3D CNNs on Distance Matrices for Human Action Recognition

Alejandro Hernandez Ruiz
IRI, CSIC-UPC
Barcelona, Spain
ahernandez@iri.upc.edu

Lorenzo Porzi
IRI, CSIC-UPC
Barcelona, Spain
lporzi@iri.upc.edu

Samuel Rota Bulò
Fondazione Bruno Kessler
Trento, Italy
Mapillary
Graz, Austria
rotabulo@fbk.eu

Francesc Moreno-Noguer
IRI, CSIC-UPC
Barcelona, Spain
fmoreno@iri.upc.edu

ABSTRACT

In this paper we are interested in recognizing human actions from sequences of 3D skeleton data. For this purpose we combine a 3D Convolutional Neural Network with body representations based on Euclidean Distance Matrices (EDMs), which have been recently shown to be very effective to capture the geometric structure of the human pose. One inherent limitation of the EDMs, however, is that they are defined up to a permutation of the skeleton joints, i.e., randomly shuffling the ordering of the joints yields many different representations. In oder to address this issue we introduce a novel architecture that simultaneously, and in an end-to-end manner, learns an optimal transformation of the joints, while optimizing the rest of parameters of the convolutional network. The proposed approach achieves state-of-the-art results on 3 benchmarks, including the recent NTU RGB-D dataset, for which we improve on previous LSTM-based methods by more than 10 percentage points, also surpassing other CNN-based methods while using almost 1000 times fewer parameters.

1 INTRODUCTION

In recent years, 3D sensing technologies, and in particular RGBD cameras have become increasingly cheap and widely available. Devices such as the Kinect or Leap Motion, and the associated software libraries, allow for accurate 3D tracking of human body parts with minimal effort. Because of this, human action recognition algorithms working directly with 3D skeletal data have gained substantial popularity in the research community. As in many other fields of research, remarkable results have recently been obtained in this task by employing deep learning-based approaches [16, 22, 26, 36, 38] exploiting large-scale datasets [22].

Human action recognition from 3D skeletal data is inherently a sequence-based problem, which can be naturally tackled in the context of deep learning using recurrent networks. Indeed, many works [16, 22, 36, 38] propose Long Short-Term Memory (LSTM) networks working directly on the 3D coordinates of the body joints [16, 38] or hand-crafted geometric features derived from these [36]. An alternative, more recent approach is that of projecting and color coding the joint trajectories in image space [17, 31], obtaining a compact representation of the whole skeleton sequence that can be processed using standard convolutional networks. This allows the authors of [17, 31] to reuse well-tested CNN architectures (*i.e.* AlexNet [15]) and to exploit large scale image datasets (*i.e.* ILSVRC2012 [21]) to pre-train their networks, obtaining the current state of the art on a challenging large-scale action recognition dataset [22].

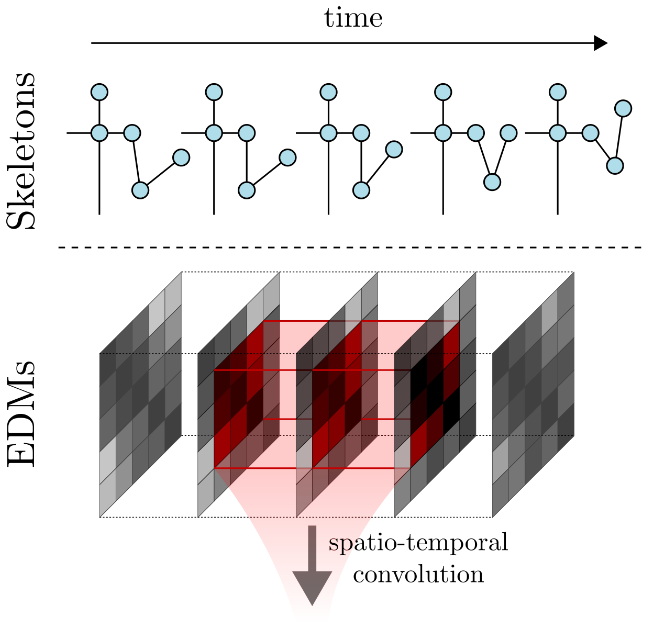

Figure 1: We encode sequences of skeletons in 3D space as stacked Euclidean Distance Matrices. By performing convolution both along the spatial and temporal dimensions our network learns to respond to the spatio-temporal dynamics of the input data.

Following from the success of [17, 31], we propose a novel action recognition CNN based on the ResNeXt architecture of Xie *et*

MM '17, October 23–27, 2017, Mountain View, CA, USA.
© 2017 ACM. ISBN 978-1-4503-4906-2/17/10…$15.00
DOI: https://doi.org/10.1145/3123266.3123299

al. [33]. Differently from these works, however, we explicitly take the temporal nature of our problem into account, by constructing a network composed of *spatio-temporal* convolution and pooling operators. Furthermore, instead of the indirect, image-based representation employed in [17, 31], we encode the input skeletons as sequences of Euclidean Distance Matrices (EDM) computed over their joints. EDMs are rigid transformation-invariant representations of sets of points, which can be effectively processed by convolutional networks, as recently shown in [19]. Intuitively, as depicted in Fig. 1, convolutional neurons can learn to respond to local EDM structures which encode the spatial configurations of groups of 3D joints. By performing convolution also along the *time* dimension, we allow the network to learn to respond to the spatio-temporal dynamics of our data.

Euclidean Distance Matrices, however, are defined up to a permutation of the points they represent. By changing the order of the joints when computing an EDM, the region of the skeleton associated with each local neighborhood of the matrix would also change. This would lead the same convolutional neurons to respond in a radically different way to the same skeleton. To sidestep this problem, we augment our EDM representation with a learned combination of the input points. Thus, we endow our network with the capability of selecting the most advantageous distance matrix configuration for the purpose of classifying actions. This only adds a negligible amount of parameters to the overall model, which can be trained end-to-end together with the rest of the network.

We evaluate our network, named DM-3DCNN, on three benchmark datasets, including the challenging and recently released NTU RGB-D [22], obtaining competitive results w.r.t. other recent methods. On this particular dataset, we obtain the new state of the art, surpassing previous LSTM-based approaches by an average of 10 % accuracy and the CNN-based approach in [17] by 3 %, while using about 1000 times fewer parameters and operations.

