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

Human skeleton joints are popular for action analysis since they can be easily extracted from videos to discard background noises. However, current skeleton representations do not fully benefit from machine learning with CNNs. We propose "Skepxels" a spatio-temporal representation for skeleton sequences to fully exploit the "local" correlations between joints using the 2D convolution kernels of CNN. We transform skeleton videos into images of flexible dimensions using Skepxels and develop a CNN-based framework for effective human action recognition using the resulting images. Skepxels encode rich spatio-temporal information about the skeleton joints in the frames by maximizing a unique distance metric, defined collaboratively over the distinct joint arrangements used in the skeletal image. Moreover, they are flexible in encoding compound semantic notions such as location and speed of the joints. The proposed action recognition exploits the representation in a hierarchical manner by first capturing the micro-temporal relations between the skeleton joints with the Skepxels and then exploiting their macro-temporal relations by computing the Fourier Temporal Pyramids over the CNN features of the skeletal images. 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 the medium-sized N-UCLA and UTH-MHAD datasets, our method outperforms the existing results by 5.7% and 9.3% respectively.

1. Introduction

Extracting human skeleton joints from videos to perform action recognition is a popular technique as it is robust to the variations in clothing, illumination conditions and background [24, 8, 47, 39, 4, 40, 41]. Moreover, recent methods can extract skeleton data in real-time from single view RGB videos [22]. Convolutional Neural Networks (CNNs) [17, 35, 6] are popular for processing raw images [44, 27, 13, 37] because they effectively exploit the correlation between the local pixels in the images, which is the key to accurate image classification. We envisage that higher human action recognition accuracy can be achieved analogously by capitalizing on the 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 the skeletons and images restrict the utility of CNNs for processing the skeleton data.

Previous attempts [14], [3] of using CNNs for human skeleton data generally use the skeleton joints of a video frame to form an image column. This severely limits the number of joints per frame in the receptive field of the 2D kernels of the CNNs, restricting the kernel's capacity to exploit the correlations between multiple skeleton joints. These methods also find it inevitable to up-sample the skeleton data to construct appropriate size images for using existing pre-trained networks. On one hand, up-sampled images suffer from ill-defined semantics; on the other, the image generation process adds noise to the data that is detrimental to the network performance. Finally, existing methods do not take into account the different ways in which skeleton joints can be arranged to form an image column.

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 the need for resampling. 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 of the sequence into one image. Thus, the resulting image is a compact representation of rich spatio-temporal information

about the action. Owing to the systematic construction of the skeletal images, it is also possible to encode multiple semantic notions about the joints in a single image - shown by encoding "location" and "velocity" of the joints.

We also propose a framework that uses the proposed Skepxels 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 [42] of the CNN features of the skeletal images. We demonstrate the use of skeletal images of different sizes with the Inception-ResNet [34]. Moreover, we also enhance the network architecture for our framework. The proposed technique is thoroughly evaluated using the NTU Human Activity Dataset [30], Northwestern-UCLA Multiview Dataset [43] and UTD Multimodal Human Action Dataset [1]. Our approach improves the stat-of-the-art performance on the large scale dataset [30] by 4.4%, whereas the 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 human action analysis is becoming even more promising as recent methods can extract skeleton data in real time using a single RGB camera [22]. Zanfir *et al*. [48] proposed a moving pose descriptor which considers both pose information and the differential quantities of the skeleton joints for human action recognition. Du *et al*. [3] 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.

Veeriah *et al*. [38] proposed the use of 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*. [4] 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.

Shahroudy *et al*. [32] 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. Their method is based on depth and skeleton data and uses the LOP (local occupancy patterns) and the HON4D (histogram of oriented 4D normals) as features. Vemulapalli and Chellappa [40] used rolling maps to represent skeletons as points in the Lie group, and modeled human actions as curves in the Lie group. By combing the logarithm map with the rolling maps, they managed to unwrap the action curves and performed classification in the non-Euclidean space.

Huang *et al*. [11] 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 generalizes the traditional neural network model to non-Euclidean Lie groups. Kerola *et al*. [15] used spatio-temporal key points and skeletons to represent an action as a temporal sequence of graphs, and then applied the spectral graph wavelet transform to create the action descriptors. Ke *et al*. [14] 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 Network was proposed for the final classification.

