Article

Spatio-Temporal Image Representation of 3D Skeletal Movements for View-Invariant Action Recognition with Deep Convolutional Neural Networks

Huy Hieu Pham [1,2,*], Houssam Salmane [1], Louahdi Khoudour [1], Alain Crouzil [2], Pablo Zegers [3] and Sergio A. Velastin [4,5]

[1] Cerema Research Center, France; {huy-hieu.pham,louahdi.khoudour,houssam.salmane}@cerema.fr
[2] Informatics Research Institute of Toulouse, Paul Sabatier University, France; alain.crouzil@irit.fr
[3] Aparnix, La Gioconda 4355, 10B, Las Condes, Santiago, Chile; pablozegers@gmail.com
[4] Cortexica Vision Systems Ltd., London; sergio.velastin@ieee.org
[5] Queen Mary University of London, London, UK and Cortexica Vision Systems Ltd., London, UK and Department of Computer Science, University Carlos III of Madrid, Madrid, Spain
* Correspondence: huy-hieu.pham@cerema.fr (Hieu Pham); Tel.: +33-6055-68269

Version March 7, 2019 submitted to Preprints

Abstract: Designing motion representations for the problem of 3D human action recognition from skeleton sequences is an important yet challenging task. An effective representation should be robust to noise, invariant to viewpoint changes and result in a good performance with low-computational demand. Two main challenges in this task include how to efficiently represent spatio-temporal patterns of skeletal movements and how to learn their discriminative features for classification task. This paper presents a novel skeleton-based representation and a deep learning framework for 3D action recognition using RGB-D sensors. We propose to build an action map called SPMF (*Skeleton Posture-Motion Feature*), which is a compact image representation built from skeleton poses and their motions. An Adaptive Histogram Equalization (AHE) algorithm is then applied on the SPMF to enhance their local patterns and form an enhanced action map, namely Enhanced-SPMF. For learning and classification tasks, we exploit Deep Convolutional Neural Networks based on the DenseNet architecture to learn directly an end-to-end mapping between input skeleton sequences and their action labels via the Enhanced-SPMFs. The proposed method is evaluated on four challenging benchmark datasets, including both individual actions, interactions, multiview and large-scale datasets. The experimental results demonstrate that the proposed method outperforms previous state-of-the-art approaches on all benchmark tasks, whilst requiring low computational time for training and inference.

Keywords: 3D human action recognition; Skeleton-based representation; SPMF; Enhanced-SPMF; AHE; D-CNNs; DenseNet

1. Introduction

Human action recognition [1] is one of the most important and challenging tasks in computer vision. Detecting and recognizing correctly what humans do in unknown videos serve as a key component of many real-world applications such as smart surveillance [2,3], human-object interaction [4,5], autonomous vehicle technology [6,7], etc. Although significant progress has been achieved over two decades of research, video-based human action recognition is still a challenging issue due to a number of obstacles, *e.g.* changes in camera viewpoint, occlusions, background, surrounding distractions, diversity in length and speed of actions [8].

As many other visual recognition tasks, traditional approaches on human action recognition [9] have focused on extracting hand-crafted local features and building local descriptors from RGB sequences provided by 2D cameras. Some typical examples that have been widely exploited with success are SIFT [10,11], HOG/HOF [12,13], HOG-3D [14], Cuboids [15], SURF [16] and Extended

SURF [17]. Since these approaches typically recognize actions based on the appearance and movement of the human body parts from a monocular RGB video sequence, they tend to lack 3D structure from the scene. Therefore, single modality human action recognition based only on RGB videos is not enough to overcome the current challenges.

The availability of low-cost and easy-to-use depth sensors such as Microsoft Kinect ^{TM}sensor [18], ASUS Xtion [19], Intel® RealSense^{TM} [20] or Orbbec Astra [21] has helped the computer vision community improve action recognition. These sensors are able to provide detailed 3D structural information of human motion, which is considered complex for traditional 2D cameras. Many action recognition approaches using RGB-D cameras have been proposed and advanced the state-of-the-art [22–28]. In particular, most of currently depth-sensing cameras have integrated real-time skeleton estimation and tracking frameworks [29,30], helping to facilitate the collection of skeleton sequences. This data source is a high-level representation allowing to describe human action in a more precise and effective way, which is suitable for the problem of action analysis and recognition.

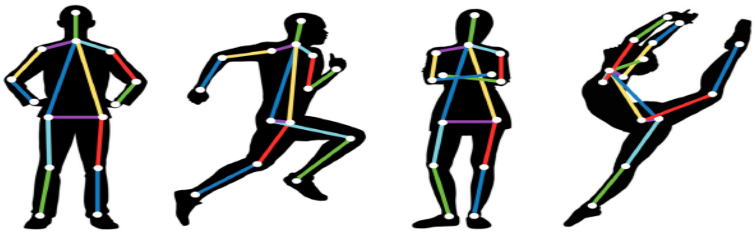


Figure 1. Illustration of skeletal data, a high-level representation, robust to variations of viewpoints as well as surrounding distractions. Biological observations [31] indicate that human beings are able to recognize actions from just the motion of a few skeleton joints. Image reproduced from the work by Bearman *et al.* [32].

Skeleton-based human action recognition is a time-series problem. As can be seen in Figure 1, raw skeletal data comprises 3D coordinates of the key joints in the human body over time. This is an effective representation for structured motion [33] because each human action can be represented through the movement of skeleton sequences. Moreover, a large set of actions can be distinguished from these movements [31]. 3D skeletal data is not only invariant to camera-viewpoint but also can be estimated in real-time. Moreover, it is available for most of depth based action datasets [34]. Hence, exploiting this data source for 3D human action recognition opens up opportunities for addressing the limitations of RGB-depth modalities-based solutions and so many skeleton-based action recognition approaches have been proposed [22,26,35–37]. Our goal is to exploit the potential of low-cost consumer depth cameras [18–21] for identifying salient spatio-temporal patterns in skeleton sequences and then explore them for improving the recognition of human actions using deep learning models.

In the literature of skeleton-based action recognition, there are two main issues that need to be solved. The first challenge is to find a skeleton-based representation that transforms the raw skeletal data into a representation that effectively captures the spatio-temporal evolutions of human skeleton joints. The second challenge is to model and recognize actions that are complex, variable and have large intra-class correlation, from the skeleton-based representation. Previous studies [22,26,35,38–46] on this topic can be divided into two main categories: skeleton-based action recognition based on hand-crafted features and skeleton-based action recognition using deep neural networks. The first group of methods uses hand-crafted local features and probabilistic graphical models such as Hidden Markov Model (HMM) [47], Conditional Random Field (CRF) [38], or Fourier Temporal Pyramid (FTP) [26] to model and classify actions. However, almost all of these approaches are shallow, data-dependent and require a lot of feature engineering. The second group of methods considers skeletal data as a time-series patterns and proposes the use of Recurrent Neural Networks (RNNs) [48], especially Recurrent Neural Networks with Long Short-Term Memory units (RNN-LSTMs) [49,50] to analyze and model the contextual information contained in the skeleton sequences. They are considered as the most popular deep learning based approach for skeleton-based action recognition and have achieved high-level performance. Although being able to model the long-term temporal of human motion, RNN-LSTMs [49,50] just consider skeleton sequences as a kind of low-level features by feeding raw skeletal data directly into the network input. The huge number of input features makes them complex, time-consuming and may easily lead to overfitting. Nevertheless, almost all of these networks act just as classifiers and do not extract high-level features for recognition tasks [51].

A practical human action recognition system should be able to detect and recognize actions from different viewpoints, robust to noise and operate in real-time. We believe that an efficient and effective representation for 3D human motion plays a decisive role in improving recognition performance. Motivated by the success of our previous work on the SPMF (*Skeleton Posture-Motion Feature*) representation [52] for video-based human action recognition, in this paper we aim to find a new skeleton-based representation and take full advantages in learning highly hierarchical image features of Deep Convolutional Neural Networks (D-CNNs) to build an end-to-end learning framework for 3D human action recognition from skeletal data. Specifically, we propose a new 3D motion representation, termed as Enhanced-SPMF (*Enhanced Skeleton Posture-Motion Feature*). Similar to the SPMF [52], the proposed Enhanced-SPMF has a 2D image structure with three color channels, which is built from a set of spatio-temporal stages, combining 3D skeleton poses and their motions. Moreover, an Adaptive Histogram Equalization (AHE) algorithm [53] is then applied to the color images to enhance their local patterns and generate more discriminative features for classification task. Figure 2 illustrates an overview of the proposed Enhanced-SPMF. To learn image features and recognize action labels from the proposed representation, different D-CNN models based on the DenseNet architecture [54] have been designed and evaluated.

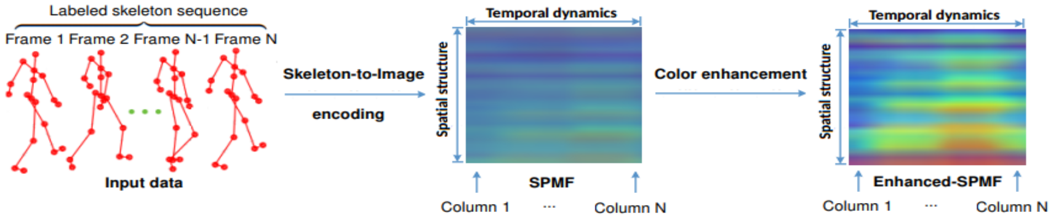


Figure 2. Overview of the proposed Enhanced-SPMF representation. Each skeleton sequence is transformed into a single RGB that is a motion map called SPMF [52]. A color enhancement technique [53] is then used to highlight the motion map and form the Enhanced-SPMF, which will be learned and classified by a deep learning model.

There are five important hypotheses that motivate us to propose a new skeleton-based representation and design DenseNets [54] for 3D human action recognition with skeletal data. **First**, human actions can be correctly represented through the skeleton movements [31,33]. **Second**, compared to RGB and depth streams that contain thousands of pixels per frame, skeletal data has a high-level abstraction with much less complexity. This makes the training and inference processes much simpler and faster. **Third**, the spatio-temporal dynamics of skeleton sequences can be transformed into color images – a kind of 3D tensor-structured representation that can be effectively learned by representation learning models as D-CNNs. **Fourth**, many different action classes share a great number of similar primitives, which interferes with action classification. Therefore, extracting essential spatio-temporal patterns from skeleton movements plays a key role in this task. **Last**, recent research results indicate that CNNs have achieved outstanding performances in many image recognition tasks [55,56]. There are a many signs that seem to indicate that the learning performance of CNNs can be significantly improved by increasing the depth of their architectures [57–60]. In particular, D-CNNs with architectures such as DenseNet [54] can improve accuracy in the image recognition task since this kind of network is able to prevent overfitting and degradation phenomena [61] by maximizing information flow and facilitating features reuse as each layer in its architecture has direct access to the features from previous layers. Therefore, we explore the use of

DenseNet in this work and optimise this architecture for learning and recognizing human actions on the proposed image-based representation.

The effectiveness of the proposed method is evaluated for four public benchmark RGB-D datasets, including MSR Action3D [62], KARD [63], SBU Kinect Interaction [64], and NTU-RGB+D datasets [43]. The hypotheses above were reinforced since the experimental results show that we achieve state-of-the-art performance on all the reported benchmarks. Furthermore, we also report the effectiveness of this approach in terms of computational cost, for both training time and inference latency. Overall, the main contributions of our study lie mainly on the following aspects:

- **Firstly**, we present Enhanced-SPMF, a new skeleton-based representation for 3D action recognition from skeletal data. The Enhanced-SPMF is an extension of SPMF that we presented in [52]. It is able to capture the spatio-temporal dynamics of skeleton movements and transforms them into a 2D structure as a single RGB image, which suits the problem of representation learning with D-CNNs. We demonstrate in this work that the proposed Enhanced-SPMF representation leads to better overall action recognition performance than the SPMF [52].
- **Secondly**, we present a deep learning framework[1] based on the DenseNet architecture [54] for learning discriminative features from the proposed Enhanced-SPMF and performing action classification. The framework directly learns an end-to-end mapping between skeleton sequences and their action labels with little pre-processing.
- **Thirdly**, we evaluate the proposed method on four highly competitive benchmark datasets and demonstrate significantly improvement over the existing state-of-the-art approaches. Our computational efficiency evaluations show that the proposed method is able to achieve high-level of performance whilst requiring low computational time for both the training and inference stages.