Contributions. To summarize, this paper includes three main contributions. *First*, we present an approach to human action recognition from 3D skeletal data represented as sequences of Euclidean Distance Matrices. To overcome the permutation ambiguity inherent in encoding the skeletons as EDMs, we compute the distance matrices from a learned combination of the joints which is general enough to include all possible permutations. *Second*, we propose a ResNeXt-inspired [33] network architecture built from spatio-temporal convolution and pooling operators and taking sequences of EDMs as input. *Third*, we evaluate our approach on three benchmarks, obtaining the new state of the art on the large-scale NTU RGB-D dataset and surpassing recent CNN-based approaches [17, 31] while employing three orders of magnitude fewer parameters and operations.

2 RELATED WORK

Human action recognition is one of the most interesting topics of computer vision, and it has many use cases within the academy, robotics, surveillance, games and entertainment multimedia; because of this, the quantity and diversity of works is impressive, and impossible to cover in a single work. We will focus then in reviewing the works we consider relevant to this particular problem and to our approach.

The diversity of works and datasets also imply diversity in the representation of the actions. We will consider two main representations: skeletal data sequences and RGBD sequences.

2.1 Skeleton Sequence based Methods

Before the advent of the deep learning methods, most works focused in designing representation of the movements that could be learned or hand crafted, and allowed a simple model to classify or make inference on them.

The first type of methods are based in Support Vector Machines (SVMs), which are simple yet powerful classifiers that can easily learn to discriminate the sequences, when the features are able to convey the relevant information. To accomplish that, these features need to be both invariant to changes in the viewpoint and length of the sequence. In [13] the authors encode the sequence using temporal and view invariant representations based on different distance measures between the joints in a skeleton, computed in time and space dimensions. Similarly, [11] encodes the sequence using covariance features of the joint locations in a small time window. In [32], sequences were represented by Histograms of the location of 3D joints (HOJ3D), and in [20], by spatio-temporal Histogram of Gradients (HoGs) in the joint angles. Another way to abstract the complexity of the movements is to learn dictionaries to group and classify them easily [18, 37]. A very abstract representation can be found in [27], where the sequences are transformed into lie groups, mapped to its lie algebra, and passed through Dynamic Time Warping (DTW) for aligning and applied pyramidal fourier analysis. A different approach was proposed in [25], were a Gaussian Mixture Model (GMM) was used to learn a compact representation of the sequence and a hierarchical Hidden Markov Model (HMM).

Alternatively, instead of classifying the whole sequence at once, we can classify the frames of such sequence, and then combine this classification in a global prediction. The idea is to be able to identify certain important frames or instances, that unequivocally identify the sequences. Roughly following this idea we find a family of methods called Multiple Instance Learning [34, 35].

More recently, a number of works leverage on the deep learning methods that are the state of the art in many pattern recognition problems. These works focus on creating models with deep architectures, that are capable to learn to classify the sequences from a very simple encoding or even raw data. We can find two main groups of architectures: those based in Recurrent Neural Networks (RNN) and variants like the Long Short Term Memory [9] (LSTM) network; and on the other hand the Convolutional Neural Networks (CNN) based methods.

The following models inspire their architectures by taking into account the separation of the body into parts of interest (e.g. left and right arms/legs, and the central part composed by torso, neck and head). The Hierarchical-RNN [5] takes each body part as the input to a separate RNN and combines them into a unique output which is used for classification. [38] proposes Co-occurrence LSTM, a modification of the LSTM cell that seeks to capture the correlation between the different parts involved in a movement. With similar inspiration, [24] proposed an LSTM with attention model that reweighs the relevance of the joints in the input at each time frame. And also we have the Part-aware LSTM [22], that modifies the LSTM cell to have the input to hidden state gates explicitly separated by each body part.

One of the most important factors to measure the information in a frame is the amount of movement. Some models build upon this idea. For instance, [26] proposes a modified LSTM model which incorporates the magnitude of the difference with respect to previous frame into the gating mechanisms of the cell. [16], introduces the ST-LSTM, a modified LSTM model that scans the input in a path following fashion over a graph, and such graph is constructed by unrolling the time component of the input; ST-LSTM also incorporates a trust gate to the cell, that basically takes an inverse distance measurement that reweighs the input to the empirically estimated trust over such input.

Regarding the CNN based methods, one of the first approaches was [4], which converted the sequences to images by representing the skeleton as a vector of pixels, and concatenating these vectors along the temporal dimension to create a single RGB image; this image was then fed to a CNN to perform action classification. More recently, a group of models leverage on widely known CNN architectures like AlexNet [15] to perform human action recognition. In these models some transformations and projections of the sequence are performed, yielding a single color image that serves as input to the network: in [31] the 3D skeleton is projected as a 2D image by framing it in the point of view of a camera along the X, Y or Z axis; the temporal dimension is converted into a color code, and finally the 2D image used as input is the superposition of all frames. It is worth to note also, that reusing known architectures is very convenient, as it allows for efficient training, and also allows for building ensemble models, that usually perform better than the single network models; in [31] the output of three networks are combined to produce an ensemble classifier. [17] follows a very similar approach, the main difference being that the projection is done over a transformed coordinate system instead of the Cartesian system, and that the ensemble model combines ten networks instead of three.

2.2 Image Sequence based Methods

Differently to the skeleton based methods, image sequence methods do not explicitly model the human body. Instead they work with raw pixels, and try to find meaningful patterns in the distributions of pixels in each frame and its transformations over the time.

To apply convolutions over a sequence of images we can either represent the input sequence as a 3D volume of information over which we can apply 3D convolutions, or we can take the time dimension in the input as different channels and perform 2D convolutions. The first approach in this category was [12] which performed 3D convolutions over the video, in a similar manner as the 2D convolutions were applied to a single image. A number of convolutional layers were used to extract features and the network output was produced by a dense layer and a softmax classifier. This relatively straightforward approach is however limited however by a high computational cost. For this reason, other types of models using 2D CNNs were proposed: the family of models comprised by stream networks [23, 29], take as input two different representations, a single image and the optical flow of the sequence; these two representations are passed through a parallel inference architecture, that fuses the extracted features in its topmost layer. Another option is to represent the whole video as a single image, by transforming its representation to a sort of histogram over the movements in the video [2, 30]. Also worthwhile of mention models that take mixed approaches. For instance, in [10] 2D CNNs are used to extract features, and 3D CNNs over the computed features fuse the spatial and temporal information.