Wang [41] 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 [16] 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. This method imports 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*. [20] 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 [33], Inception [36], ResNet [7]. Consequently, the images are up-sampled to fit the desired network architectures [3, 14] 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 proposed approach is explained below.

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.

In our experiments (Section 5), it was observed that the CNN architectures are generally able to process the skeletal information more effectively for the square/near-square shaped Skepxels. This property of the CNNs is directly attributed to the square convolution kernels used to learn the models. 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)

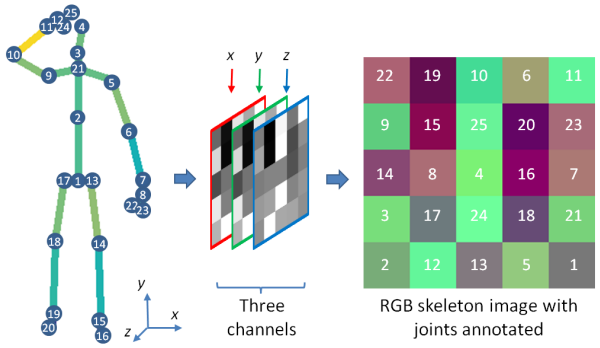

Figure 1. Illustration of a *Skepxel* rendered as an RGB image patch. The numbers on skeleton and color image share the joint description : 1-base spine, 2-middle spine, 3-neck ,4-head, 5-left shoulder, 6-left elbow, 7-left wrist, 8-left hand, 9-right shoulder, 10-right elbow, 11-right wrist, 12-right hand, 13-left hip, 14-left knee, 15-left ankle, 16-left foot, 17-right hip, 18-right knee, 19-right ankle, 20-right foot, 21-spine, 22-tip of the left hand, 23-left thumb, 24-tip of the right hand, 25-right thumb.

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} element $\mathbf{A}_i \in \mathbb{R}^{h\times w}$ 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 $\mathcal{A}^m$. 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

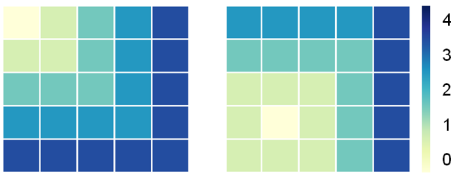

Figure 2. Illustration of the employed definition of the radial distance on 5×5 grids. If the joint α_i is located at [1,1] position in $\mathbf{A}_j^m$, the left 5×5 grid is used. For the joint location [4,2], the right grid is used. There are 25 such grids in total to measure the distance of skeleton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

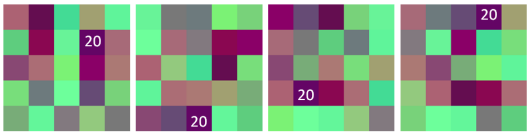

Figure 3. A group of *Skepxels* generated for a single skeleton frame. The same color corresponds to the same joint. Only joint number 20 is marked for better visibility.

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 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 four 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 $\psi_{i \in \{1,2,\ldots,m\}}$. 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 $\mathbf{I}$. As illustrated in Fig. 4, 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 $\mathbf{\Psi}_{i \in \{1,2,\ldots,n\}}$ 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. 5.

Figure 4. Illustration of *Skepxels* for a sequence of frames. The same location corresponds to the same joint.

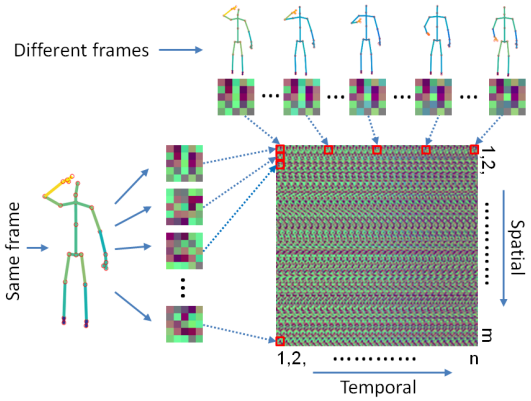

Figure 5. The final image is compactly constructed with the *Skepxels* along spatial and temporal dimensions.