The rest of this paper is organized as follows: Section 2 discusses related works. Section 3 presents the details of the proposed approach. Datasets and experiments are described in Section 4. The experimental results and analyses are provided in Section 5. Section 6 concludes the paper.

2. Related work

In this section, we briefly review the exiting literature closely related to the topic of deep learning based approaches for 3D human action recognition from skeleton sequences, including skeleton-based action recognition using hand-crafted features and deep learning-based action recognition. We encourage the readers to refer to an extensive review by Han *et al.* [65] for getting a more comprehensive picture on this topic.

2.1. Hand-crafted approaches for skeleton-based human action recognition

Earlier studies on skeleton-based human action recognition focus on finding well-designed hand-crafted features and using temporal graphical models to analyze the global temporal evolution of skeleton joints. Since when the first work on 3D human action recognition from depth data was introduced [62], many approaches for skeleton-based action recognition have been proposed [22,26,35,38–40]. The common characteristic of these approaches is that, they extract geometric features of 3D joint movements and model their temporal information by a generative model. For instance, Wang *et al.* [22] represented the human motion by means of the pairwise relative positions of the skeleton joints for generating more discriminative features. Fourier Temporal Pyramid (FTP) [22] was then proposed to model the temporal dynamics of the actions from LOPs. Vemulapalli *et al.* [26] represented the 3D geometric relationships of body parts as points in a Lie Group and then exploited Dynamic Time Warping (DTW) [66] and Fourier Temporal Pyramid (FTP) [22] to model their temporal dynamics. Xia *et al.* [35] extracted and computed histograms of 3D joint locations (HOJ-3D) to represent

[1] The implementation and models will be made publicly available at https://github.com/cerema-lab/Sensors-2018-HAR-SPMF.

actions via posture visual words. The temporal evolutions of those words are modeled by a discrete Hidden Markov Models (HMM) [67]. Instead of modeling temporal evolution of skeletons, Luo *et al.* [39] proposed a discriminative dictionary learning algorithm (called DL-GSGC) that incorporated both group sparsity and geometry constraints to learn motion features from the 3D joint positions. An encoding technique called Temporal Pyramid Matching (TPM) [39] was then used for keeping the temporal information and performing action classification.

Although promising results have been achieved, the above approaches have some limitations that are difficult to overcome. For instance in many cases, they require pre-processing input data in which the skeleton sequences need to be segmented or aligned. Unlike these approaches, we propose a skeleton-based representation and a deep learning framework for 3D human action recognition that learns to recognize actions directly from the original skeletons in an end-to-end manner, without dependence on the length of actions. Moreover, the proposed solution is general and can be applied with some other data modalities such as motion capture data [68] and the output of pose estimation algorithms [32,69].

2.2. Deep learning approaches for skeleton-based human action recognition

Approaches based on Recurrent Neural Network with Long Short-Term Memory units (RNN-LSTM) [49,70] are the most popular deep learning approach for skeleton-based action recognition and have achieved high-level performance for video-based action recognition tasks [41–46]. The temporal evolutions of skeletons are spatio-temporal patterns. Thus, they can be modeled by memory cells in the structure of RNN-LSTMs [49,70]. For instance, Du *et al.* [41] proposed to use a hierarchical RNN to model the long-term contextual information of skeletal data, in which the human skeleton was divided into five parts according to its physical structure. Each low-level part was modeled by an RNN and then combined into the final representation of high-level parts for action classification. Shahroudy *et al.* [43] introduced a part-aware LSTM human action learning model by splitting a long-term memory of the entire motion to part-based cells. The long-term context of each body part was learned independently. The output of the network was then formed as a combination of independent body part context information. Liu *et al.* [44] presented a spatio-temporal LSTM network, called ST-LSTM, for 3D action recognition from skeletal data. They proposed a skeleton-based tree traversal technique to feed the structure of the skeletal data into a sequential LSTM network and improved the performance of the ST-LSTM by adding more trust gates. Recently, Liu *et al.* [46] focused on selecting the most informative skeleton joints by using a new class of LSTM network, namely Global Context-Aware Attention LSTM (GCA-LSTM), for 3D skeleton-based action recognition. Two LSTM layers were used. The first layer encodes the input sequences and generates an initial global context memory for these sequences. Meanwhile, the second layer performs attention over the input sequences with the assistance of obtained global context memory. The attention representation was then used back to refine the global context. Multiple attention iterations are executed and the final global contextual information is used for action classification task.

Compared to the approaches based on hand-crafted local features, the RNN-LSTM based approaches and their variants have been showing superior action recognition performance. However, they tend to overemphasize the temporal information and lose the spatial information of skeletons [41–46]. RNN-LSTM based approaches still struggle to cope to scope with the complex spatio-temporal variations of skeletal movements due to a number of issues such as jitters and movement speed variability. Another drawback of the RNN-LSTM networks [49,70] is that they just model the overall temporal dynamics of actions without considering the detailed temporal dynamics of them. To overcome these limitations, we propose in this study a CNN-based approach that is able to extract discriminative features of actions and model various temporal dynamics of skeleton sequences via the proposed Enhanced-SPMF representation, including both short-term, medium-term, and long-term actions.

3. Method

The details of the proposed approach are presented in this section. Figure 3 illustrates the key components of the proposed learning framework for recognizing actions from skeleton sequences. We first show how skeleton pose and motion features can be combined to build an action map in the form of an image-based representation (Section 3.1), and how to use a color enhancement technique for improving the discriminative ability of the proposed representation (Section 3.2). We then introduce an end-to-end deep leaning framework based on DenseNets to learn and classify actions from the enhanced representations (Section 3.3). Before that, in order to put the proposed approach into context, it is useful to review the central ideas behind the original DenseNet architecture (Section 3.3.1).

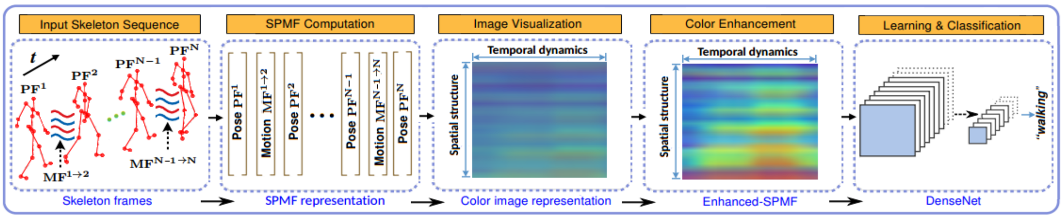


Figure 3. Schematic overview of the proposed approach. Each skeleton sequence is encoded in a single color image via a skeleton-based representation called SPMF. Each SPMF is built from pose vectors (PFs) and motion vectors (MFs) extracted from skeleton joints. They are then enhanced by an Adaptive Histogram Equalization (AHE) [53] algorithm and fed to a D-CNN for learning discriminative features and performing action classification. To achieve high-level learning performance during the training phase, we design and optimize different D-CNN models based on deep DenseNet [54], a recent state-of-the-art architecture for image recognition tasks.

3.1. SPMF: Building action map from skeletal data

One of the major challenges in exploiting D-CNNs for skeleton-based action recognition is how the spatio-temporal patterns of skeleton movements could be effectively represented and fed to D-CNNs for representation learning. As D-CNNs work well on image representations [71], our idea therefore is to encode the whole skeleton sequence into a single 2D image as a global representation for the action sequence. In general, two essential elements that determine a human action are poses and their motions. Hence, we decide to transform these two important elements into the static spatial structure of a color image with three R, G, B channels. Specifically, we propose a new representation, namely Enhanced-SPMF (*Enhanced Skeleton Pose-Motion Feature*), which is built from pose and motion vectors extracted from the skeleton joints. Note that, combining multiple kinds of geometric features such as joint coordinates, lines and planes determined by the joints will lead to lower performance than using only a single type of feature or several main type of features [72]. Moreover, it has been reported [64] that joint features such as joint-joint distance and joint-joint motion are the strongest features among many others.

3.1.1. Pose Features (PFs) computation

Given a skeleton sequence $\mathcal{S}$ with N frames, denoted by $\mathcal{S} = \{\mathbf{F}^t\}$, where $t = 1, 2, 3, ..., N$. Let $\mathbf{p}_j^t$ and $\mathbf{p}_k^t$ be the 3D coordinates of the j-th and k-th joints in $\mathbf{F}^t$. The **J**oint-**J**oint **D**istance $\mathbf{JJD}_{jk}^t$ between $\mathbf{p}_j^t$ and $\mathbf{p}_k^t$ at timestamp t is computed as

$$\mathbf{JJD}_{jk}^t = ||\mathbf{p}_j^t - \mathbf{p}_k^t||_2, \quad (t = 1, 2, 3, ..., N), \tag{1}$$

where $||\cdot||_2$ denotes the Euclidean distance between two joints. The joint distances obtained by Eq. (1) for all types of actions of a specific dataset range from $\mathbf{D}_{\min} = 0$ to $\mathbf{D}_{\max} = \texttt{max}\{\mathbf{JJD}_{jk}^t\}$. We note this distance space as $\mathcal{D}_{\textbf{original}}$. In fact, $\mathcal{D}_{\textbf{original}}$ can be transformed into a tensor-structure and fed directly to D-CNNs for learning action features. However, since $\mathcal{D}_{\textbf{original}}$ is a high-dimensional space, it could lead D-CNNs to overfit as well as being time-consuming. Thus, we need to describe the input skeleton sequences as low-dimensional signals such that they are easy to parameterize by learning models and discriminative enough for a classification task. To do that, we normalize all elements of $\mathcal{D}_{\textbf{original}}$ to the range $[\mathbf{0},\mathbf{1}]$, denoted as $\mathcal{D}_{[0,1]}$. To reflect the change in joint distances, we encode $\mathcal{D}_{[0,1]}$ into a color space using a sequential discrete color palette. The encoding process converts the joint distances $\mathbf{JJD}_{jk}^{t} \in \mathcal{D}_{[0,1]}$ into color points $\in \mathbb{N}^{3}_{[0,255]}$ performed by 256-color JET scale[2]. The use of a discrete color palette allows us to reduce complexity of input features. This helps accelerate the convergence rate of deep learning networks during the training stage.