2.3 Other Methods

There are a few other methods that influence this work, and are important to mention. In the related problem of human pose estimation, we can find [19], where the authors show that the use of CNNs over EDMs as view invariant representation of the skeletons produces good results. Also, we consider the ResNeXt [33] architecture, as a reference of a top performing modern CNN architecture.

Drawing inspiration, ideas and design choices from all the works mentioned in this section, we propose a novel method that mixes them in a sensible way and produces excellent results for this problem.

3 METHOD

Our goal is to recognize human actions or interactions from temporal skeleton data, *i.e.* sequences of human body joints encoded as 3D points varying over time. In general, the sequences might comprise multiple human bodies, but we restrict our focus to interactions that involve up to two actors, since this is the setting commonly encountered in the main benchmark datasets (*i.e.* NTU RGB-D dataset [22] and SBU Interaction dataset [35]).

The problem setting is given as follows. Let J be the number of skeleton joints and let $\mathcal{S} \subset R^{3\times\mathsf{J}}$ be the set of all possible configurations of joints for a single body skeleton at some fixed time. For convenience we simply call skeletons the elements of $\mathcal{S}$. Joints are given as 3D points and each skeleton forms a $3 \times \mathsf{J}$ matrix with joints as columns. The input space for our action (or interaction) recognition problem is given by $\mathcal{X} = (\mathcal{S} \times \mathcal{S})^{\mathsf{L}}$, *i.e.* sequences of L pairs of skeleton configurations, where L is a fixed time-window length. We consider pairs of body skeleton in the sequence to account for up to two actors in the scene. In case only a single actor is in the scene, this will be repeated twice to form a pair. The set of possible actions (or interactions) to be predicted is denoted by $\mathcal{Y} = \{1, \dots, \mathsf{K}\}$ and the action recognizer is a function $f_\theta : \mathcal{X} \to \mathcal{Y}$ parametrized by θ that assigns action labels in $\mathcal{Y}$ to sequences of pairs of skeletons in $\mathcal{X}$. The set of feasible parameterizations is denoted by Θ.

The state-of-the-art methods for human action recognition from skeleton data follow two main approaches: i) implementing the action recognition function f_θ as a deep recurrent neural network; ii) rendering the skeleton sequences as a single image and implementing f_θ as a deep CNN taking these images as input. Our solution is closer in spirit to the latter type of approaches, but instead of encoding the sequence of skeletons as a single image, we retain one additional spatial dimension for representing skeletons and introduce by construction invariance to rigid transformations. This is indeed our *first contribution* that we implement by using distance matrices defined over a learned transformation of the skeleton data. Distance matrices are in fact invariant to rigid transformations by

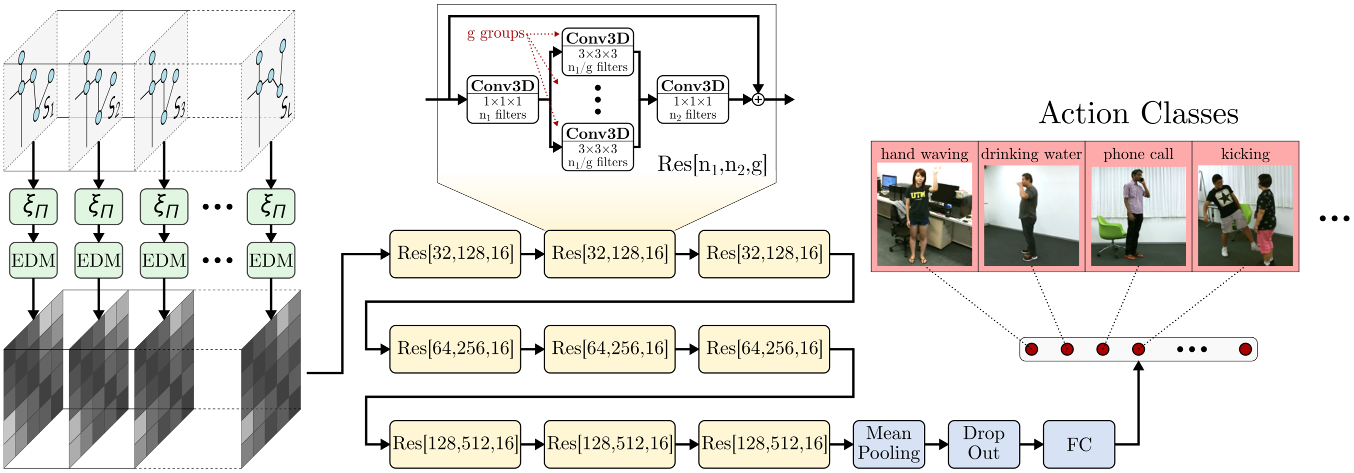

Figure 2: Overview of the proposed network architecture. Skeletons from the input sequence are transformed using a learned linear function ξ_Π and the output is used to compute a sequence of Euclidean Distance Matrices (green blocks). The matrices are stacked to form a 3D tensor, which is fed to a fully-convolutional network built from ResNeXt units [33] with 3D convolutions (yellow blocks). Finally, a classifier composed of global mean pooling, dropout and a fully connected layer with soft-max (blue blocks) is used to compute the predicted probability distribution over action classes (red blocks). Note that we are not showing the case of multiple skeletons per frame for the sake of simplicity.

nature. Our *second contribution* consists in adopting a deep neural network with spatio-temporal convolutional operators applied to distance matrices extracted from skeleton data. This contrasts with the approaches in the literature that typically rely on standard CNNs, or recurrent neural networks.