Different action videos may contain different 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 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.

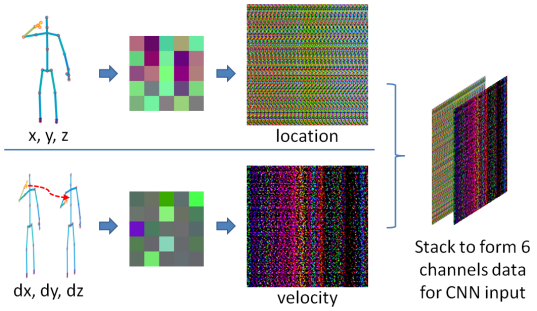

Figure 6. Joint location differences between the consecutive skeleton frames are calculated to construct the *velocity* images, which are appended to the *location* images.

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. 6. 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 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 [34] as the test bed for our representation. This recent CNN architecture has been successful in the general image classification task [2], as well as the specific tasks such as face recognition [29]. 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 [34] for the training methodology, which demonstrates the compatibility of 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 modi-

Table 1. Modified architecture of the Inception-ResNet [34]: The "STEM" part of the network is extended to fit the augmented 6-channel input images. The input and output sizes are described as $rows \times columns \times channels$. The kernel is specified as $rows \times columns \times filters, stride$.

Layer	Kernel	Output_size
Input		$180 \times 180 \times 6$
Conv2d_1a	$3 \times 3 \times 64, 2$	$89 \times 89 \times 64$
Conv2d_2a	$3 \times 3 \times 64, 1$	$87 \times 87 \times 64$
Conv2d_2b	$3 \times 3 \times 96, 1$	$87 \times 87 \times 96$
MaxPool_3a	$3 \times 3 \times 96, 2$	$43 \times 43 \times 96$
Conv2d_3b	$1 \times 1 \times 112, 1$	$43 \times 43 \times 112$
Conv2d_4a	$3 \times 3 \times 224, 1$	$41 \times 41 \times 224$
Conv2d_4b	$3 \times 3 \times 256, 2$	$20 \times 20 \times 256$
Inception-resnet-A		$20 \times 20 \times 256$
Reduction-A		$9 \times 9 \times 896$
Inception-Resnet-B		$9 \times 9 \times 896$
Reduction-B		$4 \times 4 \times 1972$
Inception-Resnet-C		$4 \times 4 \times 1792$
AvgPool_1a	$4 \times 4 \times 1792, 1$	$1 \times 1 \times 1792$
Flatten & Dropout		1792
Bottleneck		128
Softmax		class

fied the Inception-ResNet by extending the "STEM" part of the network[34]. The modified architecture is summarized in Table. 1. To train the modified network, Center loss [46] 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. The results of our experiments (reported in Section 5) demonstrate a consistent gain in the performance of the network by using the augmented images.

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 relationships 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 [23] 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.

It is noteworthy that whereas we present the macro-temporal encoding in our approach by employing subsequent processing of the CNN features, the Skepxels-based construction also allows for the direct encoding of the skeletal information over large time intervals using the larger images. Nevertheless, in this work, the main objective of the underlying approach is to demonstrate the effectiveness of the Skepxels-based representation for the common practices of exploiting the CNNs for the skeleton data. Hence, we intentionally include the explicit processing of the CNN features and show that the state-of-the-art performance is achievable with the compact representations.

4. Dataset

We perform experiments with three standard benchmark datasets for human action recognition, namely the NTU RGB+D Human Activity Dataset [30], the Northwestern-UCLA Multiview Dataset [43] and the UTD Multimodal Human Action Dataset [1]. The details of these datasets and the followed experimental protocols are given below.

4.1. NTU RGB+D Human Activity Dataset

The NTU RGB+D Human Activity Dataset [30] 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. Fig. 7 shows representative samples from this dataset.

We followed the standard evaluation protocol proposed in [30], 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.

