of multi-view motion," in *The Eighth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*. UBICOMM, 2014, pp. 2308–4278.

[5] O. Oreifej and Z. Liu, "Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 716–723.

[6] X. Yang and Y. Tian, "Super normal vector for activity recognition using depth sequences," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 804–811.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[8] Y. Du, Y. Fu, and L. Wang, "Skeleton based action recognition with convolutional neural network," in *Pattern Recognition (ACPR), 2015 3rd IAPR Asian Conference on*. IEEE, 2015, pp. 579–583.

[9] W. Li, Z. Zhang, and Z. Liu, "Action recognition based on a bag of 3d points," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*. IEEE, 2010, pp. 9–14.

[10] A. Vieira, E. Nascimento, G. Oliveira, Z. Liu, and M. Campos, "Stop: Space-time occupancy patterns for 3d action recognition from depth map sequences," *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, pp. 252–259, 2012.

[11] J. Wang, Z. Liu, J. Chorowski, Z. Chen, and Y. Wu, "Robust 3d action recognition with random occupancy patterns," in *Computer vision–ECCV 2012*. Springer, 2012, pp. 872–885.

[12] P. Wang, W. Li, C. Li, and Y. Hou, "Action recognition based on joint trajectory maps with convolutional neural networks," *arXiv preprint arXiv:1612.09401*, 2016.

[13] J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Mining actionlet ensemble for action recognition with depth cameras," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 1290–1297.

[14] X. Yang and Y. L. Tian, "Eigenjoints-based action recognition using naive-bayes-nearest-neighbor," in *Computer vision and pattern recognition workshops (CVPRW), 2012 IEEE computer society conference on*. IEEE, 2012, pp. 14–19.

[15] J. Luo, W. Wang, and H. Qi, "Group sparsity and geometry constrained dictionary learning for action recognition from depth maps," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 1809–1816.

[16] L. Xia, C.-C. Chen, and J. Aggarwal, "View invariant human action recognition using histograms of 3d joints," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*. IEEE, 2012, pp. 20–27.

[17] X. Zhao, X. Li, C. Pang, X. Zhu, and Q. Z. Sheng, "Online human gesture recognition from motion data streams," in *Proceedings of the 21st ACM international conference on Multimedia*. ACM, 2013, pp. 23–32.

[18] M. Zanfir, M. Leordeanu, and C. Sminchisescu, "The moving pose: An efficient 3d kinematics descriptor for low-latency action recognition and detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 2752–2759.

[19] D. Wu and L. Shao, "Leveraging hierarchical parametric networks for skeletal joints based action segmentation and recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 724–731.

[20] C. Wang, Y. Wang, and A. L. Yuille, "An approach to pose-based action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 915–922.

[21] Q. Ke, M. Bennamoun, S. An, F. Sohel, and F. Boussaid, "A new representation of skeleton sequences for 3d action recognition," *arXiv preprint arXiv:1703.03492*, 2017.

[22] ——, "Skeleton optical spectra based action recognition using convolutional neural networks," *arXiv preprint arXiv:1703.03492*, 2016.

[23] J. Koushik, "Understanding convolutional neural networks," *arXiv preprint arXiv:1605.09081*, 2016.

[24] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, and G. Wang, "Recent advances in convolutional neural networks," *arXiv preprint arXiv:1512.07108*, 2015.

[25] E. Park, X. Han, T. L. Berg, and A. C. Berg, "Combining multiple sources of knowledge in deep cnns for action recognition," in *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*. IEEE, 2016, pp. 1–8.

[26] M. Lin, Q. Chen, and S. Yan, "Network in network," *arXiv preprint arXiv:1312.4400*, 2013.

[27] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014, pp. 675–678.

[28] 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.

[29] 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 *Image Processing (ICIP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 168–172.

[30] D. Huang, S. Yao, Y. Wang, and F. De La Torre, "Sequential max-margin event detectors," in *European conference on computer vision*. Springer, 2014, pp. 410–424.

[31] C. Lu, J. Jia, and C.-K. Tang, "Range-sample depth feature for action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 772–779.

[32] L. Xia and J. Aggarwal, "Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2834–2841.

[33] Y. Kim, J. Chen, M.-C. Chang, X. Wang, E. M. Provost, and S. Lyu, "Modeling transition patterns between events for temporal human action segmentation and classification," in *Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on*, vol. 1. IEEE, 2015, pp. 1–8.

[34] 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," *Journal of Visual Communication and Image Representation*, vol. 25, no. 1, pp. 24–38, 2014.

[35] C. Chen, R. Jafari, and N. Kehtarnavaz, "Action recognition from depth sequences using depth motion maps-based local binary patterns," in *Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on*. IEEE, 2015, pp. 1092–1099.

View publication stats