3.1 EDMs over Transformed Skeletons

Many approaches in the literature [5, 25, 38] do not feed the action recognizer with the original skeleton data, but try manipulate the input to enforce some form of invariance to rigid transformations. This is indeed empirically shown to be beneficial to the human action recognition task. For instance, in [5] the skeletons are put into a canonical reference system through a change of coordinates. The solution we pursue to achieve invariance to rigid transformations is different and consists in representing skeletons in terms of distance matrices, which are inherently invariant to rigid transformations.

Given a $3 \times \mathsf{M}$ matrix Z of 3D points we define the corresponding Euclidean distance matrix $\mathsf{D} = \mathtt{edm}(\mathsf{Z}) \in \mathbb{R}_+^{\mathsf{M}\times\mathsf{M}}$ as the nonnegative, symmetric $\mathsf{M} \times \mathsf{M}$ matrix with (i, j)th entry given by the squared Euclidean distance between the ith and jth columns of Z. That is, $\mathsf{D}_{ij} = \|\mathsf{Z}_i - \mathsf{Z}_j\|^2$, where $\|\cdot\|$ is the Euclidean norm, and $\mathsf{Z}_i \in \mathbb{R}^3$ is the ith column of matrix Z. Distance matrices are invariant to rigid transformations (*e.g.* translations, rotations, reflections) applied to the original points and thus suit well our purpose of having a representation for skeletons that is invariant to such transformations. However, distance matrices are not invariant to permutation. This means that the EDM computed from a skeleton is sensitive to the ordering of the joints. In general, permutations might exist that have a negative impact on the final performance of the action recognition task (see Sec. 4.4), in particular if we aim to exploit local structures of the distance matrix via convolutional neural networks (see next subsection). To overcome this issue, we propose to learn a transformation of the skeleton that is sufficiently general to represent all possible permutations, and compute the distance matrices of transformed skeletons. By having this component embedded into the neural network, we give the classifier the freedom of emphasizing the importance of some joints and potentially discovering an optimal permutation of the joints that enhances local structures in the distance matrices. At the same time we preserve an invariance to rigid transformations in all layers that follow the distance matrix computation. In this work we keep the skeleton transformation simple by considering a linear operator

$$\xi_\Pi(\mathsf{S}) = \mathsf{S}\Pi\,,$$

acting on $\mathcal{S}$, where $\Pi \in \mathbb{R}^{\mathsf{J}\times\mathsf{J}}$ is a $\mathsf{J} \times \mathsf{J}$ real matrix to be learned. This transformation encompasses permutations of joints as a special case.

Next we deal with the problem of encoding pairs of skeletons in terms of EDMs, since our problem setting assumes up to two body skeletons in the scene. Accordingly, let $\mathsf{S}, \hat{\mathsf{S}} \in \mathcal{S}$ be two skeletons and remind that $\mathsf{S} = \hat{\mathsf{S}}$ if a single skeleton is present. There are different ways in which the two skeletons can be encoded using an EDM representation. We consider the following two approaches:

Decoupled encoding. The first approach simply encodes S and $\hat{\mathsf{S}}$ independently, after undergoing the transformation ξ_Π, into $\mathtt{edm}(\xi_\Pi(\mathsf{S}))$ and $\mathtt{edm}(\xi_\Pi(\hat{\mathsf{S}}))$, and stacks the two representations as they were two separate feature channels. As a result we obtain a $\mathsf{J} \times \mathsf{J} \times 2$ tensor representing the two skeletons.

Coupled encoding. The second approach concatenates the two skeletons into a single matrix of points that we denote as $\mathsf{S}|\hat{\mathsf{S}} \in \mathbb{R}^{3\times 2\mathsf{J}}$ and uses the distance matrix $\mathtt{edm}(\xi_\Pi(\mathsf{S}|\hat{\mathsf{S}}))$ as encoding. This yields a $2\mathsf{J} \times 2\mathsf{J}$ matrix representation for the two skeletons.

The encoding of skeleton data that we have detailed for a pair of skeletons is actually applied to the entire sequence of skeletons. This adds also the temporal dimension to the representations mentioned above, yielding a $\mathsf{L} \times J \times J \times 2$ tensor if we opt for the independent encoding scheme, and a $\mathsf{L} \times 2\mathsf{J} \times 2\mathsf{J}$ if we apply the joint encoding instead, where L is the temporal window length.

3.2 3D CNNs over Distance Matrices

The application of CNNs to distance matrices has recently proven effective to tackle the problem of human pose regression from skeleton data [19]. Indeed, distance matrices exhibit rich local structures (up to permutations), which can be effectively learned by convolutional filters. In this work, we extend the ideas in [19] by considering *time* as an additional spatial dimension when performing convolution. This results in a 3D spatio-temporal convolution operator which allows our network to capture the temporal evolution of the local structures encoded by the EDMs,

Formally, given a tensor $\mathsf{Z} \in \mathbb{R}^{\mathsf{T}\times\mathsf{H}\times\mathsf{W}\times\mathsf{C}}$ and a convolutional filter $\mathsf{w} \in \mathbb{R}^{\mathsf{t}\times\mathsf{h}\times\mathsf{w}\times\mathsf{C}}$, we define 3D convolution $\star$ as

$$(\mathsf{w} \star \mathsf{Z})_{i,j,k} = \sum_{i'=1}^{\mathsf{t}} \sum_{j'=1}^{\mathsf{h}} \sum_{k'=1}^{\mathsf{w}} \sum_{c=1}^{\mathsf{C}} \mathsf{w}_{i',j',k',c} \mathsf{Z}_{i+i',j+j',k+k',c} \,,$$

where the first three dimensions of Z and w are interpreted as spatial dimensions, while C are the feature channels. In practice, 3D convolution can be used as a drop-in replacement for 2D convolution in most networks, providing us with great flexibility when defining our architecture.

Building from these spatio-temporal convolution operators, we propose a network architecture inspired by the recent ResNeXt of Xie *et al.* [33] (see Fig. 2). In particular, we adapt the configuration employed in [33] for the CIFAR-10 experiments, replacing each convolution with a spatio-temporal convolution and performing the final global average pooling both over the spatial and time dimensions. To partially compensate for the increased number of parameters in our kernels compared to the ones in [33], we reduce the number of filters in each layer by a factor 2. Furthermore, differently from [33], we perform dropout on the inputs of the final fully-connected layer. For additional details refer to Appendix A.