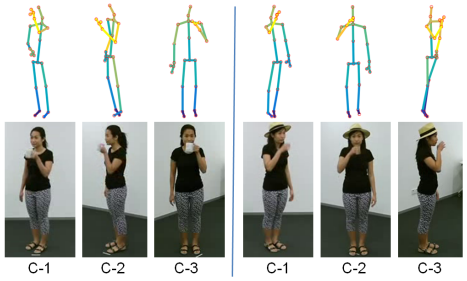

Figure 7. Skeleton and RGB sample frames from the NTU RGB+D Human Activity Dataset [30]. Three sensors C-1, C-2 and C-3 are used for recording. The left image group shows the actions recorded with the performer facing C-3. The right group is recorded when the action performer faces C-2.

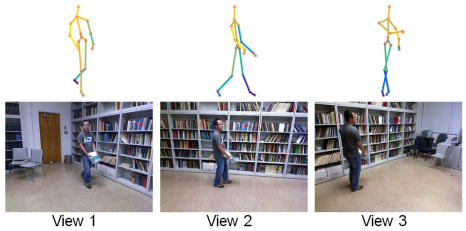

Figure 8. Sample frames of three different viewpoints from the Northwestern-UCLA Multiview Action dataset [43]

4.2. Northwestern-UCLA Multiview Dataset

This dataset [43] 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. Fig. 8 shows the representative sample frames from this dataset for the three viewpoints. 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 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.

4.3. UTD Multimodal Human Action Dataset

The UTD-MHAD dataset [1] consists of 27 different actions performed by 8 subjects. Each subject repeated the action for 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. Fig. 9 shows the sample frames from this dataset.

We follow [1] to evaluation UTH-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.

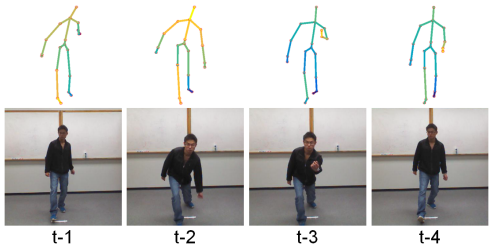

Figure 9. Four consecutive sample frames from the UTD Multimodal Human Action Dataset [1]

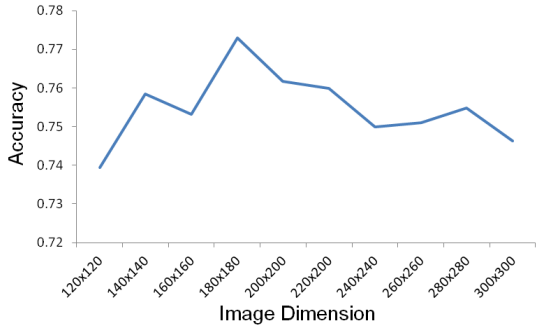

Figure 10. Action recognition performance for different skeletal image size on the NTU RGB+D Human Activity Dataset [30].

5. Experiments

5.1. 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 [30] for this purpose. According to the evaluation protocol of [30], 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. 10 shows the recognition accuracy for each setting. We eventually selected 180×180 as the image dimensions based on these results.

5.2. Evaluation on NTU RGB+D Dataset

We trained the 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 is 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 com-

Table 2. Action recognition accuracy (%) on the NTU RGB+D Human Activity Dataset.

Method	Data type	Cross Subject	Cross View
Baseline			
Lie Group [39]	Joints	50.1	52.8
Deep RNN [30]	Joints	56.3	64.1
HBRNN-L [4]	Joints	59.1	64.0
Dynamic Skeletons [10]	Joints	60.2	65.2
Deep LSTM [30]	Joints	60.7	67.3
LieNet [11]	Joints	61.4	67.0
P-LSTM [30]	Joints	62.9	70.3
LTMD [21]	Depth	66.2	-
ST-LSTM [20]	Joints	69.2	77.7
DSSCA-SSLM [31]	RGB-D	74.9	-
Interaction Learning [25]	Joints-D	75.2	83.1
Clips+CNN+MTLN [14]	Joints	79.6	84.8
Proposed			
$\text{Skepxel}_{\text{loc}}$	Joints	77.4	87.0
$\text{Skepxel}_{\text{loc+vel}}$	Joints	**81.3**	**89.2**

bined 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 architecture mentioned in Table 1 for our evaluation in the $\text{Skepxel}_{\text{loc+vel}}$ mode. Table 2 compares the performance of our approach with the existing techniques on the NTU RGB+D dataset. The proposed method is able to improve state-of-the-art accuracy by 4.4% in the $\text{Skepxel}_{\text{loc+vel}}$ mode.