Besides the distance information, the orientation between joints is also important for describing human motions. The **J**oint-**J**oint **O**rientation $\mathbf{JJO}_{jk}^{t}$ from joint $\mathbf{p}_j^t$ to $\mathbf{p}_k^t$ at time-stamp t, represented by the vector $\mathbf{JJO}_{jk}^{t}$ and computed as

$$\mathbf{JJO}_{jk}^{t} = \overrightarrow{\mathbf{p}_j^t\mathbf{p}_k^t} = \mathbf{p}_j^t - \mathbf{p}_k^t, \quad (t = 1, 2, 3, ..., N). \tag{2}$$

Each vector $\mathbf{JJO}_{jk}^{t}$ is a 3D vector where all of its components $\mathbf{p}$ can be normalized to the range $[\mathbf{0},\mathbf{255}]$. This can be done via the following transformation

$$\mathbf{p}_{\text{norm}} = \texttt{floor}(255 \times \frac{\mathbf{p} - \mathbf{c}_{\min}}{\mathbf{c}_{\max} - \mathbf{c}_{\min}}), \tag{3}$$

where $\mathbf{p}_{\text{norm}}$ indicates the normalized value, $\mathbf{c}_{\max}$ and $\mathbf{c}_{\min}$ are the maximum and minimum values of all coordinates over the training set, respectively. The function $\texttt{floor}(\cdot)$ rounds down to the nearest integer. We consider three components (x, y, z) of $\mathbf{JJO}_{jk}^{t}$ after normalization as the corresponding three components (R, G, B) of a color pixel and define "*a human pose*" at timestamp t by vector $\mathbf{PF}^{t}$ that describes the distance and orientation relationship between skeleton joints,

$$\mathbf{PF}^{t} = \left[\mathbf{JJD}_{jk}^{t} +\!\!\!+ \mathbf{JJO}_{jk}^{t}\right], \quad (t = 1, 2, 3, ..., N). \tag{4}$$

Here the symbol $(+\!\!\!+)$ horizontally concatenates vectors $\mathbf{JJD}_{jk}^{t}$ and $\mathbf{JJO}_{jk}^{t}$ together.

3.1.2. Motion Features (MFs) computation

Let $\mathbf{p}_j^t$ and $\mathbf{p}_k^{t+1}$ denote the 3D coordinates of the j-th and k-th joints at two consecutive frames $\mathbf{F}^t$ and $\mathbf{F}^{t+1}$. Similarly to $\mathbf{JJD}_{jk}^{t}$ in Eq. (1), the **J**oint-**J**oint **D**istance $\mathbf{JJD}_{jk}^{t,t+1}$ between $\mathbf{p}_j^t$ and $\mathbf{p}_k^{t+1}$ is computed as

$$\mathbf{JJD}_{jk}^{t,t+1} = ||\mathbf{p}_j^t - \mathbf{p}_k^{t+1}||_2, \quad (t = 1, 2, 3, ..., N-1). \tag{5}$$

Also, similarly to Eq. (2), the **J**oint-**J**oint **O**rientation $\mathbf{JJO}_{jk}^{t,t+1}$ from joint $\mathbf{p}_j^t$ to $\mathbf{p}_k^{t+1}$, represented by the vector $\mathbf{JJO}_{jk}^{t,t+1}$ as

$$\mathbf{JJO}_{jk}^{t,t+1} = \overrightarrow{\mathbf{p}_j^t\mathbf{p}_k^{t+1}} = \mathbf{p}_j^t - \mathbf{p}_k^{t+1}, \quad (t = 1, 2, ..., N-1). \tag{6}$$

We define "*a human motion*" from t to $t+1$ by vector $\mathbf{MF}^{t\to t+1}$, in which

$$\mathbf{MF}^{t\to t+1} = \left[\mathbf{JJD}_{jk}^{t,t+1} +\!\!\!+ \mathbf{JJO}_{jk}^{t,t+1}\right], \quad (t = 1, 2, ..., N-1), \tag{7}$$

where $\mathbf{JJD}_{jk}^{t,t+1}$ and $\mathbf{JJO}_{jk}^{t,t+1}$ are encoded as 3D vectors and qualified for the color representation as $\mathbf{JJD}_{jk}^{t}$ and $\mathbf{JJO}_{jk}^{t}$, respectively.

[2] A JET color map is based on the order of colors in the spectrum of visible light, ranging from blue to red, and passing through the cyan, yellow, and orange.

3.1.3. Building global action map from PFs and MFs

Based on the obtained **PFs** and **MFs**, we propose a new skeleton-based representation called **SPMF** for 3D human action recognition. To this end, all **PFs** and **MFs** computed from the skeleton sequence $\mathcal{S}$ are concatenated into a single feature vector in temporal order from the beginning to the end of the action. It is a global representation for the whole skeleton sequence $\mathcal{S}$ without dependence on the range of action and can be obtained by

$$\mathbf{SPMF}^{\mathcal{S}} = [\mathbf{PF}^{1} \mathbin{+\!\!+} \mathbf{MF}^{1\to 2} \mathbin{+\!\!+} \mathbf{PF}^{2} \mathbin{+\!\!+} ... \mathbin{+\!\!+} \mathbf{PF}^{t} \mathbin{+\!\!+} \mathbf{MF}^{t\to t+1} \mathbin{+\!\!+} \mathbf{PF}^{t+1} ... \mathbin{+\!\!+} \mathbf{PF}^{N-1} \mathbin{+\!\!+} \mathbf{MF}^{N-1\to N} \mathbin{+\!\!+} \mathbf{PF}^{N}]. \tag{8}$$

Figure 4 (*top row*) shows some **SPMFs** obtained from the MSR Action3D dataset [62] in which all images are resized to 32 × 32 pixels. Before computing the **SPMF**, a Savitzky-Golay smoothing filter [41,73] is adopted to reduce the effect of noise on skeletal data. In the experiments, we use the filter

$$\mathbf{f}^{t} = \frac{-3\mathbf{c}^{t-2} + 12\mathbf{c}^{t-1} + 17\mathbf{c}^{t} + 12\mathbf{c}^{t+1} - 3\mathbf{c}^{t+2}}{35}, \tag{9}$$

where $\mathbf{c}^{t}$ denotes the skeleton joint coordinates of frame $\mathbf{F}^{t}$ $(t = 1, 2, ..., N)$ and $\mathbf{f}^{t}$ denotes the filtering result. This filter design method is described in detailed in **Appendix A**.

3.2. *Enhanced-SPMF: Building enhanced action map*

The skeleton-based representations obtained by Eq. (8) mainly reflect the spatio-temporal distribution of skeleton joints. We visualize these representations and observe that they tend to be low contrast images, as shown in Figure 4 (*top row*). In this case, a color enhancement method can be useful for increasing contrast and highlighting the texture and edges of the motion maps. Therefore, it is necessary to enhance the local features on the generated color images after encoding. The Adaptive Histogram Equalization (AHE) [53] is a common approach for this task. This technique is capable of enhancing the local features of an image. Mathematically, let $\boldsymbol{I}$ be a given digital image, represented as a r-by-c matrix of integer pixels with intensity levels in the range $[0, \mathcal{L} - 1]$. The histogram of image $\boldsymbol{I}$ will be defined by

$$H_k = n_k, \tag{10}$$

where n_k is the number of pixels in $\boldsymbol{I}$ with intensity k. The probability of occurrence of intensity level k in $\boldsymbol{I}$ can be estimated by

$$p_k = \frac{n_k}{r \times c}, \quad (k = 0, 1, 2, ..., \mathcal{L} - 1). \tag{11}$$

The histogram equalized image is defined by transforming the pixel intensities, n, of $\boldsymbol{I}$ by the function

$$T(n) = \texttt{floor}((\mathcal{L} - 1) \sum_{k=0}^{n} p_k), \quad (n = 0, 1, 2, ..., \mathcal{L} - 1), \tag{12}$$

The Histogram Equalization (HE) method is used for increasing the global contrast of the image. However, it cannot solve the problem of increasing local contrast. To overcome this limitation, the image needs to be divided into $\mathcal{R}$ regions and the HE is then applied in each and every one of these regions. This technique is called the Adaptive Histogram Equalization algorithm (AHE) [53]. The bottom row of Figure 4 shows samples of the enhanced motion map with $\mathcal{R}$ = 8 on 32 × 32 images, which we refer to it as Enhanced-SPMF, for some actions from the MSR Action 3D dataset [62].

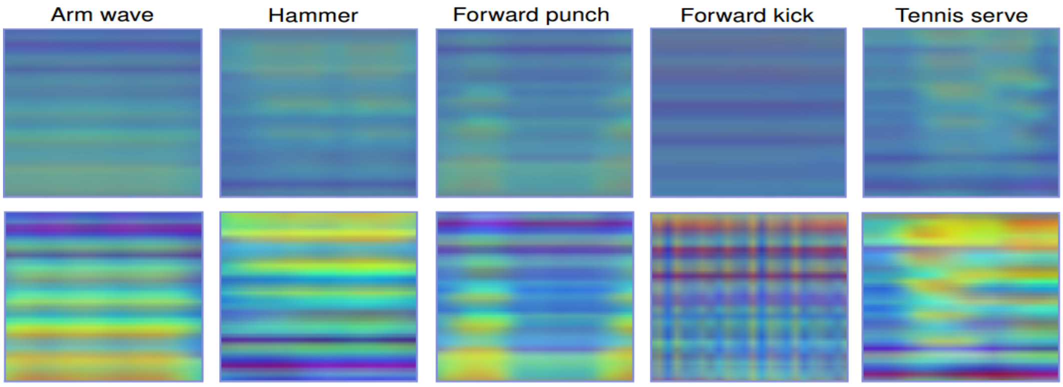


Figure 4. Results of the skeleton-to-image mapping process. The top row shows the proposed SPMF representations obtained from some samples of the MSR Action3D dataset [62]. The change in color reflects the change of distance and orientation between the joints. The bottom row shows generated images after applying the AHE algorithm [53].

3.3. Deep learning model

3.3.1. Densely Connected Convolutional Networks

DenseNet [54], considered as the current state-of-the-art CNN architecture, has some interesting properties. In this architecture [54], each layer is connected to all the others within a dense block and all layers can access to the feature maps from their preceding layers. Besides, each layer receives direct information flow from the loss function through the shortcut connections. These properties help DenseNet [54] to be less prone to overfitting for supervised learning problems. Mathematically, traditional CNN architectures, e.g. AlexNet [55] or VGGNet [57] connect the output feature maps $\mathbf{x}_{l-1}$ of the $(l-1)^{\text{th}}$ layer as input to the l^{th} layer and try to learn a mapping function

$$\mathbf{x}_l = \mathcal{H}_l(\mathbf{x}_{l-1}), \tag{13}$$

where $\mathcal{H}_l(\cdot)$ is a non-linear transformation and usually implemented via a series of operations such as Convolution (**Conv.**), Rectified Linear Unit (**ReLU**) [74], Pooling [75], and Batch Normalization (**BN**) [76]. When increasing the depth of the network, the network training process becomes complex due to the vanishing-gradient problem and the degradation phenomenon [61] (see **Appendix B** for more detail). To solve these problems, He *et al.* introduced ResNet [59]. The key idea behind the ResNet architecture [59] is the presence of shortcut connections that bypass the non-linear transformations $\mathcal{H}_l(\cdot)$ with an identity function $\texttt{id}(x) = x$. By this way, each ResNet building block [59] produces a feature map $\mathbf{x}_l$ by performing the following computation

$$\mathbf{x}_l = \mathcal{H}_l(\mathbf{x}_{l-1}) + \mathbf{x}_{l-1}. \tag{14}$$

Inspired by the philosophy of ResNet [59], to maximize information flow through layers, Huang *et al.* proposed DenseNet [54] with a simple connectivity pattern: the l^{th} layer in a dense block receives the feature maps of all preceding layers as inputs. That means

$$\mathbf{x}_l = \mathcal{H}_l([\mathbf{x}_0 +\!\!+ \mathbf{x}_1 +\!\!+ \mathbf{x}_2 +\!\!+ ... +\!\!+ \mathbf{x}_{l-1}]), \tag{15}$$

where $[\mathbf{x}_0 +\!\!+ \mathbf{x}_1 +\!\!+ \mathbf{x}_2 +\!\!+ ... +\!\!+ \mathbf{x}_{l-1}]$ is a single tensor constructed by concatenation of the previous layer's output feature maps. Additionally, all layers in the architecture receive direct supervision signals from the loss function through the shortcut connections. In this manner, the network is easy to optimize and resistant to overfitting. In DenseNet [54], multiple dense blocks are connected via transition layers. Each transition layer consists of a convolutional layer and followed by an average pooling layer that

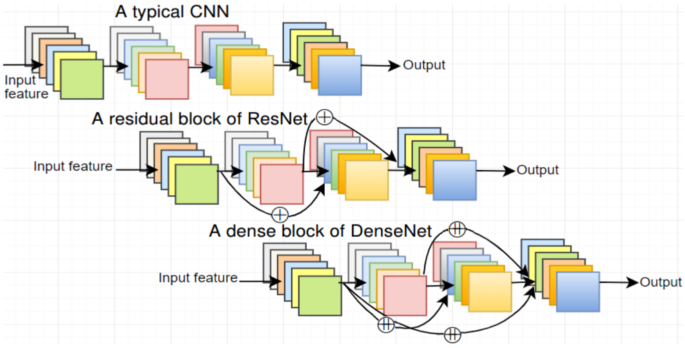


Figure 5. Illustration of the structure of a typical CNN [55] (*top row*), a ResNet building block [59] (*middle row*) and a DenseNet building block [54] (*bottom row*). The symbols ⊕ and ⊕ denote the summation and concatenation operators, respectively.

change the size of feature maps[3]. Each block with its transition layer produces k feature maps and the parameter k is called as the *"growth rate"* of the network. The non-linear function $\mathcal{H}_l(\cdot)$ in the original work [54] is a composite function of three consecutive operations: **BN-ReLU-Conv**.

