Skepxels: Spatio-temporal Image Representation of Human Skeleton Joints for Action Recognition

Jian Liu Naveed Akhtar Ajmal Mian
School of Computer Science and Software Engineering
The University of Western Australia
jian.liu@research.uwa.edu.au, naveed.akhtar@uwa.edu.au, ajmal.mian@uwa.edu.au

Abstract

Existing human joint representations do not fully exploit the learning power of Convolutional Neural Networks (CNNs). We propose a representation for skeleton joint sequences that is both spatial and spatio-temporal with respect to the receptive fields of convolution kernels of CNN to facilitate learning from spacial locations of the joints as well as their transitions over time. Our representation allows for better hierarchical learning by CNNs as we transform skeleton sequences into images of flexible dimensions encoding rich spatial and spatio-temporal information about the joints by maximizing a unique distance metric, defined collaboratively over the distinct joint arrangements. Our representation additionally encodes the relative joint velocities. The proposed action recognition exploits the representation in a hierarchical manner by first capturing the micro-temporal relations between the skeleton joints using CNN and then exploiting their macro-temporal relations by computing the Fourier Temporal Pyramids. We extend the Inception-ResNet CNN architecture with the proposed method and improve the state-of-the-art accuracy by 4.4% *on the large scale NTU human activity dataset. On NUCLA and UTD-MHAD datasets, our method outperforms the existing results by* 5.7% *and* 9.3% *respectively.*

1. Introduction

Human action recognition has applications in smart surveillance and human-computer interaction etc. Extracting human skeleton joints from videos to perform this task is a popular choice because it removes irrelevant information such as clothing texture, illumination conditions and background [4, 51, 27, 10, 50, 42, 6, 43, 44], and recent methods can extract skeleton data in real-time from single view RGB videos [25]. Convolutional Neural Networks (CNNs) [20, 38, 8] can learn powerful hierarchical representations from raw images [47, 30, 16, 40] by exploiting correlations between local pixels, which is the key to accurate image classification. We envisage that higher human action recognition accuracy can be achieved analogously by capitalizing on local correlations between the skeleton joints. This is possible by arranging the skeleton joints in images and allowing CNNs to be directly trained on such images. However, the low number of joints and the inherent dissimilarity between skeletons and images restrict the use of CNNs for processing skeleton data. A major motivation behind this work is to fully exploit the perpetual advances in CNNs for skeletal action data. We demonstrate that CNNs can lead to state-of-the-art action recognition performance using skeletal time-series data alone under the proposed representation.

Previous attempts [17, 5] of using CNNs arrange the skeleton joints in an image column, paying no attention to the different possibilities of joint arrangements. Since the number of joint is $\sim$25, these methods find it inevitable to up-sample the resulting images by over eight folds to match the input size of pre-trained CNNs. Consequently, the convolution kernels at the first few CNN layers have a receptive field over one joint or a linear combination of only two joints. The covolution kernels can neither learn correlations between multiple different joint combinations nor learn spatial only features. These ill-defined semantics severely limit what the employed CNN can learn.

We propose an atomic visual unit *Skepxel* - skeleton picture element or skeleton pixel, to construct skeletal images of flexible dimensions that can be directly processed by modern CNN architectures without any re-sampling. Skepxels are constructed by organizing a set of distinct skeleton joint arrangements from multiple frames into a single tensor. The set is chosen under a unique distance metric that is collectively defined over the joint arrangements for each frame. Unlike previous works where skeleton joints of a frame were arranged in a column, we arrange them in a 2D grid to take full advantage of the 2D kernels in CNNs. The temporal evolution of the joints is captured by employing Skepxels from multiple frames into one image forming a compact representation of rich spatial only (poses) and spatio-temporal information about the action. Using 5×5 Skepxels, many 3×3 kernels at the first two CNN layers have receptive fields focused on individual Skepxels to learn

spatial only features which is unique to our representation. Kernels at all layers learn spatio-temporal features similar to other representations. Owing to the systematic construction of the skeletal images, it is possible to encode multiple semantic notions about the joints in an image, by encoding "location" and "velocity" of the joints.

We also contribute a framework that uses the proposed representation for human action recognition. To that end, we hierarchically capture the micro-temporal relations between the joints in the frames using Skepxels and exploit the macro-temporal relations between the frames by computing the Fourier Temporal Pyramids [45] of the CNN features of the skeletal images. We demonstrate the use of skeletal images of different sizes with the Inception-ResNet [37]. Moreover, we also enhance the network architecture for the proposed framework. The proposed technique is thoroughly evaluated using the NTU Human Activity Dataset [33], Northwestern-UCLA Multiview Dataset [46] and UTD Multimodal Human Action Dataset [2]. Our approach improves the state-of-the-art performance on the large scale dataset [33] by 4.4%, whereas the accuracy gain on the remaining two datasets is 5.7% and 9.3%.

2. Related Work

With the easy availability of reliable human skeleton data from RGB-D sensors, the use of skeleton information in human action recognition is becoming very popular. Skeleton based action analysis is becoming even more promising because of the possibility of extracting skeleton data in real time using a single RGB camera [25]. Skeleton data can be directly used to recognize human actions. For instance, Devanne *et al*. [4] represented the 3-D coordinates of skeleton joints and their change over time as trajectories, and formulated the action recognition problem as computing the similarity between the shape of trajectories in a Riemannian manifold. The joint trajectories model the temporal dynamics of actions, and remain invariant to geometric transformation. Yang *et al*. [51] proposed a mid-level granularity of joints called skelets, which can be used to describe the intrinsic interdependencies between skeleton joints and action classes. To balance the skelet-wise and action-wise relevance, a joint structured sparsity inducing regularization is also integrated into their framework.

Skeleton information is also commonly used in guiding the action representation in other image and video modalities. Cao *et al*. [1] used extracted body joints to guide the selection of convolutional layer activations of RGB input action videos. They pooled the activations of 3-D convolutional feature maps according to the position of body joints, and thus created discriminative spatio-temporal video descriptors for action recognition. To facilitate end-to-end training, they proposed a two-stream framework with bilinear pooling, with one stream extracting visual features and the other locating key-points of the features maps.