5.3. Evaluation on the N-UCLA Dataset

We took the CNN model trained for the NTU RGB+D cross-view evaluation as a baseline. Firstly, we directly applied this model on the N-UCLA dataset to evaluate the generalization of our model on the unseen skeleton data. Secondly, we fine-tuned the model with the N-UCLA dataset and conducted the evaluation again.

Table 3 summarizes our results on the N-UCLA dataset. The proposed method for the skeleton images alone achieves 83.0% accuracy without fine-tunning on the target dataset, which demonstrates the generalization of our technique. After fine-tunning, the average accuracy increases by 2.2%. The state-of-the-art performance is achieved when we combined the skeleton and the velocity images, improving the accuracy over the nearest competitor by 5.7%.

5.4. Evaluation on the UTD-MHAD Dataset

For the UTH-MHAD dataset, we evaluated the performance of our technique using the models pre-trained with the NTU dataset. The performance of our technique for the different models is summarized in Table 4. The proposed approach achieves a significant accuracy gain of 9.3% on this dataset.

Table 3. Action recognition accuracy (%) on the N-UCLA Multi-view dataset. $V^3_{1,2}$ means that view 1 and 2 were used for training and view 3 was used for testing.

Method	Data	$V^3_{1,2}$	$V^2_{1,3}$	$V^1_{2,3}$	Mean
Baseline					
Hankelets [18]	RGB	45.2	-	-	45.2
JOULE [10]	RGB-D	70.0	44.7	33.3	49.3
DVV [19]	Depth	58.5	55.2	39.3	51.0
CVP [49]	Depth	60.6	55.8	39.5	52.0
AOG [43]	Depth	73.3	-	-	-
nCTE [5]	RGB	68.6	68.3	52.1	63.0
NKTM [26]	RGB	75.8	73.3	59.1	69.4
R-NKTM [28]	RGB	78.1	-	-	-
HPM+TM [27]	Depth	**91.9**	75.2	71.9	79.7
Proposed					
$\text{Skepxel}_{\text{loc}}$ (w/o ft)	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**

Table 4. Action recognition accuracy (%) on UTH-MHAD dataset.

Method	Data	Mean
Baseline		
ELC-KSVD [50]	Joints	76.2
kinect-Inertia [1]	Depth-Inertial	79.1
Cov3DJ [12]	Joints	85.6
SOS [9]	Joints	87.0
JTM [45]	RGB	87.9
Proposed		
$\text{Skepxel}_{\text{loc}}$ (w/o ft)	Joints	94.7
$\text{Skepxel}_{\text{loc}}$	Joints	96.5
$\text{Skepxel}_{\text{loc+vel}}$	Joints	**97.2**

6. Conclusion

A novel method is proposed to map the skeletal data to images that are effectively processed by the CNN architectures. The method exploits a basic building block, termed *Skepxel* - skeleton picture element, to construct the skeletal images of arbitrary dimensions. The resulting images encode fine spatio-temporal information about the human skeleton under multiple informative joint arrangements in different frames of the skeleton videos. This representation is further extended to incorporate the joint speed information in the videos. Moreover, it is also shown that the proposed compact representation can be easily used to successfully capture the macro-temporal details in the videos. When used with the Inception-ResNet architecture, the proposed skeletal image representation result in the state-of-the-art skeleton action recognition performance on the stan-

dard large scale action recognition datasets.

References

[1] 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] 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.

[3] 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.

[4] 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.

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

[6] 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.

[7] 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.

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

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

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

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

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

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

[25] 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.

[26] 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.

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

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

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

[34] 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.

[35] 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.

[36] 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.

[37] 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.

[38] 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.

[39] 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.

[40] 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.

[41] 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.

[42] 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.

[43] 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.

[44] 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.

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

[46] 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.

[47] 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.

[48] 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.

[49] 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.

[50] 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 lmage Computing: Techniques and Applications (DlCTA), 2014 International Conference on*, pages 1–8. IEEE, 2014.