3.3.2. Network design

We propose to design and optimize deep DenseNets [54] for learning and classifying human actions on the Enhanced-SPMFs. To study how recognition performance varies with architecture size, we explore different network configurations. The following configurations are used in our experiments: DenseNet (L = 100, k = 12) ; DenseNet (L = 250, k = 24); and DenseNet (L = 190, k = 40), where L is the depth of the network and k is the network growth rate. On all datasets, we use three dense blocks on 32×32 input images. In this design, $\mathcal{H}_l(\cdot)$ is defined as Batch Normalization (**BN**) [76], followed by an advanced activation layer called Exponential Linear Unit (**ELU**) [77] and 3×3 Convolution (**Conv.**). A Dropout [77] with a rate of 0.2 is used after each Convolution to prevent overfitting. After the feature extraction stage, a Full Connected (FC) layer is used for classification task in which the number of neurons for this FC layer is equal to the number of action classes in each dataset. The proposed networks can be trained in an end-to-end manner by gradient descent using Adam update rule [78]. During the training stage, we minimize a cross-entropy loss function, which is measured by the difference between the true action label $\mathbf{y}$ and the predicted action $\hat{\mathbf{y}}$ by the networks over the training samples $\mathcal{X}$. In other words, the network will be trained to solve the following optimization problem

$$\text{Arg min}_{\mathcal{W}} \left(\mathcal{L}_{\mathcal{X}}(\mathbf{y}, \hat{\mathbf{y}})\right) = \text{Arg min}_{\mathcal{W}} \left(-\frac{1}{M}\sum_{i=1}^{M}\sum_{j=1}^{C}\mathbf{y}_{ij}\log\hat{\mathbf{y}}_{ij}\right), \tag{16}$$

where $\mathcal{W}$ is the set of weights that will be optimized by the model, M denotes the number of samples in training set $\mathcal{X}$ and C is the number of action classes.

4. Experiments

We investigate the effectiveness of the proposed approach on four public benchmark action recognition datasets[4]: MSR Action3D [62], KARD [63], SBU Kinect Interaction [64], NTU-RGB+D [43] and provide comparisons with the current state-of-the-art models on each benchmark. The detailed description of each dataset is provided in Section 4.1. The implementation and training methodology are described in Section 4.2.

[3] The concatenation operation used in Eq. (15) is not viable when the size of feature maps changes.

[4] We refer the interested reader to a survey of Zhang *et al.* [34] for a full description of the current RGB-D based action recogntion datasets.

Table 1. The list of actions in three subsets **AS1**, **AS2**, and **AS3** of the MSR Action 3D dataset [62].

AS1	AS2	AS3
[a02] Horizontal arm wave	*[a01] High arm wave*	*[a06] High throw*
[a03] Hammer	*[a04] Hand catch*	*[a14] Forward kick*
[a05] Forward punch	*[a07] Draw* x	*[a15] Side kick*
[a06] High throw	*[a08] Draw tick*	*[a16] Jogging*
[a10] Hand clap	*[a09] Draw circle*	*[a17] Tennis swing*
[a13] Bend	*[a11] Two hand wave*	*[a18] Tennis serve*
[a18] Tennis serve	*[a12] Forward kick*	*[a19] Golf swing*
[a20] Pickup & Throw	*[a14] Side-boxing*	*[a20] Pickup & Throw*

4.1. Datasets and settings

MSR Action3D dataset [62]: This Kinect 1 captured dataset contains 20 actions performed by 10 subjects. Each skeleton is composed of 20 joints. The MSR Action3D [62] is challenging due to its high inter-action similarities. There are 567 action sequences in total, however, 10 sequences are not valid since the skeletons were missing. Thus, our experiments were conducted on 557 valid sequences. We follow the standard protocol proposed by Li *et al.* [62]. Specifically, the whole dataset is divided into three subsets: **AS1**, **AS2** and **AS3**. Table 1 provides a list of actions in each subset, in which all subjects with IDs 1, 3, 5, 7, 9 are selected for training and the remaining subjects with IDs 2, 4, 6, 8, 10 are used for test. Very deep neural networks such as the deep DenseNet architecture require a lot of data to train and optimize. Unfortunately, we have only 557 skeleton sequences on the MSR Action3D dataset [62]. Therefore, some data augmentation techniques, i.e. random cropping, vertical flipping, and rotation with $\alpha = 90^\circ$ have been applied on this dataset to minimize overfitting. Figure 6 illustrates three data augmentation techniques that were used in our experiments.

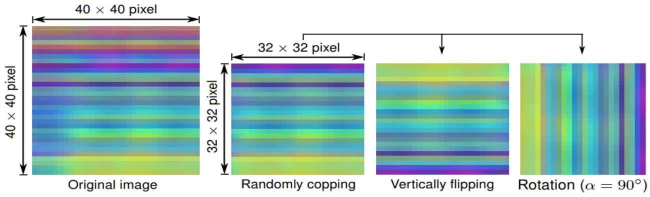


Figure 6. Illustration of data augmentation techniques that we used to generate more training samples.

Kinect Activity Recognition Dataset (KARD) [63]: The KARD [63] is a Kinect 1 dataset that contains 18 actions and 540 video sequences in total. Each action is performed three times by 10 subjects. It is composed of RGB, depth and skeleton frames in which each skeleton frame contains 15 key joints. The authors of the dataset [63] proposed to divide it into three subsets (i.e. **Action Set 1**, **Action Set 2**, and **Action Set 3**), as listed in Table 2. For each subset, three experiments have been proposed. Specifically, the first experiment (**Experiment A**) uses one-third of the dataset for training and the rest for test. Meanwhile, the second experiment (**Experiment B**) uses two-thirds of the dataset for training and the rest for test. The last experiment (**Experiment C**) uses half of the dataset for training and the other half for testing. As was the case for MSR Action3D dataset [62], data augmentation techniques (i.e. random cropping, vertically flipping, and rotation with $\alpha = 90^\circ$) were also applied.

SBU Kinect Interaction dataset [64]: This dataset was collected using the Kinect v1 sensor. It contains 282 skeleton sequences and 6822 frames performed by 7 participants. Each frame of the SBU Kinect dataset [64] contains skeleton joints of two subjects corresponding to an interaction, each skeleton has 15 key joints. There are 8 interactions in total, including *approaching*, *departing*, *pushing*, *kicking*, *punching*, *exchanging objects*, *hugging*, and *shaking hands*. This dataset is challenging due to the

Table 2. The list of actions in three subsets of the KARD dataset [63].

Action Set 1	Action Set 2	Action Set 3
Horizontal arm wave	*High arm wave*	*Draw tick*
Two-hand wave	*Side kick*	*Drink*
Bend	*Catch cap*	*Sit down*
Phone call	*Draw tick*	*Phone call*
Stand up	*Hand clap*	*Take umbrella*
Forward kick	*Forward kick*	*Toss paper*
Draw X	*Bend*	*High throw*
Walk	*Sit down*	*Horizontal arm wave*

fact that the joint coordinates exhibit low accuracy. Moreover, they contain non-periodic actions as well as very similar body movements. For instance, there are some pairs of actions that are difficult to distinguish such as *exchanging objects* – *shaking hands* or *pushing* – *punching*. We randomly split the whole dataset into 5 folds, in which 4 folds are used for training and the remaining 1 fold is used for test. It should be noted that each skeleton frame provided by the SBU dataset [64] contains two separate subjects. Therefore, we consider them as two data samples and feature computation is conducted separately for the two skeletons. Additionally, data augmentation (i.e. random cropping, vertically flipping, rotation with $\alpha = 90°$) has been also applied on the SBU dataset [64].

NTU-RGB+D dataset [43]: This Kinect 2 captured dataset is a very large-scale RGB-D dataset. To the best of our knowledge, the NTU-RGB+D dataset [43] is currently the largest and state-of-the-art benchmark dataset with skeletal data for human action analysis. It provides more than 56 thousand video samples, 4 million frames, collected from 40 distinct subjects for 60 different action classes. The following actions are provided by the NTU-RGB+D dataset [43]: *drinking, eating, brushing teeth, brushing hair, dropping, picking up, throwing, sitting down, standing up, clapping, reading, writing, tearing up paper, wearing jacket, taking off jacket, wearing a shoe, taking off a shoe, wearing on glasses, taking off glasses, putting on a hat/cap, taking off a hat/cap, cheering up, hand waving, kicking something, reaching into self pocket, hopping, jumping up, making/answering a phone call, playing with phone, typing, pointing to something, taking selfie, checking time, rubbing two hands together, bowing, shaking head, wiping face, saluting, putting palms together, crossing hands in front. sneezing/coughing, staggering, falling down, touching head, touching chest, touching back, touching neck, vomiting, fanning self. punching/slapping other person, kicking other person, pushing other person, patting others back, pointing to the other person, hugging, giving something to other person, touching other persons pocket, handshaking, walking towards each other,* and *walking apart from each other*. In the NTU-RGB+D dataset [43], each skeleton contains the 3D coordinates of 25 body joints. The authors [43] of this dataset suggested two different evaluation criteria, including **Cross-Subject** evaluation and **Cross-View** evaluation. For the Cross-Subject setting, the sequences performed by 20 subjects (with IDs 1, 2, 4, 5, 8, 9, 13, 14, 15, 16, 17, 18, 19, 25, 27, 28, 31, 34, 35, and 38) are used for training and the rest sequences are used for test. In Cross-View setting, the sequences provided by cameras 2 and 3 are used for training while sequences from camera 1 are used for test. This setting allows to evaluate the ability to recognize actions under multiple-viewpoints of the proposed skeleton-based representation. We do not apply any data augmentation technique on the NTU-RGB+D [43] due to the very large-scale nature of this dataset [43].