Zanfir *et al*. [52] proposed a moving pose descriptor which considers both pose information and the differential quantities of the skeleton joints for human action recognition. Their approach is non-parametric and therefore can be used with small amount of training data or even with one-shot training. Du *et al*. [5] transformed the skeleton sequences into images by concatenating the joint coordinates as vectors and arranged these vectors in a chronological order as columns of an image. The generated images are resized and passed through a series of adaptive filter banks. Their approach is based on global spatial and temporal information and does not exploit the local correlation of joints in skeletons. In contrast, our approach models the global and local temporal variations simultaneously.

Veeriah *et al*. [41] proposed to use a differential Recurrent Neural Network (dRNN) to learn the salient spatio-temporal structure in a skeleton action. They used the notion of "Derivative of States" to quantify the information gain caused by the salient motions between the successive frames, which guides the dRNN to gate the information that is memorized through time. Their method relies on concatenating 5 types of hand-crafted skeleton features to train the proposed network. Similarly, Du *et al*. [6] applied a hierarchical RNN to model skeleton actions. They divided the human skeleton into five parts according to human physical structure. Each part is fed into a bi-directional RNN and the outputs are hierarchically fused for the higher layers. One potential limitation of this approach is that the definition of body part is dataset-specific, which causes extra preprocessing when applied to different action datasets.

Shahroudy *et al*. [35] also used the division of body parts and proposed a multimodal-multipart learning method to represent the dynamics and appearance of body. They selected the discriminative body parts by integrating a part selection process into their learning framework. In addition to the skeleton based features, they also used hand-crafted features for depth modality, such as LOP (local occupancy patterns) and local HON4D (histogram of oriented 4D normals) around each body joint. Vemulapalli and Chellappa [43] represented skeletons using the relative 3D rotations between various body parts. They applied concept of rolling maps to model skeletons as points in the Lie group, and then modeled human actions as curves in the Lie group.

Based on the intuition that the traditional Lie group features may be too shallow to learn a robust recognition algorithm for skeleton data, Huang *et al*. [13] incorporated the Lie group structure into deep learning, to transform the high-dimensional Lie group trajectory into temporally aligned Lie group features for skeleton-based action recognition. Their learning structure (LieNet) generalizes the traditional neural network model to non-Euclidean Lie groups. One issue with LieNet is that it is mainly designed to learn spatial features of skeleton data, and does not take full ad-

vantage of the rich temporal cues of human actions. To leverage both spatial and temporal information in skeleton sequences, Kerola *et al*. [18] used a novel graph representation to model skeletons and keypoints as a temporal sequence of graphs, and applied the spectral graph wavelet transform to create the action descriptors.

Ke *et al*. [17] transformed a skeleton sequence into three clips of gray-scale images. Each clip consists of four images, which encode the spatial relationship between the joints by inserting reference joints into the arranged joint chains. They employed the pre-trained VGG19 model to extract image features and applied the temporal mean pooling to represent an action. A Multi-Task Learning was proposed for classification. Wang and Wang [44] proposed a two-stream RNN architecture to simultaneously exploit the spatial relationship of joints and temporal dynamics of the skeleton sequences. In the spatial RNN stream, they used a chain-like sequence and a traversal sequence to model the spatial dependency, which restricts modeling all possibilities of the joint movements.

Kim and Reiter [19] proposed a Res-TCN architecture to learn spatial-temporal representation for skeleton actions. They constructed per-frame inputs to the Res-TCN by flatting 3D coordinates of the joints and concatenating values for all the joints in a skeleton. Their method improves interpretability for skeleton action data, however, it does not effectively leverage the rich spatio-temporal relationships between different body joints. To better represent the structure of skeleton data, Liu *et al*. [23] proposed a tree traversal algorithm to take the adjacency graph of the body joints into account. They processed the joints in top-down and bottom-up directions to keep the contextual information from both the descendants and the ancestors of the joints. Although this traversal algorithm discovers spatial dependency patterns, it has the limitation that the dependency of joints from different tree branches can not be easily modeled.

3. Proposed Approach

Restricted by the small number of joints in a human skeleton, existing approaches for converting the skeleton data into images generally result in smaller size images than what is required for the mainstream CNN architectures e.g. VGG [36], Inception [39], ResNet [9]. Consequently, the images are up-sampled to fit the desired network architectures [5, 17] which imports unnecessary noise in the data. This also compromises the effectiveness of the network kernels that are unable to operate on physically meaningful discrete joints. One potential solution is to design new CNN architectures that are better suited to the smaller images. However, small input image size restricts the receptive fields of the convolution kernels as well as the network depth. As a result, the network may not be able to appropriately model the skeleton data.

In this paper, we address this problem by mapping the skeleton data from a fixed length sequence to an image with the help of a basic building block (similar to pixel). The resulting image is rich in both spatial and temporal information of the skeleton sequences, and can be constructed to match arbitrary input dimensions of the existing network architectures. The approach is explained below.

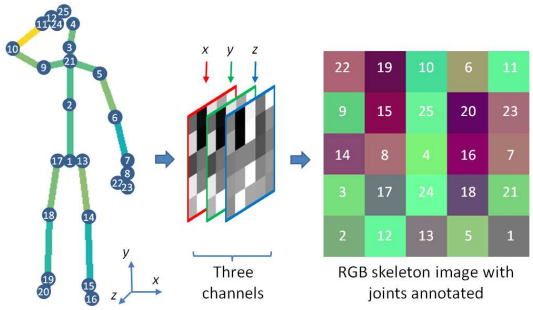

Figure 1. Illustration of a *Skepxel* rendered as an RGB image patch. The numbers on skeleton and color image share the joint description. e.g. 3-neck, 18-right knee, 21-spine, etc.