3.3 Network Training

Given a training set $\mathcal{T} \subset \mathcal{X} \times \mathcal{Y}$ we estimate an action recognition function $f_\theta : \mathcal{X} \to \mathcal{Y}$ by minimizing the regularized empirical risk

$$R(\theta; \mathcal{T}) = \frac{1}{|\mathcal{T}|} \sum_{(X,y)\in\mathcal{T}}^{\mathsf{N}} \ell(f_\theta(X), y) + \lambda\Omega(\theta) \,,$$

over Θ, where $\ell : \mathcal{Y} \times \mathcal{Y}$ is a loss function penalizing wrong predictions and $\Omega : \Theta \to \mathbb{R}$ is a regulariser. In our experiments, ℓ coincides with the standard log-loss and Ω with the ℓ_2 norm.

The minimization of the empirical risk is performed using stochastic gradient descent. Details about the hyperparameters of the optimizer are provided in the experimental section.

4 EXPERIMENTS

In the following Sec. 4.3 we study the performance of the proposed action recognition method by conducting an extensive evaluation on three benchmark datasets (see Sec. 4.1), including the recent large-scale NTU RGB-D [22]. Furthermore, in Sec. 4.4 we perform an in-depth ablation study to evaluate the effects of learning the shuffle matrix under different sets of constraints. Additional details about our network architecture and training procedure are reported in Sec. 4.2.

4.1 Datasets

In this work we consider three benchmark datasets, described in the following.

NTU RGB-D. The NTU RGB-D dataset [22] is, to the best of our knowledge, the largest-scale publicly available action recognition dataset. It contains over 56 thousand sequences, captured with multiple Kinect 2 sensors, of 40 actors performing 60 different actions in 17 different setups. Each action is repeated 2 times for each actor / setup pair and recorded from three different views at the same time. For each sequence, both RGB-D videos and 3D skeletons with 25 joints, automatically extracted using the Kinect 2 software, are made available. Depending on the action class, one or two actors can be present in the same sequence at the same time. Following the experimental protocol in [22], we consider both a cross-view and a cross-subject setting, splitting training and testing data on the basis of, respectively, the view from which the action is recorded or the actor performing it.

MSRC12 Gestures dataset. The MSRC12 Gesture dataset [6] contains 594 video sequences, captured with a Kinect, of 30 actors performing 12 actions. Each sequence contains several repetitions of the action of interest, for a total of 6244 action instances. As in [17, 31], we follow a cross-view evaluation protocol. Differently from NTU RGB-D, the skeleton detections provided with this dataset are computed using the Kinect v1 software, and contain skeletons with 20 joints.

SBU Interaction dataset. The SBU Interaction dataset [35] focuses solely on actions involving two interacting actors. It contains ≈ 300 sequences, subdivided in 21 sets, each containing one or two repetitions of each of 8 action classes, performed by a different pairing of subjects from a set of 7. Skeleton detections with 15 joints, extracted using the PrimeSense software, are provided together with the original RGB-D video sequences. In our experiments we follow the 5-fold cross-validation protocol also adopted in [35].

4.2 Implementation and Training Details

We train our DM-3DCNN network by stochastic gradient descent using the Adam [14] algorithm, with a batch size of 32 and parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$. When considering the NTU RGB-D and MSRC12 datasets, we adopt the following training schedule: we start with a learning rate of 10^{-3}, reducing it by a factor 10 after 40 epochs and again after 60 epochs, training for a total of 80 epochs. The network parameters are initialized following the method from [7]. For SBU, given its considerably small size, we train by fine-tuning from the network trained on NTU RGB-D: we initialize all convolutional filters from the values learned on NTU RGB-D, while learning the final fully connected classifier from scratch, and train for 300 iterations with an exponential learning rate decay from 10^{-4} to 10^{-5}. In all cases, the regularization factor (*i.e.* weight decay) is set to $\lambda = 5 \times 10^{-4}$. We implement

Table 1: Results on the NTU RGB-D dataset.

Method	Cross Subject	Cross View
LSTM-based methods		
Deep LSTM [22]	60.7 %	67.3 %
Part-aware LSTM [22]	62.9 %	70.3 %
ST-LSTM [16]	69.2 %	77.7 %
Multilayer LSTM in [36]	64.9 %	79.7 %
CNN-based methods		
Liu *et al.* [17] best single	73.5 %	84.0 %
Liu *et al.* [17] ensemble	80.0 %	87.2 %
DM-3DCNN	**82.0** %	**89.5** %

Table 2: Results on the MSRC12 dataset.

Method	Accuracy
Hand-crafted features	
LC-KSVD [37]	90.2 %
Cov3DJ [11]	91.7 %
CNN-based methods	
Du *et al.* [4]	84.5 %
Wang *et al.* [31]	93.1 %
Liu *et al.* [17] best single	93.2 %
Liu *et al.* [17] ensemble	**96.6** %
DM-3DCNN	95.8 %
DM-3DCNN ensemble of 5	**96.6** %

our networks using the TensorFlow [1] framework and run our experiments on a single Nvidia GTX 1080 GPU[1].

Following the experimental setting in [22], in all datasets we down-sample the input sequences along the temporal dimension by subdividing it into 20 equally spaced sections and randomly selecting a frame from each. During training a different random sampling is considered each time a sequence is fed to the network, as we observed that this provides a useful regularizing effect. As mentioned in Sec.3.1, depending on the dataset, we consider up to two input skeletons at each time step. Since the datasets considered in our experimental evaluation do not define any explicit semantic about the ordering of the skeletons, during training we randomly select which one is interpreted as S and $\hat{S}$ each time a sequence is loaded. The same sampling procedure, both for sequence down-sampling and skeleton swapping, is also performed at test time, and all results in the following are reported as the average of the accuracies over 10 independent runs.

4.3 Comparison with State of the Art

Before performing our main evaluation, we conduct a set of preliminary cross-validation experiments on held-out training data from NTU RGB-D and SBU, in order to select which EDM encoding to use (see Sec.3.1). Interestingly, we observe that for NTU RGB-D the *decoupled* encoding exhibits the best performance, while for SBU the *coupled* encoding is favored. This is not surprising: the SBU dataset is mostly focused on interactions, thus the cross-skeleton distances encoded in the coupled EDM contain valuable information for our network. On the other hand, interaction classes are a strict minority in NTU RGB-D, thus making the more compact representation of the decoupled encoding a better fit for this dataset. Note that we do not need to choose which encoding to use in the MSRC12 case, as it only contains sequences with one skeleton.