4.2. Implementation details

For all the datasets, the proposed Enhanced-SPMF representations are computed directly from the raw skeleton sequences without using a fixed number of frames. For computational efficiency, all the image representations are resized to 32 × 32 pixels. The three network configurations: DenseNet ($L = 100$, $k = 12$); DenseNet ($L = 250$, $k = 24$); and DenseNet ($L = 190$, $k = 40$) were implemented and evaluated in Python with the support of the Keras framework using TensorFlow as backend. During the training stage, we use mini-batches of 32 images for all networks. The weights are initialized as per

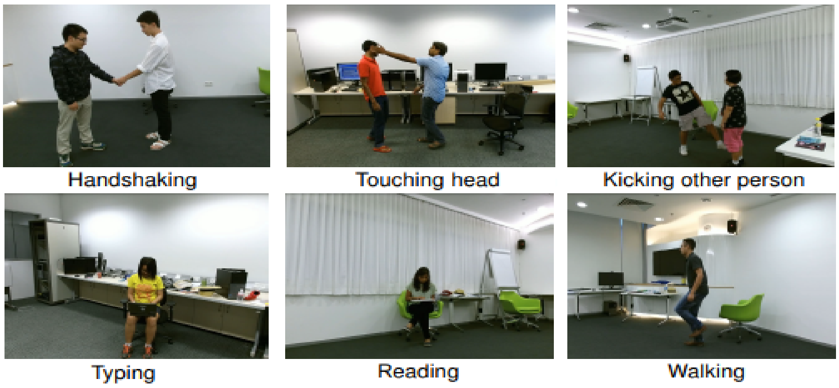


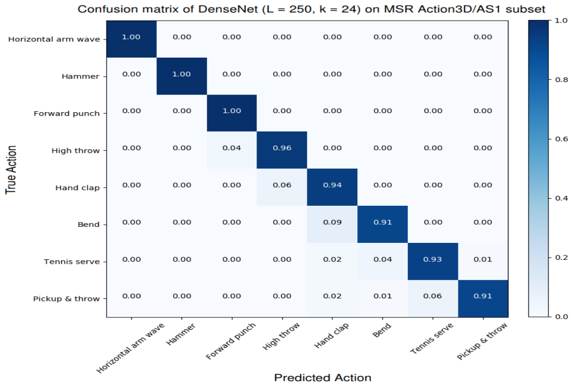
Figure 7. Some action classes of the NTU-RGB+D dataset [43]. Video samples have been captured by 3 Microsoft Kinect v2 sensors concurrently at 30 FPS. The 3D skeletal data contains the three dimensional locations of 25 major body joints, at each frame.

the He initialization technique [79]. Adam optimizer [78] is used with defaut parameters (i.e., $\beta_1 = 0.9$ and $\beta_2 = 0.999$). Additionally, we use a dynamic learning rate during training. The initial learning rate is set to 0.01 and is decreased by a factor of 0.1 after every 50 epochs. All networks are trained for 300 epochs from scratch.

5. Experimental result and analysis

5.1. Results and comparisons with the state-of-the-art

Results on MSR Action3D dataset: Experimental results and comparisons of the proposed method with the current state-of-the-art approaches on the MSR Action3D dataset [62] are summarized in Table 3. We compare the proposed method with Bag of 3D Points [62], Depth Motion Maps [80], Bi-LSTM [81], Lie Group Representation [26], FTP-SVM [81], Hierarchical LSTM [41], ST-LSTM Trust Gates [44], Graph-Based Motion [40], ST-NBNN [82], ST-NBMIM [83], S-T Pyramid [84], Ensemble TS-LSTM v2 [85] and SPMF Inception-ResNet-222 [52] using the same evaluation protocol. The proposed DenseNets (L = 100, k = 12) and DenseNet (L = 190, k = 40) achieve average accuracies of 98.76% and 98.94%, respectively. Meanwhile, the best recognition accuracies are obtained by the proposed DenseNet (L = 250, k = 24) with a total average accuracy of 99.10%. This result outperforms many previous approaches [26,40,41,44,62,80–84], demonstrating the superiority of the proposed method. Figure 9 (*first row*) shows learning curves of the proposed DenseNets on the AS1 subset/MSR Action3D dataset [62]. The recognition accuracy for each action class in the AS1 subset by the DenseNet (L = 250, k = 24) is provided in Figure 8 via its confusion matrix.



	Horizontal arm wave	Hammer	Forward punch	High throw	Hand clap	Bend	Tennis serve	Pickup & throw
Horizontal arm wave	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Hammer	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
Forward punch	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
High throw	0.00	0.00	0.04	0.96	0.00	0.00	0.00	0.00
Hand clap	0.00	0.00	0.00	0.06	0.94	0.00	0.00	0.00
Bend	0.00	0.00	0.00	0.00	0.09	0.91	0.00	0.00
Tennis serve	0.00	0.00	0.00	0.00	0.02	0.04	0.93	0.01
Pickup & throw	0.00	0.00	0.00	0.00	0.02	0.01	0.06	0.91

Figure 8. Confusion matrix of the DenseNet (L = 250, k = 24) on the MSR Action3D/AS1 dataset. Ground truth action labels are on rows and predictions by the proposed method are on columns.

Table 3. Experimental results and comparison of the proposed method with state-the-art approaches on the MSR Action3D dataset [62]. The list is ordered by recognition performance, in which results that outperform previous works are in **bold**, while the best accuracies are in **blue**.

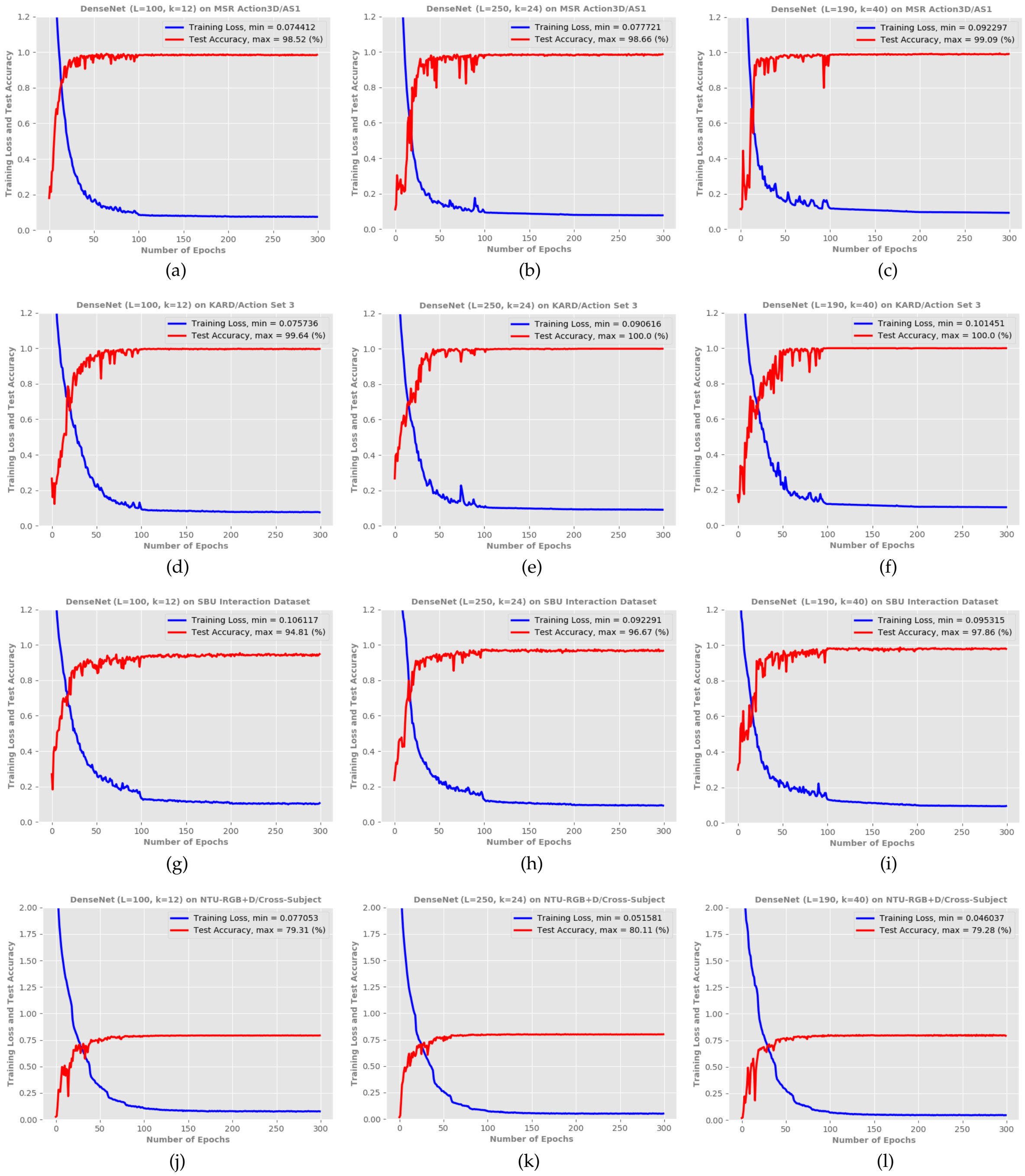
Method (protocol of [62])	**Year**	**AS1**	**AS2**	**AS3**	**Aver.**
Bag of 3D Points [62]	2010	72.90%	71.90%	71.90%	74.70%
Depth Motion Maps [80]	2016	96.20%	83.20%	92.00%	90.47%
Bi-LSTM [81]	2018	92.72%	84.93%	**97.89%**	91.84%
Lie Group Representation [26]	2014	95.29%	83.87%	98.22%	92.46%
FTP-SVM [81]	2018	95.87%	86.72%	**100.0%**	94.19%
Hierarchical LSTM [41]	2015	99.33%	94.64%	95.50%	94.49%
ST-LSTM Trust Gates [44]	2016	N/A	N/A	N/A	94.80%
Graph-Based Motion [40]	2016	93.60%	95.50%	95.10%	94.80%
ST-NBNN [82]	2017	91.50%	95.60%	97.30%	94.80%
ST-NBMIM [83]	2018	92.50%	95.60%	98.20%	95.30%
S-T Pyramid [84]	2015	99.10%	92.90%	96.40%	96.10%
Ensemble TS-LSTM v2 [85]	2017	95.24%	96.43%	**100.0%**	97.22%
SPMF Inception-ResNet-222 [52]	2018	97.54%	98.73%	99.41%	98.56%
Enhanced-SPMF DenseNet (L = 100, k = 12) **(ours)**	2018	**98.52%**	**98.66%**	99.09%	**98.76%**
Enhanced-SPMF DenseNet (L = 250, k = 24) **(ours)**	2018	**98.83%**	**99.06%**	99.40%	**99.10%**
Enhanced-SPMF DenseNet (L = 190, k = 40) **(ours)**	2018	**98.60%**	**98.87%**	99.36%	**98.94%**

Results on KARD dataset: We performed a total of 9 experiments over three experiments A, B, and C on the KARD dataset [63]. Table 4 summarizes the obtained results on this dataset. We compute the average recognition accuracy over the three experiments and compare it with existing techniques including Hand-crafted Features [63], Posture Feature+Multi-class SVM [86], and Key Postures+Multi-class SVM [87]. As can be seen in Table 4, the proposed DenseNet (L = 250, k = 24) is able to improve state-of-the-art accuracy by 9.15% over Hand-crafted Features [63], 2.78% over Posture Feature+Multi-class SVM [86] and 0.68% over Key Postures+Multi-class SVM [87]. This result confirms that the proposed deep learning framework trained on the Enhanced-SPMFs is able to achieve better performance in the recognition of actions compared to hand-crafted based approaches.

Table 4. Average recognition accuracies (%) over three experiments A, B, and C and comparison with previous works on the KARD dataset [63]. The best accuracies are in **blue**. Results that surpass previous works are in **bold**.