3.1. Skeleton Picture Elements (Skepxels)

We propose to map a skeleton sequence to an image $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$ with the help of *Skepxels*. A Skepxel is a tensor $\boldsymbol{\psi} \in \mathbb{R}^{h \times w \times 3}$ obtained by arranging the indices of the skeleton joints in a 2D-grid and encoding their coordinate values along the third dimension. We treat the skeleton in a video as a set $\mathcal{S} \subseteq \mathbb{R}^3$ such that its j^{th} element, i.e. $\mathbf{s}_j \in \mathbb{R}^3$ represents the Cartesian coordinates of the j^{th} skeleton joint. Thus, the cardinality of $\mathcal{S}$, i.e. $|\mathcal{S}| \in \mathbb{R}$ denotes the total number of joints in the skeleton. For $\boldsymbol{\psi}$, it entails $h \times w = |\mathcal{S}|$. This formulation allows us to represent a Skepxel as a three-channel image patch, as illustrated in Fig. 1. We eventually construct the image $\mathbf{I}$ by concatenating multiple Skepxels for a skeleton sequence.

Owing to the square shaped kernels of CNN architectures, the skeletal information in images is likely to be processed more effectively for square/near square shaped building blocks of the images. Therefore, our representation constrains the height and the width of the Skepxels to be as similar as possible.

3.2. Compact spatial coding with Skepxels

A Skepxel constructed for a given skeleton frame encodes the spatial locations of the skeleton joints. Considering the convolution operations involved in CNN learning, it is apparent that different arrangements of the joints in a Skepxel can result in a different behavior of the models. This is fortuitous, as we can encode more information in the image $\mathbf{I}$ for the CNNs by constructing it with multiple Skepxels that employ different joint arrangements. However, the image must use only a few (but highly relevant) Skepxels for keeping the representation of the skeleton sequence compact.

Let $\mathcal{A} \subseteq \mathbb{R}^{h \times w}$ be a set of 2D-arrays, with its i^{th} ele-

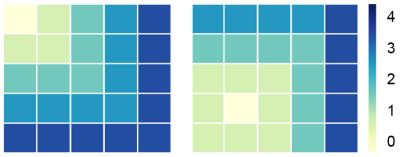

Figure 2. Illustration of the employed definition of the radial distance on 5×5 grids. If the joint α_i is located at [1,1] position in $\mathbf{A}_j^m$, the left 5×5 grid is used. For the joint location [4,2], the right grid is used. There are 25 such grids in total to measure the distance of skeleton joints among m arrangements.

ment $\mathbf{A}_i \in \mathbb{R}^{h \times w}$ representing the i^{th} possible arrangement of the skeleton joints for a Skepxel. The cardinality of this set can be given as $|\mathcal{A}| = (h \times w)!$. Even for a video containing only a 25-joint skeleton, the total number of possible arrangements of the joints for a Skepxel is $\sim 1.55 \times 10^{25}$. Assume that we wish to use only m Skepxels in $\mathbf{I}$ for the sake of compactness, we must then select the joint arrangements for those Skepxels from a possible ${}^{|\mathcal{A}|}C_m$ combinations, which becomes a prohibitively large number for the practical cases (e.g. ${}^{(4\times4)!}C_{16} > 10^{199}$). Therefore, a principled approach is required to choose the suitable arrangements of the joints to form the desired Skepxels.

To select the m arrangements for the same number of Skepxels, we define a metric $\Delta(\mathcal{A}^m) \rightarrow \gamma$ over an arbitrary subset $\mathcal{A}^m$ of $\mathcal{A}$, where $|\mathcal{A}^m| = m$, such that

$$\Delta(\mathcal{A}^m) = \sum_{j=1}^{|\mathcal{A}^m|} \sum_{i=1}^{|\mathcal{S}|} \delta(\alpha_i, \mathbf{A}_j^m). \tag{1}$$

In Eq. (1), $\mathbf{A}_j^m$ denotes the j^{th} element of $\mathcal{A}^m$ and α_i is the i^{th} element of the set $\{1, 2, ..., |\mathcal{S}|\}$. The function $\delta(.,.)$ computes the cumulative radial distance between the location of the joint α_i in $\mathbf{A}_j^m$ and its locations in the remaining elements of $\mathcal{A}^m$. Let (x, y) denote the indices of α_i in $\mathbf{A}_j^m$, and (x_q, y_q) denote its indices in any other $\mathbf{A}_q^m \in \mathcal{A}^m$, then $\delta(\alpha_i, \mathbf{A}_j^m) = \sum_{q \neq j, q=1}^{|\mathcal{A}^m|-1} \max(abs(x - x_q), abs(y - y_q))$, where $abs(.)$ computes the absolute value. As per the definition of $\Delta(.)$, γ is a distance metric defined over a set of m possible arrangements of the skeleton joints such that a higher value of γ implies a better scattering of the joints in the considered m arrangements. The notion of the radial distance used in Eq. (1) is illustrated in Fig. 2. Noticing the image patterns in the figure, we can see the relevance of this metric for the CNNs that employ square shaped kernels, as compared to the other metrics, e.g. Manhattan distance.

Due to better scattering, the skeleton joint arrangements with the larger γ values are generally preferred by the CNN architectures to achieve higher accuracy. Moreover, different sets of arrangements with similar γ values were found to achieve similar accuracies. Interestingly, this implies that for the CNNs the relative positions of the joints in the Skepxels become more important as compared to their absolute positions. This observation preempts us to construct

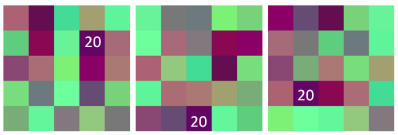

Figure 3. *Skepxels* generated for one skeleton frame. Same color corresponds to the same joint. Only joint 20 is marked.

Skepxels with the skeleton joint arrangements based on the semantics of the joints. On the other hand, selection of the best set of arrangements from the ${}^{|\mathcal{A}|}C_m$ possibilities is an NP-hard problem for all practical cases.