NTU RGB-D

Compared to most previous datasets [6, 28, 35], NTU RGB-D contains 1-2 orders of magnitude more data, captured with the improved Kinect 2 sensor. Nonetheless, we note that the skeleton detections provided with the dataset still contain a substantial amount of noise, to the point that, even for a human observer, it can be hard to recognize the actions just by looking at the skeletons.

In a first set of experiments, we compare the performance of our method against recent LSTM-based and CNN-based approaches to human action recognition from skeleton data. In particular, we consider: the part-aware LSTM in [22]; the spatio-temporal LSTM (ST-LSTM) in [22]; the multi-layer LSTM with geometric features in [36]; the ensemble of CNNs approach in [17]. For the method in [17] we report both the performance of the ensemble and that of the best single network in the ensemble. Finally, we also consider a plain 3-units LSTM baseline, as reported in [22].

Table 1 summarizes our results, highlight the advantages of our method when dealing with this large and challenging dataset. Compared to the best LSTM-based approach we observe an absolute increase in accuracy of ≈ 8 % in the cross-subject and ≈ 6 % in the cross-view setting. Similarly, we obtain a ≈ 2 % improvement over the CNN-based approach in [17] in both settings under exam. It is worth noting that the networks used in [17] have a considerably larger number of parameters than DM-3DCNN, *i.e.* $\approx 6 \times 10^7$ parameters for each network in the ensemble [15] and $\approx 6 \times 10^8$ overall, compared to 6.1×10^5 parameters in DM-3DCNN. This suggests that our EDM-based encoding is indeed more effective at representing sequences of skeletal data, compared to the image-based one adopted in [17], as our network is able to exploit it to obtain superior performance while using ≈ 1000 times less parameters.

MSRC12

In the next set of experiments, we focus our attention on the MSRC-12 dataset. Here we consider two traditional approaches based on hand-crafted features, *i.e.* LC-KSVD [37] and Cov3DJ [11]; and three CNN-based approaches, *i.e.* Du *et al.* [4], Wang *et al.* [31] and Liu *et al.* [17]. The results are reported in Tab.2. It is clear that the sequences in this dataset are considerably less challenging than those in NTU RGB-D, as most methods under exam are able to achieve greater than 90 % accuracy. Among the non-ensemble models, DM-3DCNN obtains the highest accuracy, also surpassing the ensemble of CNNs in [31]. Interestingly, DM-3DCNN performs better than the single best CNN of [17], while being slightly surpassed by their ensemble, at the cost of employing ≈ 1000 times more parameters and, consequently, ≈ 1000 times more operations. Furthermore, differently from DM-3DCNN, the networks in [17] also exploits

[1] The source code is available at: https://github.com/magnux/DMNN

Table 3: Results on the SBU Interaction dataset.

Method	Accuracy
Other LSTM-based methods	
HBRNN [5]	80.4 %
Deep LSTM [38]	86.0 %
Co-occurrence LSTM [38]	90.4 %
ST-LSTM [16]	93.3 %
DM-3DCNN	**93.7 %**

a vast amount of additional data, as they are pre-trained on the ILSVRC2012 data [21].

Given the results of Liu *et al.* [17], both in NTU RGB-D (Tab. 1) and MSRC-12 (Tab. 2), it appears that an ensemble of networks, each trained on a different representation of the skeleton sequences, can significantly outperform the single models. This concept can easily be extended to our approach, *e.g.* by feeding a different permutation of the joints to each network in the ensemble. To explore this idea, we train five independent instances of DM-3DCNN, using the default joint ordering and four additional permutations[2], and average their output probabilities to obtain the final predictions. The results are reported in Tab.2 in the "DM-3DCNN ensemble of 5" row. Using our ensemble we are able to fill the performance gap with the method of Liu *et al.* [17], while still using ≈ 200 times fewer parameters. Note, however, that in our case the relative improvement going from the single model to the ensemble is smaller than in [17], further validating the effectiveness of our network.

SBU Interaction

In our final comparison with state of the art methods, we consider the interaction-focused SBU dataset. Table 3 reports the results obtained with DM-3DCNN and four RNN-based approaches: the hierarchical recurrent network of [5], the co-occurrence LSTM of [38] and the spatio-temporal LSTM of [16]. We also include in the comparison a plain LSTM model as reported in [38]. DM-3DCNN shows the highest accuracy, surpassing ST-LSTM by 0.4 %, a considerably lower advantage when compared to that obtained in the NTU RGB-D dataset. A possible explanation of this difference lies in the relative size of the two datasets. In fact, SBU contains about 200 times less sequences than NTU RGB-D, suggesting that our DM-3DCNN can be more effective at exploiting large-scale datasets compared to the LSTM-based approach in [38].

4.4 In-depth Analysis of DM-3DCNN

As noted in Sec.3.1, the joint permutation considered when forming the EDMs to be fed to the network can have a substantial impact on classification accuracy. To compensate for this, we propose to calculate the EDMs on a learned linear combination ξ_Π of the joints, which encompasses all possible permutations as special cases. In order to validate this approach, we perform an ablation study on the NTU RGB-D dataset and collect the results in Fig.3. In particular, we consider variations of DM-3DCNN trained with EDMs computed on the original joints (Original) or their transformation with ξ_Π (Original + ξ_Π). Learning ξ_Π produces an observable increase in accuracy, both in the cross-subject and cross-view settings.