Method (protocol of [63])	**Year**	**Acc. (%)**
Hand-crafted Features [63]	2015	90.83%
Posture Feature+Multi-class SVM [86]	2016	97.20%
Key Postures+Multi-class SVM [87]	2016	99.30%
Enhanced-SPMF DenseNet (L = 100, k = 12) **(ours)**	2018	**99.74%**
Enhanced-SPMF DenseNet (L = 250, k = 24) **(ours)**	2018	**99.98%**
Enhanced-SPMF DenseNet (L = 190, k = 40) **(ours)**	2018	**99.88%**

Results on SBU Kinect Interaction dataset: As reported in Table 5, the proposed DenseNet (L = 250, k = 40) achieved an accuracy of 97.86% and outperforms many existing state-of-the-art approaches including Raw Skeleton [64], Joint Features [64], HBRNN [41], CHARM [88], Deep LSTM [45], Joint Features [89], ST-LSTM [44], Co-occurrence+Deep LSTM [45], STA-LSTM [90], ST-LSTM+Trust Gates [44], ST-NBMIM [83], Clips+CNN+MTLN [91], Two-stream RNN [92], and GCA-LSTM network [93]. Using only skeleton modality, the proposed method outperforms hand-crafted feature based approaches such as Raw Skeleton [64], Joint Features [64] and recent state-of-the-art RNN-based approaches [41,44,45,90,92,93]. In particular, the proposed method achieves a significant accuracy gain of 2.96% compared to the nearest competitor GCA-LSTM network [93]. This result demonstrates that the proposed deep learning framework is able to learn discriminative spatio-temporal features of skeleton joints containing in the proposed motion representation for classification task.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Figure 9. Training curves on the MSR Action3D [62], KARD [63], SBU Kinect Interaction [64], and NTU-RGB+D [43] datasets. Almost all designed networks are able to reach the optimal weights after the first 100 epochs.

Table 5. Action recognition accuracies (%) and comparison with previous works on the SBU Kinect Interaction dataset [64]. The best accuracies are in **blue**. Results that surpass previous works are in **bold**.

Method (protocol of [64])	**Year**	**Acc. (%)**
Raw Skeleton [64]	2012	49.70%
Joint Features [64]	2012	80.30%
HBRNN [41] (reported in [92])	2015	80.40%
CHARM [88]	2015	83.90%
Deep LSTM [45]	2017	86.03%
Joint Features [89]	2014	86.90%
ST-LSTM [44]	2016	88.60%
Co-occurrence+Deep LSTM [45]	2018	90.41%
STA-LSTM [90]	2017	91.51%
ST-LSTM+Trust Gates [44]	2018	93.30%
ST-NBMIM [83]	2018	93.30%
Clips+CNN+MTLN [91]	2017	93.57%
Two-stream RNN [92]	2017	94.80%
GCA-LSTM network [93]	2018	94.90%
Enhanced-SPMF DenseNet (L = 100, k = 12) (**ours**)	2018	**94.81%**†
Enhanced-SPMF DenseNet (L = 250,k = 24) (**ours**)	2018	**96.67%**
Enhanced-SPMF DenseNet (L = 190, k = 40) (**ours**)	2018	**97.86%**

Results on NTU-RGB+D dataset: For the NTU-RGB+D dataset [43], the best configuration DenseNet (L = 250, k = 40) achieves an accuracy of 80.11% on the Cross-Subject evaluation and 86.82% on the Cross-View evaluation, as summarized in Table 6. These results demonstrate the effectiveness of the proposed representation and deep learning framework since they surpass previous state-of-the-art techniques such as Lie Group Representation [26], Hierarchical RNN [41], Dynamic Skeletons [94], Two-Layer P-LSTM [43], ST-LSTM Trust Gates [44], Geometric Features [72], Two-Stream RNN [92], Enhanced Skeleton [95], Lie Group Skeleton+CNN [96], GCA-LSTM [93] and SPMF Inception-ResNet-222 [52]. With a high recognition rate on the Cross-View evaluation (86.82%) where the sequences provided by cameras 2 and 3 are used for training and sequences from camera 1 are used for test, the proposed method shows its effectiveness for dealing with view-independent human action recognition problem. Figure 9 (*last row*) shows the training loss and test accuracy of the DenseNet (L = 250, k = 24) on this dataset.

5.2. Analysis of the effect of the AHE algorithm on SPMF

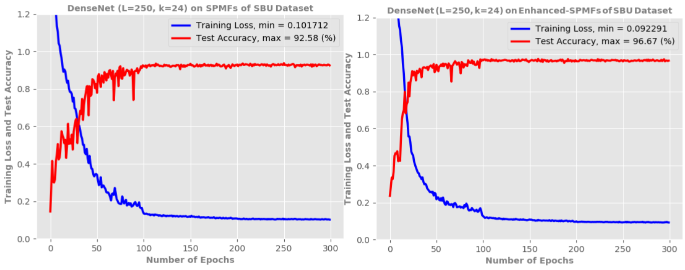
We believe that the use of the AHE algorithm [53] helps the proposed representation to be more discriminative, which improves recognition accuracy. To verify this hypothesis, we carried out additional experiments on the SBU Kinect Interaction dataset [64]. Specifically, we trained the proposed DenseNet ($L = 250, k = 24$) on both the SPMFs and Enhanced-SPMFs. During training, the same hyper-parameters and training methodology were applied. The experimental results indicate that the proposed deep network achieves better recognition accuracy when trained on the Enhanced-SPMFs. As reported in Figure 10, applying the AHE algorithm [53] helps improving the accuracy by 4.09%. This result validates our hypothesis above.

5.3. Visualization of deep feature maps

Different action classes have different discriminative characteristics. To better understand the internal operation of the proposed deep networks and to study what they learned from the skeleton-based representation, we input different Enhanced-SPMFs corresponding to different action classes of the MSR Action3D dataset [62] to the DenseNet ($L = 100$, $k = 12$) and visualize the individual feature maps learned by the network at the end of a dense block (intermediate layer). We observe that the designed network is able to extract discriminative features from the Enhanced-SPMF

Table 6. Experimental results and comparison of the proposed method with previous approaches on the NTU-RGB+D dataset [43]. The best accuracies are in **blue**. Results that surpass previous works are in **bold**.

Method (protocol of [43])	**Year**	**Cross-Subject**	**Cross-View**
Lie Group Representation [26]	2014	50.10%	52.80%
Hierarchical RNN [41]	2016	59.07%	63.97%
Dynamic Skeletons [94]	2015	60.20%	65.20%
Two-Layer P-LSTM [43]	2016	62.93%	70.27%
ST-LSTM Trust Gates [44]	2016	69.20%	77.70%
Geometric Features [72]	2017	70.26%	82.39%
Two-Stream RNN [92]	2017	71.30%	79.50%
Enhanced Skeleton [95]	2017	75.97%	82.56%
Lie Group Skeleton+CNN [96]	2017	75.20%	83.10%
GCA-LSTM [93]	2018	76.10%	84.00%
SPMF Inception-ResNet-222 [52]	2018	**78.89%**	**86.15%**
Enhanced-SPMF DenseNet (L = 100, k = 12) **(ours)**	2018	**79.31%**	**86.64%**
Enhanced-SPMF DenseNet (L = 250, k = 24) **(ours)**	2018	**80.11%**	**86.82%**
Enhanced-SPMF DenseNet (L = 190, k = 40) **(ours)**	2018	**79.28%**	**86.68%**



(a) (b)

Figure 10. Training loss and test accuracy of the proposed DenseNet ($L = 100$, $k = 12$) on the SBU dataset [64]. Figure 10a shows the obtained result when trained on SPMFs, while Figure 10b reports the obtained result when trained on Enhanced-SPMFs.

representations. This is expressed through the color of each learned feature map, as can be seen in Figure 11. These discriminative features play a key role in classifying actions.

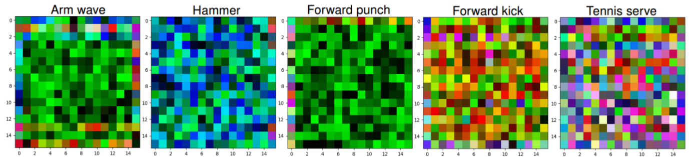


Figure 11. Visualization of feature maps learned by the proposed DenseNet ($L = 100$, $k = 12$) from several samples of the MSR Action3D dataset [62]. Best viewed in color.

5.4. Computational efficiency evaluation

In this section, we take the AS1 subset of MSR Action3D dataset [62] and the DenseNet (L = 100, k = 12) to evaluate the computational efficiency of the proposed method. Figure 12 illustrates three main stages of the deep learning framework for learning and recognizing actions from skeleton sequences, including an encoding process from input skeleton sequences to color images (**Stage 1**); a supervised training stage (**Stage 2**); and an inference stage (**Stage 3**). With the implementation in Python using

Keras and training on a single GeForce GTX 1080 Ti GPU[5], the proposed deep network that only has 6.0M parameters takes less than six hours to reach convergence. During this stage, it take 0.164 seconds per skeleton sequence. Latency required to predict a new skeleton sequence using the pre-trained model, including the **Stage 1** that is executed on a CPU and the **Stage 3** is about 74.8×10^{-3} seconds per sequence. Additionally, it should be noted that the computation of the Enhanced-SPMFs can be implemented and optimized on a GPU for real-time applications. This result verifies the effectiveness of the proposed learning framework in terms of computational cost.

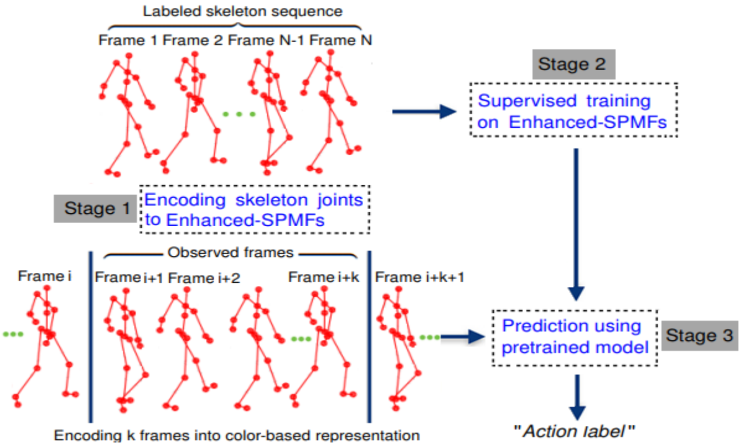

Labeled skeleton sequence
Frame 1 Frame 2 Frame N-1 Frame N
Stage 2
Supervised training on Enhanced-SPMFs
Stage 1
Encoding skeleton joints to Enhanced-SPMFs
Observed frames
Frame i
Frame i+1 Frame i+2
Frame i+k
Frame i+k+1
Prediction using pretrained model
Stage 3
Encoding k frames into color-based representation
"Action label"

Figure 12. Three main stages of the proposed deep learning framework for recognizing human actions from skeleton sequences.

Table 7. Execution time of each stage of the proposed deep learning framework.

Stage	Average processing time (second/sequence)	
1	20.8×10^{-3}	(Intel Core i7 3.2GHz CPU)
2	0.164	(GTX 1080 Ti GPU)
3	74.8×10^{-3}	(CPU + GPU time)

5.5. Limitations

The use of the Savitzky-Golay filter [73] helps reduce the effect of noise on the raw skeleton sequences. However, the proposed approach cannot overcome the problem of missing data. In other words, as the Enhanced-SPMF is a global representation for the whole skeleton sequence, data error of local fragments in the input sequences could cut down the recognition rate. Another open problem of the proposed approach is how to scope with Online Action Recognition (OAR) task. Specifically, how to detect and recognize human actions from unsegmented streams in a continuous manner, where boundaries between different kinds of actions within the stream are unknown. A common solution for OAR is the sliding window based methods [97,98]. These approaches consider the temporal coherence within the window for prediction. We can also apply this idea to solve the current problem. E.g., during the online inference phase, we use a sliding window on the original skeleton sequences or on image-coded representations (i.e. Enhanced-SPMFs) and then predicting action by pretrained deep learning model, as we showed in Figure 12 (**Stage 3**). However, we understand that the performance of this approach is sensitive to the window size. Either too large or too small window size could lead to a

[5] Details about the GPU specification are available at https://www.nvidia.com/en-us/geforce/products/10series/geforce-gtx-1080-ti/.

significant drop in recognition performance. Another solution is to use Temporal Attention Networks (TANs) [99–102] that incorporates temporal attention model for video-based action recognition.