We devise a pragmatic strategy to find a suitable set of the skeleton joint arrangements for the desired m Skepxels. That is, we empirically choose a threshold γ_t for the Skepxels and generate m matrices in $\mathbb{R}^{h \times w}$ such that the coefficients of the matrices are sampled uniformly at random in the range $[1, h \times w]$, without replacement. We consider these matrices as the elements of $\mathcal{A}^m$ if their γ value is larger than γ_t. We use the resulting $\mathcal{A}^m$ to construct the m Skepxels. The Skepxels thus created encode a largely varied skeleton joint arrangements in a compact manner. Fig. 3, illustrates three Skepxels created by the proposed scheme for a single skeleton frame containing 25 joints. The Skepxels are shown as RGB image patches. In our approach, we let $m = H/h$ and construct a tensor $\mathbf{\Psi} \in \mathbb{R}^{H \times w \times 3}$ by the row-concatenation of the Skepxels $\boldsymbol{\psi}_{i \in \{1,2,...,m\}}$. The constructed tensor $\mathbf{\Psi}$ is rich in the spatial information of the joints in a single frame of the video.

3.3. Compact temporal coding with Skepxels

To account for the temporal dimension in a sequence of the skeleton frames, we compute the tensor $\mathbf{\Psi}_i$ for the i^{th} frame in the n-frame sequence and concatenate those tensors in a column-wise manner to construct the desired image $\mathbf{I}$. For a sequence of frames, the appearance of a Skepxel changes specifically at the locations of the active joints for the action - indicating effective encoding of the action dynamics by Skepxels. The concatenation of $\mathbf{\Psi}_{i \in \{1,2,...,n\}}$ ensures that the dynamics are recorded in $\mathbf{I}$ under m suitable Skepxels, making the representation spatially and temporally rich. The formation of the final image by concatenating $\mathbf{\Psi}_i, \forall i$ is illustrated in Fig. 4.

Different action videos may contain various number of skeletal frames. For the videos that comprise the skeleton sequences with more than n frames, we create multiple images from the same video and label them according to the action label. For the videos with fewer than n frames, we found that the simple strategy of interpolating between the frames works well to construct the image of the desired size. Note that, the images resulting from the proposed method capture the temporal dynamics in the raw skeleton data. By fixing the length of the temporal window to n, the images are able to encode the micro-temporal movements that are expected to model the fine motion patterns contributing to

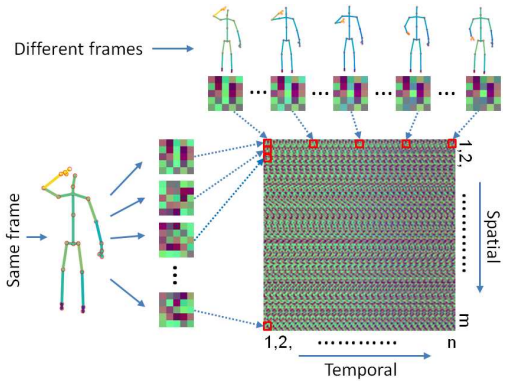

Figure 4. The final image is compactly constructed with the *Skepxels* along spatial and temporal dimensions.

the classification of the entire action video. In Section 3.7, we also discuss the exploitation of the macro-temporal relationships with the proposed representation.

3.4. Modeling joint speed with Skepxels

The modular approach to construct images with the Skepxels not only allows us to easily match the input dimensions of an existing CNN architecture, it also provides the flexibility to encode a notion that is semantically different than the "locations" of the skeleton joints. We exploit this fact to extend our representation to the skeleton joint "speeds" in the frame sequences. To that end, we construct the Skepxels similar to the procedure described above, however instead of using the Cartesian coordinate values for the joints we use the differences of these values for the same joints in the consecutive frames. A Skepxel thus created encodes the speeds of the joint movements, where the time unit is governed by the video frame-rate. We refer to the final tensors constructed with the joint coordinates as the *location* images, and the tensors constructed using the joint speeds as the *velocity* images.

For many actions, the speed variations among different skeleton joints is an important cue for distinguishing between them (e.g. `walking` and `running`), and it is almost always supplementary to the information encoded in the absolute locations of the joints. Therefore, in our representation, we augment the final image by appending the three speed channels dx, dy, dz to the three location channels x, y, z. This augmentation is illustrated in Fig. 5. We note that unless allowed by the CNN architecture under consideration, the augmentation with the speed channels is not mandatory in our representation. Nevertheless, it is desirable for better action recognition accuracy, which will become evident from our experiments in Section 5.

3.5. Normalization and data augmentation

Before converting the skeleton data into images using the proposed method, we perform the normalization of the raw skeleton data. To do so, we anchor the `hip` joint in a skeleton to the origin of the used Cartesian coordinates, and

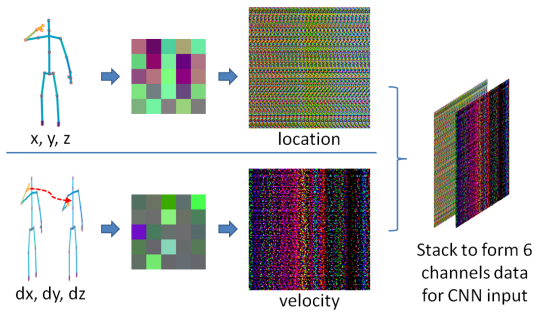

Figure 5. Joint differences between consecutive frames give *velocity* images, which are appended to the *location* images.

align the virtual vector between the `left-shoulder` and the `right-shoulder` of the skeleton to the x-axis of the coordinate system. This normalization strategy also results in mitigating the translation and viewpoint variation effects in the skeleton data by filtering out the motion-irrelevant noises. A further normalization is performed over the channels of the resulting images to restrict the values of the pixels in the range $[0, 255]$. Both types of normalizations are carried out on the training as well as the testing data.

In order to augment the data, we make use of the additive Gaussian noise. We draw samples from the zero Mean Gaussian distribution with 0.02 Standard Deviation and add those samples to the skeleton joints in the frame sequences to double the training data size. This augmentation strategy is based on the observation that slight variations in the joint locations/speeds generally do not vary the skeletal information significantly enough to change the label of the associated action. For our experiments, doubling the training data size already resulted in a significant performance gain over the existing approaches. Therefore, no further data augmentation was deemed necessary for the experiments.