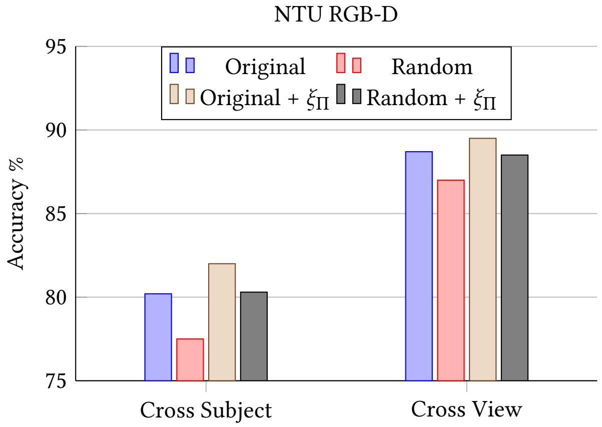

Figure 3: Ablation study on the NTU RGB-D dataset, comparing four different settings of DM-3DCNN: without joint transformation, using the original permutation (Original) or a random one (Random); with joint transformation, using the original permutation (Original + ξ_Π) or a random one (Random + ξ_Π).

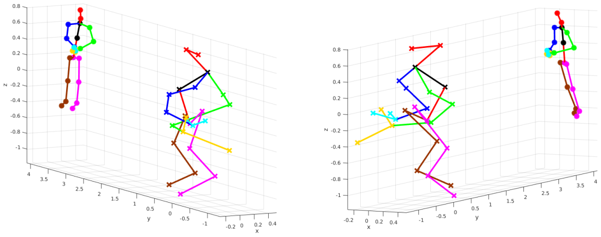

Figure 4: Original skeleton points from NTU RGB-D (circular markers) end transformed points using ξ_Π (cross markers). Two views are shown to better convey the 3D shape of the data. (Image best viewed on screen)

While ξ_Π is in principle able to produce any permutation, we still expect the initial ordering of the input joints to play a role, as we are learning Π by minimizing an highly non-convex function. To test this effect, we re-run the experiments above, this time considering a different, randomly selected permutation of the joints instead of the original one given in the dataset. The resulting accuracies, visualized in Fig. 3 as Random and Random + ξ_Π, are noticeably lower than those obtained with Original and Original + ξ_Π. This can be easily explained by observing that the original permutation is not random, but instead (loosely) follows the structure of the skeleton, keeping joints from distinct body parts close together and thus, intuitively, producing more informative local structures in the EDMs. Interestingly, however, when learning ξ_Π our network

[2] The permutations are selected by separating the joints in six subsets corresponding to left/right arm, left/right leg, head and torso and randomly shuffling the subsets while keeping the order of the joints in each subset fixed.

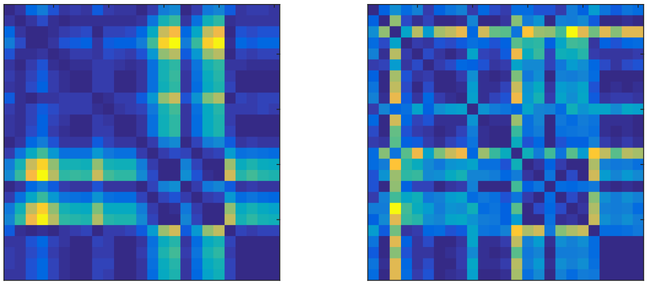

Figure 5: Distance matrices corresponding to the points in Fig.4. Left: original points. Right: transformed points.

is able to overcome the disadvantage imposed by the random shuffling and reach the same accuracy obtained with the hand-picked permutation.

Since we are not applying any constraint on the matrix Π, in general ξ_Π will transform the skeletons in complex, if useful, ways. In order to gain some more insights about the action of ξ_Π on the points, in Fig.4 we plot an example of original and transformed skeletons from NTU RGB-D. Two main phenomena are immediately apparent: i) the transformed points are shifted towards the origin of the coordinates system; ii) the limbs appear to be stretched, while the torso becomes comparatively more compressed. (ii) can be explained as the network giving more importance to the joints in the arm and legs, which, intuitively, can be more discriminative for the task of recognizing actions. For another perspective on the effect of ξ_Π, in Fig.5 we plot the distance matrices corresponding to the points in Fig.4. Here, the patterns visible in the EDM of the transformed points appear to be more contrasted than those in the original one, with stronger edges and corners.

5 CONCLUSION

In this work we presented a novel DNN architecture for recognizing human actions from 3D skeletal data. Our approach is based on two main ideas: (i) representing sequences of skeletons as sequences of euclidean distance matrix (EDM) over a learned transformation of the skeletons' joints; (ii) processing these sequences using a 3D convolutional neural network. We validated our method on three benchmark datasets, obtaining state of the art results. In particular, on the large-scale NTU RGB-D dataset we achieved an improvement in accuracy of $\approx 2\,\%$ over the previous state of the art approach, while using almost 1000 times fewer parameters and operations. A promising direction for future research involves unifying skeleton detection from RGB-D video and action recognition in a single, end-to-end differentiable architecture, *e.g.* by combining our approach with those in [3] and [19].

A NETWORK ARCHITECTURE DETAILS

Following the terminology employed in [33], our network is composed of three stages of 3 ResNeXt blocks each. Each block is obtained from the bottleneck template $\begin{bmatrix} 1\times1\times1, 32 \\ 3\times3\times3, 32 \\ 1\times1\times1, 128 \end{bmatrix}$, with cardinality $C = 16$ and pre-activation structure [8]. The network starts with a $3\times3\times3$ convolution with 32 filters. In the second and third block we halve the spatial resolution of the feature maps by applying a stride of $2\times2\times2$ on the first ResNeXt block. Correspondingly, we increase the depth of the feature maps by a factor 2. The third stage is followed by global 3D average pooling and a fully connected layer producing the final prediction.

ACKNOWLEDGMENTS

This work has been supported by the Spanish Ministry of Economy and Competitiveness through the ROBINSTRUCT project (TIN2014-58178-R) and the María de Maeztu grant to IRI (MDM-2016-0656). Additionally, we gratefully acknowledge financial support from the DIGIMAP project, funded under grant #860375 by the Austrian Research Promotion Agency (FFG). Finally, we thank Nvidia for the hardware donation under the GPU grant program.

REFERENCES

[1] 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. (2015). http://tensorflow.org/ Software available from tensorflow.org.

[2] Hakan Bilen, Basura Fernando, Efstratios Gavves, Andrea Vedaldi, and Stephen Gould. 2016. Dynamic image networks for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 3034–3042.