6. Conclusion

In this paper, we present an efficient and effective deep learning framework for 3D human action recognition from skeleton sequences. A novel motion representation, termed as Enhanced-SPMF, which captures the spatio-temporal information of skeleton movements and transforms them into color images has been proposed. Different Deep Convolutional Neural Networks (D-CNNs) based on the DenseNet architecture have been designed and optimized to learn and recognize actions from the proposed representation, in an end-to-end manner. We exploited the Adaptive Histogram Equalization (AHE) technique to enhance the local textures of color images and generate more discriminative features for learning and classification tasks. Extensive empirical evaluations on four challenging public datasets demonstrate the effectiveness of the proposed approach on both individual actions, interactions, multiview and large-scale datasets. In particular, we also indicate that the proposed method is invariant to viewpoint changes and requires low computational cost for training and inference. We hope that this study opens up a new door to exploit the big potential of skeletal data, which helps to address the current challenges in building real-world action recognition applications.

Author Contributions: Conceptualization, methodology, software, validation, investigation, resources, visualization, writing–original draft preparation, H.H.P; Data curation, formal analysis, writing–review and editing, all authors; Supervision; project administration; funding acquisition, A.C., P.Z., L.K and S.A.V.

Funding: Sergio A. Velastin is grateful for funding received from the Universidad Carlos III de Madrid, the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement N° 600371, el Ministerio de Economía, Industria y Competitividad (COFUND2013-51509) el Ministerio de Educación, Cultura y Deporte (CEI-15-17) and Banco Santander.

Acknowledgments: This research was carried out at the Cerema Research Center and Informatics Research Institute of Toulouse, Paul Sabatier University, France. The authors would like to express our thanks to all the people who have made helpful comments and suggestions on a previous draft.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A Savitzky-Golay Smoothing Filter

Savitzky-Golay (S-G) filter is a *low-pass* filter based on local least-squares polynomial approximation that is often used to smooth noisy data. The 3D skeleton joints obtained from depth cameras can be considered as a series of equally spaced data in the time domain, applying S-G filter on raw skeletal data helps reduce the level of noise while maintaining the 3D geometric characteristics of the input sequences.

Considering a sequence of $N = 2M + 1$ input data points $x[n]$ centered at $n = 0$, denoted as

$$\mathbf{x} = [x_{-M}, ..., x_{-1}, x_0, x_1, ..., x_M]^T. \tag{A1}$$

The N data samples of $\mathbf{x}$ can be fitted by a polynomial

$$p(n) = \sum_{k=0}^{N} c_k n^k. \tag{A2}$$

To best fit the given data $\mathbf{x}$, Savitzky and Golay [73] proposed a method of data smoothing by finding the vector of polynomial coefficients $\mathbf{c} = [c_0, c_1, ..., c_N]^T$ that minimize the mean-squares approximation error

$$\mathcal{E}_N = \sum_{n=-M}^{M} \left(\sum_{k=0}^{N} c_k n^k - x[n] \right)^2. \tag{A3}$$

To this end, one solution is to determine a set of coefficients that satisfies the partial derivative equation is equal to zero

$$\frac{\partial \mathcal{E}_N}{\partial a_i} = \sum_{n=-M}^{M} 2n^i \left(\sum_{k=0}^{N} c_k n^k - x[n] \right) = 0 \text{ with } i = 0, 1, ..., N. \tag{A4}$$

Eq. (A4) is equivalent to

$$\sum_{k=0}^{N} \left(\sum_{n=-M}^{M} n^{i+k} \right) c_k = \sum_{n=-M}^{M} n^i x[n]. \tag{A5}$$

Defining a matrix $\mathbf{A} = \{\alpha_{n,i}\}$ as the matrix with elements

$$\alpha_{n,i} = n^i \tag{A6}$$

where $-M \leq n \leq M$ and $i = 0, 1, ..., N$. The matrix **A** is called the design matrix for the polynomial approximation problem. Note that, the transpose of **A** is $\mathbf{A}^T = \{\alpha_{i,n}\}$ and the product matrix $\mathbf{B} = \mathbf{A}^T\mathbf{A}$ is a symmetric matrix with elements

$$\beta_{i,k} = \sum_{n=-M}^{M} \alpha_{i,n}\alpha_{n,k} = \sum_{n=-M}^{M} n^{i+k} \tag{A7}$$

Therefore, Eq. (A5) can be rewritten in matrix form as

$$\mathbf{B} \cdot \mathbf{c} = \mathbf{A}^T \cdot \mathbf{A} \cdot \mathbf{c} = \mathbf{A}^T \cdot \mathbf{x}. \tag{A8}$$

The polynomial coefficients can be determined as

$$\mathbf{c} = (\mathbf{A}^T \cdot \mathbf{A})^{-1} \cdot (\mathbf{A}^T \cdot \mathbf{x}\ ). \tag{A9}$$

For example, for smoothing by a 5-point quadratic polynomial with $N = 5, M = -2, -1, 0, 1, 2$, the the tth filtering result, y_t is given by

$$y_t = \frac{-3x_{t-2} + 12x_{t-1} + 17x_t + 12x_{t+1} - 3x_{t+2}}{35}. \tag{A10}$$

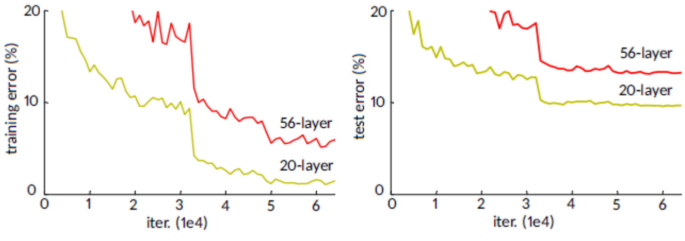
The Eq. (A10) above was used in our experiments to reduce the effect of noises on the raw skeleton data.

Appendix B Degradation phenomenon in training very deep neural networks

Very deep neural networks demonstrate to have a high performance on many visual-related tasks [57–60]. However, they are very difficult to optimize. One of the main challenges for training deeper networks is the vanishing and exploding gradient problems [103]. Specifically, when the network is deep enough, the supervision signals from the output layer can be completely attenuated or exploded on their way back towards the previous layers. Therefore, the network cannot learn the parameters effectively. These obstacles can be solved by recent advanced techniques in deep learning such as Normalized Initialization [104] or Batch Normalization [76]. When the deep networks start converging, a degradation phenomenon occurs. Due to this, the training and test errors increase if more layers are added to a deep architecture. This phenomenon is called by the degradation phenomenon. The following figures show an experimental result [59] related to this phenomenon.

References

1. Aggarwal, J.K.; Ryoo, M.S. Human activity analysis: A review. *ACM Computing Surveys* **2011**, *43*, 16.



(a) (b)

Figure A1. Degradation phenomenon during training D-CNNs. (**a**) Training error and (**b**) test error on CIFAR-10 [105] with 20-layer and 56-layer CNNs reported by He *et al.* [59]. The deeper network has higher error for both training and test phases.

2. Boiman, O.; Irani, M. Detecting irregularities in images and in video. *International Journal of Computer Vision* **2007**, *74*, 17–31.
3. Lin, W.; Sun, M.T.; Poovandran, R.; Zhang, Z. Human activity recognition for video surveillance. IEEE International Symposium on Circuits and Systems (ISCAS), 2008, pp. 2737–2740.
4. Gupta, A.; Kembhavi, A.; Davis, L.S. Observing human-object interactions: Using spatial and functional compatibility for recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2009**, *31*, 1775–1789.
5. Yao, B.; Fei-Fei, L. Recognizing human-object interactions in still images by modeling the mutual context of objects and human poses. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2012**, *34*, 1691–1703.
6. Dagli, I.; Brost, M.; Breuel, G. Action recognition and prediction for driver assistance systems using dynamic belief networks. International Conference on Object-Oriented and Internet-Based Technologies, Concepts, and Applications for a Networked World, 2002, pp. 179–194.
7. Fridman, L.; Brown, D.E.; Glazer, M.; Angell, W.; Dodd, S.; Jenik, B.; Terwilliger, J.; Kindelsberger, J.; Ding, L.; Seaman, S.; others. MIT Autonomous Vehicle Technology Study: Large-Scale Deep Learning Based Analysis of Driver Behavior and Interaction with Automation. *arXiv preprint arXiv:1711.06976* **2017**.
8. Poppe, R. A survey on vision-based human action recognition. *Image and Vision Computing* **2010**, *28*, 976–990.
9. Weinland, D.; Ronfard, R.; Boyer, E. A survey of vision-based methods for action representation, segmentation and recognition. *Computer Vision and Image Understanding* **2011**, *115*, 224–241.
10. Lowe, D.G. Object recognition from local scale-invariant features. IEEE International Conference on Computer Vision (ICCV), 1999, Vol. 2, pp. 1150–1157.
11. Lowe, D. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* **2004**, *60*, 91–110.
12. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2005, Vol. 1, pp. 886–893.
13. Laptev, I.; Marszalek, M.; Schmid, C.; Rozenfeld, B. Learning realistic human actions from movies. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2008, pp. 1–8.
14. Klaser, A.; Marszałek, M.; Schmid, C. A spatio-temporal descriptor based on 3D-gradients. British Machine Vision Conference (BMVC), 2008, pp. 275–1.
15. Dollár, P.; Rabaud, V.; Cottrell, G.; Belongie, S. Behavior recognition via sparse spatio-temporal features. IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS), 2005, pp. 65–72.
16. Bay, H.; Tuytelaars, T.; Van Gool, L. SURF: Speeded up robust features. European Conference on Computer Vision (ECCV), 2006, pp. 404–417.
17. Willems, G.; Tuytelaars, T.; Van Gool, L. An efficient dense and scale-invariant spatio-temporal interest point detector. European Conference on Computer Vision (ECCV), 2008, pp. 650–663.
18. Zhang, Z. Microsoft Kinect sensor and its effect. *IEEE Multimedia* **2012**, *19*, 4–10.
19. ASUS. The ASUS Xtion PRO Depth Sensor. https://www.asus.com/3D-Sensor/Xtion_PRO/. Accessed: 2018-04-06.