3.6. Processing skeletal images with CNNs

Due to its flexibility, the proposed mapping of the skeletal information to the image-like tensors allows us to exploit a wide variety of existing (and potentially future) CNN architectures to effectively process the information in the skeleton frame sequences. To demonstrate this, we employ the Inception-ResNet [37] as the test bed for our representation. This recent CNN architecture has been successful in the general image classification task [3], as well as the specific tasks such as face recognition [32]. More importantly, the architecture allows for a variable input image size both in terms of the spatial dimensions and the number of color channels of the image.

First, we trained the Inception-ResNet from scratch by constructing the skeletal images of different dimensions (without the speed channel augmentation). This training resulted in a competitive performance of the network for a variety of image sizes - details provided in Section 5. We strictly followed the original work [37] for the training methodology, which demonstrates the compatibility of

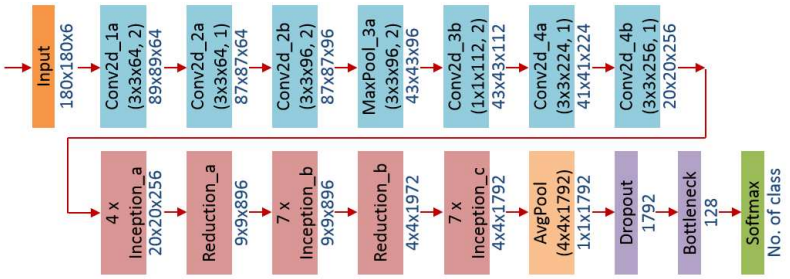

Figure 6. Modified architecture of Inception-ResNet [37]: The "STEM" part of the network is extended to fit the augmented 6-channel input images. The input/output sizes are described as $rows \times cols \times channels$. The kernel is specified as $rows \times cols \times filters, stride$.

the proposed representation with the existing frameworks. In our experiments, training the network from scratch was consistently found to be more effective than fine tuning the existing models. We conjecture that the visible difference of the patterns in the skeleton images and the images of the natural scenes is the main reason for this phenomenon. Hence, it is recommended to train the network from scratch for the full exploitation of the proposed representation.

To demonstrate the additional benefits of augmenting the skeletal image with the speeds of the skeleton joints, we also trained the Inception-ResNet for the augmented images. Recall, in that case the resulting image has six channels - three channels each for the joint locations and the joint speeds. To account for the additional information, we modified the Inception-ResNet by extending the "STEM" part of the network[37]. The modified architecture is summarized in Fig. 6. To train the modified network, Center loss [49] is added to the cross entropy to form the final loss function. We optimized the network with the $RMSProp$ optimizer, and selected the initial learning rate as 0.1.

3.7. Macro-temporal encoding and classification

Once it is possible to process the skeleton data with the desired CNN, it also becomes practicable to exploit the CNN features to further process the skeletal information. For instance, as noted in Section 3.3, a single skeleton image used in this work represents the temporal information for only n skeletal frames, which encodes the micro-temporal patterns in an action. To explore the long term temporal relationship of the skeleton joints, we can further perform a macro-temporal encoding over the CNN features. We perform this encoding as follows.

Given a skeleton action video, we first construct the 'Q' possible skeleton images for the video. These images are forward passed through the network and the features $\boldsymbol{\xi}_{i\in\{1,2,\ldots,Q\}} \in \mathbb{R}^{1792}$ from the `prelogit` layer of the Inception-Resnet are extracted. We compute the Short Fourier Transform [26] over $\boldsymbol{\xi}_i, \forall i$ and retain 'z' low frequency components of the computed transform. Next, the column vectors $\boldsymbol{\xi}_i$ are divided into two equal segments along their row-dimension, and the Fourier Transform is again applied to retain another set of 'z' low frequency components for each segment. The procedure is repeated 'ℓ' times and all the $2^{\ell-1} \times z$ resulting components are concatenated to represent the video. These features are used for training an SVM classifier. We used $\ell = 3$ in our experiments in Section 5. The features computed with the above method take into account the whole skeletal sequence in the videos, thereby accounting for the macro-temporal relations between the skeleton joints.

4. Datasets

NTU RGB+D Dataset: The NTU RGB+D Human Activity Dataset [33] is a large-scale RGB+D dataset for human activity analysis. This dataset has been collected with the Kinect v2 sensor and it includes 56,880 action samples each for RGB, depth, skeleton and infra-red videos. Since we are concerned with the skeleton sequences only, we use the skeleton part of the dataset to evaluate our method. In the dataset, there are 40 human subjects performing 60 types of actions including 50 single person actions and 10 two-person interactions. Three sensors were used to capture the data simultaneously from three horizontal angles: $-45°, 0°, 45°$, and every action performer performed the action twice, facing the left or right sensor respectively. Moreover, the height of the sensors and their distances to the action performer have been adjusted in the dataset to get further viewpoint variations. The NTU RGB+D dataset is one of the largest and the most complex cross-view action dataset of its kind to date. We followed the standard evaluation protocol proposed in [33], which includes cross-subject and cross-view evaluations. For the cross-subject case, 40 subjects are equally split into training and testing groups. For the cross-view protocol, the videos captured by the sensor C-2 and C-3 are used as the training samples, whereas the videos captured by the sensor C-1 are used for testing.

Northwestern-UCLA Dataset: This dataset [46] contains RGB, Depth and skeleton videos captured simultaneously from three different viewpoints with the Kinect v1 sensor, while we only use skeleton data in our experiments. The dataset contains videos of 10 subjects performing 10 actions: (1) pick up with one hand, (2) pick up with two hands, (3) drop trash, (4) walk around, (5) sit down, (6) stand up, (7) donning, (8) doffing, (9) throw, and (10) carry. The three viewpoints are: (a) left, (b) front, and (c) right. This dataset

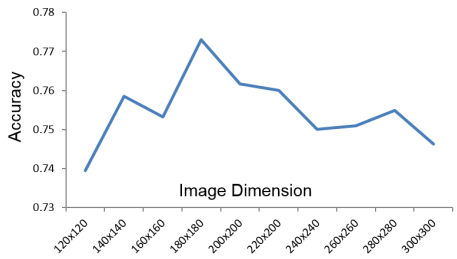

Figure 7. Action recognition performance for different skeletal image size on the NTU RGB+D Human Activity Dataset [33].

is challenging because some videos share the same "walking" pattern before and after the actual action is performed. Moreover, some actions such as "pick up with on hand" and "pick up with two hands" are hard to distinguish from different viewpoints. We use skeleton videos captured from two views for training and the third view for testing, which produces three possible cross-view combinations.