[3] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2016. Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. *arXiv preprint arXiv:1611.08050* (2016).

[4] Yong Du, Yun Fu, and Liang Wang. 2015. Skeleton based action recognition with convolutional neural network. In *Proceedings of the IAPR Asian Conference on Pattern Recognition (ACPR)*. IEEE, 579–583.

[5] Yong Du, Wei Wang, and Liang Wang. 2015. Hierarchical recurrent neural network for skeleton based action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1110–1118.

[6] Simon Fothergill, Helena Mentis, Pushmeet Kohli, and Sebastian Nowozin. 2012. Instructing people for training gestural interactive systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1737–1746.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 1026–1034.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Identity mappings in deep residual networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 630–645.

[9] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[10] Farzad Husain, Babette Dellen, and Carme Torras. 2016. Action recognition based on efficient deep feature learning in the spatio-temporal domain. *IEEE Robotics and Automation Letters* 1, 2 (2016), 984–991.

[11] Mohamed E Hussein, Marwan Torki, Mohammad Abdelaziz Gowayyed, and Motaz El-Saban. 2013. Human Action Recognition Using a Temporal Hierarchy of Covariance Descriptors on 3D Joint Locations. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, Vol. 13. 2466–2472.

[12] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 2013. 3D convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence* 35, 1 (2013), 221–231.

[13] Imran N Junejo, Emilie Dexter, Ivan Laptev, and Patrick Perez. 2011. View-independent action recognition from temporal self-similarities. *IEEE transactions on pattern analysis and machine intelligence* 33, 1 (2011), 172–185.

[14] Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[15] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Proceedings of the Neural Information Processing Systems Conference (NIPS)*. 1097–1105.

[16] Jun Liu, Amir Shahroudy, Dong Xu, and Gang Wang. 2016. Spatio-temporal LSTM with trust gates for 3D human action recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 816–833.

[17] Mengyuan Liu, Hong Liu, and Chen Chen. 2017. Enhanced skeleton visualization for view invariant human action recognition. *Pattern Recognition* (2017).

[18] Jiajia Luo, Wei Wang, and Hairong Qi. 2013. Group sparsity and geometry constrained dictionary learning for action recognition from depth maps. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 1809–1816.

[19] Francesc Moreno-Noguer. 2017. 3D Human Pose Estimation from a Single Image via Distance Matrix Regression. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017).

[20] Eshed Ohn-Bar and Mohan Trivedi. 2013. Joint angles similarities and HOG2 for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 465–470.

[21] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)* 115, 3 (2015), 211–252.

[22] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. 2016. NTU RGB+ D: A large scale dataset for 3D human activity analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1010–1019.

[23] Karen Simonyan and Andrew Zisserman. 2014. Two-stream convolutional networks for action recognition in videos. In *Proceedings of the Neural Information Processing Systems Conference (NIPS)*. 568–576.

[24] Sijie Song, Cuiling Lan, Junliang Xing, Wenjun Zeng, and Jiaying Liu. 2016. An End-to-End Spatio-Temporal Attention Model for Human Action Recognition from Skeleton Data. *arXiv preprint arXiv:1611.06067* (2016).

[25] Jaeyong Sung, Colin Ponce, Bart Selman, and Ashutosh Saxena. 2012. Unstructured human activity detection from rgbd images. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 842–849.

[26] Vivek Veeriah, Naifan Zhuang, and Guo-Jun Qi. 2015. Differential recurrent neural networks for action recognition. In *Proceedings of the International Conference on Computer Vision (ICCV)*. 4041–4049.

[27] Raviteja Vemulapalli, Felipe Arrate, and Rama Chellappa. 2014. Human action recognition by representing 3d skeletons as points in a lie group. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 588–595.

[28] Jiang Wang, Xiaohan Nie, Yin Xia, Ying Wu, and Song-Chun Zhu. 2014. Cross-view action modeling, learning and recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2649–2656.

[29] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. 2016. Temporal segment networks: towards good practices for deep action recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 20–36.

[30] Pichao Wang, Wanqing Li, Zhimin Gao, Yuyao Zhang, Chang Tang, and Philip Ogunbona. 2017. Scene flow to action map: A new representation for rgb-d based action recognition with convolutional neural networks. *arXiv preprint arXiv:1702.08652* (2017).

[31] Pichao Wang, Zhaoyang Li, Yonghong Hou, and Wanqing Li. 2016. Action recognition based on joint trajectory maps using convolutional neural networks. In *Proceedings of the ACM Conference on Multimedia (ACM'MM)*. ACM, 102–106.

[32] Lu Xia, Chia-Chih Chen, and JK Aggarwal. 2012. View invariant human action recognition using histograms of 3d joints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, 20–27.

[33] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. 2016. Aggregated residual transformations for deep neural networks. *arXiv preprint arXiv:1611.05431* (2016).

[34] Yanhua Yang, Cheng Deng, Shangqian Gao, Wei Liu, Dapeng Tao, and Xinbo Gao. 2017. Discriminative Multi-instance Multitask Learning for 3D Action Recognition. *IEEE Transactions on Multimedia* 19, 3 (2017), 519–529.

[35] Kiwon Yun, Jean Honorio, Debaleena Chattopadhyay, Tamara L Berg, and Dimitris Samaras. 2012. Two-person interaction detection using body-pose features and multiple instance learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, 28–35.

[36] Songyang Zhang, Xiaoming Liu, and Jun Xiao. 2017. On Geometric Features for Skeleton-Based Action Recognition using Multilayer LSTM Networks. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*.

[37] Lijuan Zhou, Wanqing Li, Yuyao Zhang, Philip Ogunbona, Duc Thanh Nguyen, and Hanling Zhang. 2014. Discriminative key pose extraction using extended lc-ksvd for action recognition. In *Proceedings of the International Conference on Digital Image Computing: Techniques and Applications (DICTA)*. IEEE, 1–8.

[38] Wentao Zhu, Cuiling Lan, Junliang Xing, Wenjun Zeng, Yanghao Li, Li Shen, and Xiaohui Xie. 2016. Co-occurrence feature learning for skeleton based action recognition using regularized deep lstm networks. *arXiv preprint arXiv:1603.07772* (2016).