UTD Multimodal Action Dataset: The UTD-MHAD dataset [2] consists of 27 actions performed by 8 subjects. Each subject repeated the action 4 times, resulting in 861 action sequences in total. The RGB, depth, skeleton and the inertial sensor signals were recorded. We only use skeleton videos in our experiments. We follow [2] to evaluation UTD-MHAD dataset with cross-subject protocol, which means the data from subject 1, 3, 5, 7 is used for training, and the data form subject 2, 4, 6, 8 is used for testing.

5. Experiments

Skeleton Image Dimension: We first analyze the performance of the models trained with different sizes of the skeleton images to choose a suitable image size for our experiments. We used the NTU RGB+D Human Activity Dataset [33] for this purpose. According to the evaluation protocol of [33], we split the training samples into training and validation subset. Only the *location* images were evaluated. After the best image size was chosen, we applied it to both *location* and *velocity* images, and conducted the comprehensive experiments. During our evaluation for the image size selection, we increased the image size from 120×120 to 300×300, with a step of 20 pixels. Fig. 7 shows the recognition accuracy for each setting. We eventually selected 180×180 as the image dimensions based on these results. These skeletal image dimensions were kept the same in our experiments with the other data sets as well.

Evaluation on NTU RGB+D Dataset: We trained our CNN model from scratch for the NTU RGB+D dataset. The model was trained twice for cross-subject and cross-view evaluations respectively. We first evaluated the proposed method with the *location* images only, where the input tensor to the network was in $\mathbb{R}^{H\times W\times 3}$. We call this evaluation as Skepxel$_{\text{loc}}$ mode. Then, we evaluated our method in Skepxel$_{\text{loc+vel}}$ mode, where we combined the *location* and *velocity* images to train the network with the input tensors in $\mathbb{R}^{H\times W\times 6}$. We used the network defined in Fig. 6 for our evaluation in the Skepxel$_{\text{loc+vel}}$ mode. Note that some action clips were performed by two persons. In this case we encode each skeleton individually in alternating frames to form a skepxel-based image. This also enable us to readily use the normalization method describe in Section 3.5. Table 1 compares the performance of our approach with the existing techniques on the NTU dataset. Our method is able to improve the accuracy by 4.4% in the Skepxel$_{\text{loc+vel}}$ mode over the nearest competitor.

Table 1. Action recognition accuracy (%) on the NTU Dataset.

Method	Data	Cross Subject	Cross View
Baseline			
Lie Group [42]	Joints	50.1	52.8
Deep RNN [33]	Joints	56.3	64.1
HBRNN-L [6]	Joints	59.1	64.0
Dynamic Skeleton [12]	Joints	60.2	65.2
Deep LSTM [33]	Joints	60.7	67.3
LieNet [13]	Joints	61.4	67.0
P-LSTM [33]	Joints	62.9	70.3
LTMD [24]	Depth	66.2	-
ST-LSTM [23]	Joints	69.2	77.7
DSSCA-SSLM [34]	RGB-D	74.9	-
Interaction Learning [28]	Joints-D	75.2	83.1
Clips+CNN+MTLN [17]	Joints	79.6	84.8
Proposed			
Skepxel$_{\text{loc}}$	Joints	77.4	87.0
Skepxel$_{\text{loc+vel}}$	Joints	**81.3**	**89.2**

Evaluation on the NUCLA Dataset: We took the CNN model trained for the NTU cross-view evaluation as a baseline. Firstly, we directly applied this model on the NUCLA dataset to evaluate the generalization of our model on the unseen skeleton data. Secondly, we fine-tuned the model with the NUCLA dataset and conducted the evaluation again to evaluate performance on this dataset. Table 2 summarizes our results on the NUCLA dataset. The proposed method for the skeleton images alone achieves 83.0% average accuracy without fine-tuning on the target dataset, which demonstrates the generalization of our technique. After fine-tuning, the average accuracy increases by 2.2%. The best performance is achieved when we combined the skeleton and the velocity images, improving the accuracy over the nearest competitor by 5.7%.

Evaluation on the UTD-MHAD Dataset: For the UTD-MHAD dataset, we evaluated the performance of our technique using the models pre-trained with the NTU dataset. Table 3 summarizes our results. The proposed approach achieves a significant accuracy gain of 9.3% over the nearest competitor. We note that our Skepxels representation can be used with multiple existing CNN architectures, which provides the opportunity to extract varied network features. Exploiting ensembles/concatenation of such features, it is pos-

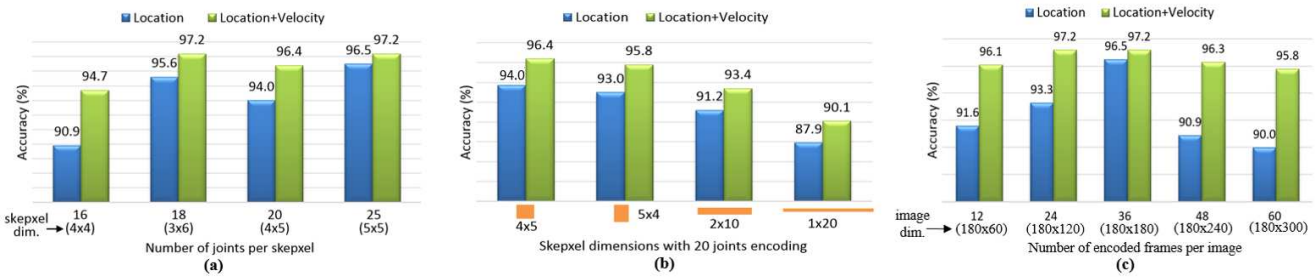

Figure 8. Performance on UTD-MHAD dataset [2] with different parameters. (a) Number of joints are varied by dropping/interpolating joints. (b) Skepxel dimensions are varied. In (a) and (b), the number of skepxels and frames are chosen such that the final image is 180×180. (c) 36 skepxels of size 5×5 are used per frame, and the number of frames is varied. All images are resized to 180×180.

Table 2. Accuracy (%) on the NUCLA dataset. $V^3_{1,2}$ means view 1, 2 were used for training and view 3 was used for testing. $\text{Skepxel}^*_{\text{loc}}$ used the NTU cross-view model without fine-tuning.

Method	Data	$V^3_{1,2}$	$V^2_{1,3}$	$V^1_{2,3}$	Mean
Baseline					
Hankelets [21]	RGB	45.2	-	-	45.2
JOULE [12]	RGB/D	70.0	44.7	33.3	49.3
DVV [22]	Depth	58.5	55.2	39.3	51.0
CVP [53]	Depth	60.6	55.8	39.5	52.0
AOG [46]	Depth	73.3	-	-	-
nCTE [7]	RGB	68.6	68.3	52.1	63.0
NKTM [29]	RGB	75.8	73.3	59.1	69.4
R-NKTM [31]	RGB	78.1	-	-	-
HPM+TM [30]	Depth	**91.9**	75.2	71.9	79.7
Proposed					
$\text{Skepxel}^*_{\text{loc}}$	Joints	89.9	83.9	75.2	83.0
$\text{Skepxel}_{\text{loc}}$	Joints	88.8	85.3	**81.6**	85.2
$\text{Skepxel}_{\text{loc+vel}}$	Joints	91.5	**85.5**	79.2	**85.4**

sible to achieve further performance gain using our method. We provide discussion on this aspect of our approach in the supplementary material of the paper.

6. Ablation Experiments

Different Skepxel Encoding Schemes: We demonstrate how the overall recognition performance is affected by (a) altering the number of joints encoded per Skepxel, (b) changing the Skepxel dimensions with fixed number of joints; and (c) changing the number of frames encoded per image. We chose UTD-MHAD dataset [2] for these experiments because the proposed representation achieved significant performance improvements for this dataset. The summary of the results of our ablation experiments is presented in Fig. 8. The overall results demonstrate effective encoding by skepxels. The location+velocity encoding is always able to improve the performance, which is intuitive because of more information being encoded in the representation.

Testing Skepxels on RGB Actions To demonstrate that the effectiveness of Skepxels is not limited to the datasets that provide precise skeleton information, we also performed experiments by extracting inaccurate skeleton information from RGB videos, and feeding the resulting skeltons to our approach. We extracted the skeletons from UTD-MHAD dataset [2] using the DeeperCut method [15] that gives 14 joints per frame. We added 2 more joints by interpolating between left/right hip, and hip/neck and formed images with 4×4 skepxels, assigning zeros to the z-axis values. With this setting, the recognition accuracies for the UTD-MHAD dataset are 87.2% and 92.3% for *loc.* and *loc.+vel.*, respectively. These results are comparable to the first two bars in Fig. 8 that are correspondingly computed for 4×4 skepxels with the provided accurate 3d skeletons.

Table 3. Action recognition accuracy (%) on UTD-MHAD dataset. $\text{Skepxel}^*_{\text{loc}}$ used the NTU cross-view model without fine-tuning.

Method	Data	Mean
Baseline		
ELC-KSVD [54]	Joints	76.2
kinect-Inertia [2]	Depth	79.1
Cov3DJ [14]	Joints	85.6
SOS [11]	Joints	87.0
JTM [48]	Joints	87.9
Proposed		
$\text{Skepxel}^*_{\text{loc}}$	Joints	94.7
$\text{Skepxel}_{\text{loc}}$	Joints	96.5
$\text{Skepxel}_{\text{loc+vel}}$	Joints	**97.2**

7. Conclusion

We proposed a representation that maps human joint data to images for effective processing by CNN architectures. The method exploits a basic building block, termed *Skepxel* to construct the skeletal images of arbitrary dimensions that encode spatial and spatio-temporal information of human joint locations and velocities under multiple informative joint arrangements. We showed that the proposed representation can be used to successfully capture the macro-temporal details with any CNN architecture leading to state-of-the-art human action recognition on benchmark datasets.

Acknowledgement

This research was sponsored by the Australian Research Council (ARC) grant DP160101458 and partially supported by ARC grant DP190102443. The GPU used for this research was donated by the NVIDIA Corporation.

References

[1] C. Cao, Y. Zhang, C. Zhang, and H. Lu. Body joint guided 3-d deep convolutional descriptors for action recognition. *IEEE transactions on cybernetics*, 48(3):1095–1108, 2018. 2

[2] 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*, pages 168–172. IEEE, 2015. 2, 7, 8

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009. 5

[4] M. Devanne, H. Wannous, S. Berretti, P. Pala, M. Daoudi, and A. Del Bimbo. 3-d human action recognition by shape analysis of motion trajectories on riemannian manifold. *IEEE transactions on cybernetics*, 45(7):1340–1352, 2015. 1, 2

[5] 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*, pages 579–583. IEEE, 2015. 1, 2, 3

[6] Y. Du, W. Wang, and L. Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1110–1118, 2015. 1, 2, 7

[7] A. Gupta, J. Martinez, J. J. Little, and R. J. Woodham. 3d pose from motion for cross-view action recognition via non-linear circulant temporal encoding. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2601–2608, 2014. 8

[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 1

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[10] S. Herath, M. Harandi, and F. Porikli. Going deeper into action recognition: A survey. *Image and Vision Computing*, 60:4–21, 2017. 1

[11] Y. Hou, Z. Li, P. Wang, and W. Li. Skeleton optical spectra based action recognition using convolutional neural networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 2016. 8

[12] J.-F. Hu, W.-S. Zheng, J. Lai, and J. Zhang. Jointly learning heterogeneous features for rgb-d activity recognition. In *IEEE conference on Computer Vision and Pattern Recognition*, pages 5344–5352, 2015. 7, 8

[13] Z. Huang, C. Wan, T. Probst, and L. Van Gool. Deep learning on lie groups for skeleton-based action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 2, 7

[14] M. E. Hussein, M. Torki, M. A. Gowayyed, and M. El-Saban. Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations. In *IJCAI*, volume 13, pages 2466–2472, 2013. 8

[15] E. Insafutdinov, L. Pishchulin, B. Andres, M. Andriluka, and B. Schiele. Deepercut: A deeper, stronger, and faster multi-person pose estimation model. In *European Conference on Computer Vision*, pages 34–50. Springer, 2016. 8

[16] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014. 1

[17] 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. 1, 3, 7

[18] T. Kerola, N. Inoue, and K. Shinoda. Cross-view human action recognition from depth maps using spectral graph sequences. *Computer Vision and Image Understanding*, 154:108–126, 2017. 3

[19] T. S. Kim and A. Reiter. Interpretable 3d human action analysis with temporal convolutional networks. 2017. 3

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 1

[21] B. Li, O. I. Camps, and M. Sznaier. Cross-view activity recognition using hankelets. In *IEEE conference on Computer Vision and Pattern Recognition*, pages 1362–1369, 2012. 8

[22] R. Li and T. Zickler. Discriminative virtual views for cross-view action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2855–2862, 2012. 8

[23] J. Liu, A. Shahroudy, D. Xu, and G. Wang. Spatio-temporal lstm with trust gates for 3d human action recognition. In *European Conference on Computer Vision*, pages 816–833, 2016. 3, 7

[24] Z. Luo, B. Peng, D.-A. Huang, A. Alahi, and L. Fei-Fei. Unsupervised learning of long-term motion dynamics for videos. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 7

[25] D. Mehta, S. Sridhar, O. Sotnychenko, H. Rhodin, M. Shafiei, H.-P. Seidel, W. Xu, D. Casas, and C. Theobalt. Vnect: Real-time 3d human pose estimation with a single rgb camera. *ACM Transactions on Graphics*, 2017. 1, 2

[26] A. V. Oppenheim. *Discrete-time signal processing*. Pearson Education India, 1999. 6

[27] R. Poppe. A survey on vision-based human action recognition. *Image and vision computing*, 28(6):976–990, 2010. 1

[28] H. Rahmani and M. Bennamoun. Learning action recognition model from depth and skeleton videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5832–5841, 2017. 7

[29] H. Rahmani and A. Mian. Learning a non-linear knowledge transfer model for cross-view action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2458–2466, 2015. 8

[30] H. Rahmani and A. Mian. 3d action recognition from novel viewpoints. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1506–1515, 2016. 1, 8

[31] H. Rahmani, A. Mian, and M. Shah. Learning a deep model for human action recognition from novel viewpoints. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 8

[32] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 815–823, 2015. 5

[33] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang. Ntu rgb+d: A large scale dataset for 3d human activity analysis. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1010–1019, 2016. 2, 6, 7

[34] A. Shahroudy, T.-T. Ng, Y. Gong, and G. Wang. Deep multimodal feature analysis for action recognition in rgb+ d videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 7

[35] A. Shahroudy, T.-T. Ng, Q. Yang, and G. Wang. Multimodal multipart learning for action recognition in depth videos. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 38(10):2123–2129, 2016. 2

[36] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 3

[37] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, pages 4278–4284, 2017. 2, 5, 6

[38] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015. 1

[39] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 3

[40] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *IEEE international conference on Computer Vision*, pages 4489–4497, 2015. 1

[41] V. Veeriah, N. Zhuang, and G.-J. Qi. Differential recurrent neural networks for action recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4041–4049, 2015. 2

[42] R. Vemulapalli, F. Arrate, and R. Chellappa. Human action recognition by representing 3d skeletons as points in a lie group. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 588–595, 2014. 1, 7

[43] R. Vemulapalli and R. Chellapa. Rolling rotations for recognizing human actions from 3d skeletal data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4471–4479, 2016. 1, 2

[44] H. Wang and L. Wang. Modeling temporal dynamics and spatial configurations of actions using two-stream recurrent neural networks. *arXiv preprint arXiv:1704.02581*, 2017. 1, 3

[45] 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*, pages 1290–1297. IEEE, 2012. 2

[46] J. Wang, X. Nie, Y. Xia, Y. Wu, and S.-C. Zhu. Cross-view action modeling, learning and recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2649–2656, 2014. 2, 6, 8

[47] L. Wang, Y. Qiao, and X. Tang. Action recognition with trajectory-pooled deep-convolutional descriptors. In *IEEE conference on Computer Vision and Pattern Recognition*, pages 4305–4314, 2015. 1

[48] P. Wang, Z. Li, Y. Hou, and W. Li. Action recognition based on joint trajectory maps using convolutional neural networks. In *Proceedings of the 2016 ACM on Multimedia Conference*, pages 102–106. ACM, 2016. 8

[49] Y. Wen, K. Zhang, Z. Li, and Y. Qiao. A discriminative feature learning approach for deep face recognition. In *European Conference on Computer Vision*, pages 499–515. Springer, 2016. 6

[50] 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*, pages 20–27. IEEE, 2012. 1

[51] Y. Yang, C. Deng, D. Tao, S. Zhang, W. Liu, and X. Gao. Latent max-margin multitask learning with skelets for 3-d action recognition. *IEEE Trans. Cybernetics*, 47(2):439–448, 2017. 1, 2

[52] 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*, pages 2752–2759, 2013. 2

[53] Z. Zhang, C. Wang, B. Xiao, W. Zhou, S. Liu, and C. Shi. Cross-view action recognition via a continuous virtual path. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2690–2697, 2013. 8

[54] L. Zhou, W. Li, Y. Zhang, P. Ogunbona, D. T. Nguyen, and H. Zhang. Discriminative key pose extraction using extended lc-ksvd for action recognition. In *Digital Image Computing: Techniques and Applications (DICTA), 2014 International Conference on*, pages 1–8. IEEE, 2014. 8