20. Intel. The Intel RealSense Depth Camera PRO Depth Sensor. https://software.intel.com/en-us/realsense/d400. Accessed: 2018-04-06.
21. Orbbec. Orbbec Astra, Astra S & Astra Pro. https://orbbec3d.com/product-astra/. Accessed: 2018-04-06.
22. Wang, J.; Liu, Z.; Wu, Y.; Yuan, J. Mining actionlet ensemble for action recognition with depth cameras. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012, pp. 1290–1297.
23. Oreifej, O.; Liu, Z. HON4D: Histogram of oriented 4D normals for activity recognition from depth sequences. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013, pp. 716–723.
24. Xia, L.; Aggarwal, J. Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013, pp. 2834–2841.
25. Rahmani, H.; Mahmood, A.; Huynh, D.Q.; Mian, A. HOPC: Histogram of oriented principal components of 3D pointclouds for action recognition. European Conference on Computer Vision (ECCV), 2014, pp. 742–757.
26. Vemulapalli, R.; Arrate, F.; Chellappa, R. Human action recognition by representing 3D skeletons as points in a lie group. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 588–595.
27. Wang, J.; Liu, Z.; Wu, Y. Learning actionlet ensemble for 3D human action recognition. In *Human Action Recognition with Depth Cameras*; Springer, 2014; pp. 11–40.
28. Yang, X.; Tian, Y. Super normal vector for human activity recognition with depth cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2017**, *39*, 1028–1039.
29. Shotton, J.; Fitzgibbon, A.; Cook, M.; Sharp, T.; Finocchio, M.; Moore, R.; Kipman, A.; Blake, A. Real-time human pose recognition in parts from single depth images. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011, pp. 1297–1304.
30. Ye, M.; Yang, R. Real-time simultaneous pose and shape estimation for articulated objects using a single depth camera. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 2345–2352.
31. Johansson, G. Visual motion perception. *Scientific American* **1975**, *232*, 76–89.
32. Bearman, A.; Dong, C. Human pose estimation and activity classification using convolutional neural networks. *CS231n Course Project Reports* **2015**.
33. Gu, J.; Ding, X.; Wang, S.; Wu, Y. Action and gait recognition from recovered 3D human joints. *IEEE Transactions on Systems, Man, and Cybernetics* **2010**, *40*, 1021–1033.
34. Zhang, J.; Li, W.; Ogunbona, P.O.; Wang, P.; Tang, C. RGB-D-based action recognition datasets: A survey. *Pattern Recognition* **2016**, *60*, 86–105.
35. Xia, L.; Chen, C.C.; Aggarwal, J. View invariant human action recognition using histograms of 3D joints. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012, pp. 20–27.
36. Chaudhry, R.; Ofli, F.; Kurillo, G.; Bajcsy, R.; Vidal, R. Bio-inspired dynamic 3D discriminative skeletal features for human action recognition. IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 471–478.
37. Ding, W.; Liu, K.; Fu, X.; Cheng, F. Profile HMMs for skeleton-based human action recognition. *Signal Processing: Image Communication* **2016**, *42*, 109–119.
38. Han, L.; Wu, X.; Liang, W.; Hou, G.; Jia, Y. Discriminative human action recognition in the learned hierarchical manifold space. *Image and Vision Computing* **2010**, *28*, 836–849.
39. Luo, J.; Wang, W.; Qi, H. Group sparsity and geometry constrained dictionary learning for action recognition from depth maps. IEEE International Conference on Computer Vision (ICCV), 2013, pp. 1809–1816.
40. Wang, P.; Yuan, C.; Hu, W.; Li, B.; Zhang, Y. Graph based skeleton motion representation and similarity measurement for action recognition. European Conference on Computer Vision (ECCV), 2016, pp. 370–385.
41. Du, Y.; Wang, W.; Wang, L. Hierarchical recurrent neural network for skeleton based action recognition. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1110–1118.
42. Veeriah, V.; Zhuang, N.; Qi, G.J. Differential recurrent neural networks for action recognition. IEEE International Conference on Computer Vision (ICCV), 2015, pp. 4041–4049.
43. Shahroudy, A.; Liu, J.; Ng, T.T.; Wang, G. NTU RGB+D: A large scale dataset for 3D human activity analysis. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 1010–1019.
44. Liu, J.; Shahroudy, A.; Xu, D.; Wang, G. Spatio-temporal LSTM with trust gates for 3D human action recognition. European Conference on Computer Vision (ECCV), 2016, pp. 816–833.

45. Zhu, W.; Lan, C.; Xing, J.; Zeng, W.; Li, Y.; Shen, L.; Xie, X.; others. Co-occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks. Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI), 2016, p. 8.
46. Liu, J.; Wang, G.; Hu, P.; Duan, L.Y.; Kot, A.C. Global context-aware attention LSTM networks for 3D action recognition. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 3671–3680.
47. Lv, F.; Nevatia, R. Recognition and segmentation of 3D human action using HMM and multi-class Adaboost. *European Conference on Computer Vision (ECCV)* **2006**, pp. 359–372.
48. Schuster, M.; Paliwal, K.K. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing* **1997**, *45*, 2673–2681.
49. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Computation* **1997**, *9*, 1735–1780.
50. Graves, A.; Fernández, S.; Schmidhuber, J. Bidirectional LSTM networks for improved phoneme classification and recognition. International Conference on Artificial Neural Networks (ICANN), 2005, pp. 799–804.
51. Sainath, T.N.; Vinyals, O.; Senior, A.; Sak, H. Convolutional, Long Short-Term Memory, Fully Connected Deep Neural Networks. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2015, pp. 4580–4584.
52. Hieu Pham, H.; Khoudour, L.; Crouzil, A.; Zegers, P.; Velastin, S. Skeletal movement to color map: A novel representation for 3D action recognition with Inception Residual networks. IEEE International Conference on Image Processing (ICIP), 2018, pp. 3483–3487.
53. Pizer, S.M.; Amburn, E.P.; Austin, J.D.; Cromartie, R.; Geselowitz, A.; Greer, T.; ter Haar Romeny, B.; Zimmerman, J.B.; Zuiderveld, K. Adaptive histogram equalization and its variations. *Computer Vision, Graphics, and Image Processing* **1987**, *39*, 355–368.
54. Huang, G.; Liu, Z.; Weinberger, K.Q.; van der Maaten, L. Densely connected convolutional networks. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, Vol. 1, p. 3.
55. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems (NIPS), 2012, pp. 1097–1105.
56. Karpathy, A.; Toderici, G.; Shetty, S.; Leung, T.; Sukthankar, R.; Fei-Fei, L. Large-scale video classification with convolutional neural networks. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 1725–1732.
57. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* **2014**.
58. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1–9.
59. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.
60. Telgarsky, M. Benefits of depth in neural networks. *arXiv preprint arXiv:1602.04485* **2016**.
61. He, K.; Sun, J. Convolutional neural networks at constrained time cost. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 5353–5360.
62. Li, W.; Zhang, Z.; Liu, Z. Action recognition based on a bag of 3D points. IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPR), 2010, pp. 9–14.
63. Gaglio, S.; Re, G.L.; Morana, M. Human Activity Recognition Process Using 3-D Posture Data. *IEEE Trans. Human-Machine Systems* **2015**, *45*, 586–597.
64. Yun, K.; Honorio, J.; Chattopadhyay, D.; Berg, T.L.; Samaras, D. Two-person interaction detection using body-pose features and multiple instance learning. IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPR), 2012, pp. 28–35.
65. Han, F.; Reily, B.; Hoff, W.; Zhang, H. Space-time representation of people based on 3D skeletal data: A review. *Computer Vision and Image Understanding* **2017**, *158*, 85–105.
66. Müller, M. Dynamic time warping. *Information retrieval for music and motion* **2007**, pp. 69–84.
67. Eddy, S.R. Hidden Markov Models. *Current opinion in structural biology* **1996**, *6*, 361–365.
68. Kirk, A.G.; O'Brien, J.F.; Forsyth, D.A. Skeletal parameter estimation from optical motion capture data. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2005, Vol. 2, pp. 782–788.

69. Cao, Z.; Simon, T.; Wei, S.E.; Sheikh, Y. Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. IEEE Computer Vision and Pattern Recognition (CVPR), 2017.
70. Graves, A. Supervised sequence labelling with recurrent neural networks. Studies in Computational Intelligence, 2012, Vol. 385.
71. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436.
72. Zhang, S.; Liu, X.; Xiao, J. On geometric features for skeleton-based action recognition using multilayer LSTM networks. IEEE Winter Conference on Applications of Computer Vision (WACV), 2017, pp. 148–157.
73. Savitzky, A.; Golay, M.J. Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry* **1964**, *36*, 1627–1639.
74. Glorot, X.; Bordes, A.; Bengio, Y. Deep sparse rectifier neural networks. Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS), 2011, pp. 315–323.
75. Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* **1998**, *86*, 2278–2324.
76. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. International Conference on Machine Learning (ICML), 2015, pp. 448–456.
77. Clevert, D.A.; Unterthiner, T.; Hochreiter, S. Fast and accurate deep network learning by Exponential Linear Units (ELUs). *arXiv preprint arXiv:1511.07289* **2015**.
78. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* **2014**.
79. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1026–1034.
80. Chen, C.; Liu, K.; Kehtarnavaz, N. Real-time human action recognition based on depth motion maps. *Journal of Real-Time Image Processing* **2016**, *12*, 155–163.
81. Tanfous, A.B.; Drira, H.; Amor, B.B. Coding Kendall's shape trajectories for 3D action recognition. IEEE Computer Vision and Pattern Recognition (CVPR), 2018.
82. Weng, J.; Weng, C.; Yuan, J. Spatio-temporal Naive-Bayes Nearest-Neighbor (ST-NBNN) for skeleton-based action recognition. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 4171–4180.
83. Weng, J.; Weng, C.; Yuan, J.; Liu, Z. Discriminative spatio-temporal pattern discovery for 3D action recognition. *IEEE Transactions on Circuits and Systems for Video Technology* **2018**, pp. 1–1.
84. Xu, H.; Chen, E.; Liang, C.; Qi, L.; Guan, L. Spatio-temporal pyramid model based on depth maps for action recognition. IEEE International Workshop on Multimedia Signal Processing (MMSP), 2015, pp. 1–6.
85. Li, C.; Wang, P.; Wang, S.; Hou, Y.; Li, W. Skeleton-based action recognition using LSTM and CNN. IEEE International Conference on Multimedia & Expo Workshops (ICMEW), 2017, pp. 585–590.
86. Cippitelli, E.; Gasparrini, S.; Gambi, E.; Spinsante, S. A human activity recognition system using skeleton data from RGB-D sensors. *Computational Intelligence and Neuroscience* **2016**, p. 21.
87. Ling, J.; Tian, L.; Li, C. 3D human activity recognition using skeletal data from RGB-D sensors. International Symposium on Visual Computing (ISVC), 2016, pp. 133–142.
88. Li, W.; Wen, L.; Choo Chuah, M.; Lyu, S. Category-blind human action recognition: A practical recognition system. IEEE International Conference on Computer Vision (ICCV), 2015, pp. 4444–4452.
89. Ji, Y.; Ye, G.; Cheng, H. Interactive body part contrast mining for human interaction recognition. IEEE International Conference on Multimedia and Expo Workshops (ICMEW), 2014, pp. 1–6.
90. Song, S.; Lan, C.; Xing, J.; Zeng, W.; Liu, J. An end-to-end spatio-temporal attention model for human action recognition from skeleton data. AAAI Conference on Artificial Intelligence (AAAI), 2017, Vol. 1, pp. 4263–4270.
91. Ke, Q.; Bennamoun, M.; An, S.; Sohel, F.; Boussaid, F. A new representation of skeleton sequences for 3D action recognition. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 4570–4579.
92. Wang, H.; Wang, L. Modeling temporal dynamics and spatial configurations of actions using two-stream recurrent neural networks. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 3633–3642.
93. Liu, J.; Wang, G.; Duan, L.Y.; Abdiyeva, K.; Kot, A.C. Skeleton-based human action recognition with global context-aware attention LSTM networks. *IEEE Transactions on Image Processing* **2018**, *27*, 1586–1599.

94. Hu, J.; Zheng, W.S.; Lai, J.H.; Jianguo, Z. Jointly learning heterogeneous features for RGB-D activity recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2015**, *39*.
95. Liu, M.; Liu, H.; Chen, C. Enhanced skeleton visualization for view invariant human action recognition. *Pattern Recognition* **2017**, *68*, 346–362.
96. Rahmani, H.; Bennamoun, M. Learning action recognition model from depth and skeleton videos. *IEEE International Conference on Computer Vision (ICCV)* **2017**, pp. 5832–5841.
97. Kulkarni, K.; Evangelidis, G.; Cech, J.; Horaud, R. Continuous action recognition based on sequence alignment. *International Journal of Computer Vision* **2015**, *112*, 90–114.
98. Kviatkovsky, I.; Rivlin, E.; Shimshoni, I. Online action recognition using covariance of shape and motion. *Computer Vision and Image Understanding* **2014**, *129*, 15–26.
99. Mnih, V.; Heess, N.; Graves, A.; others. Recurrent models of visual attention. Advances in Neural Information Processing Systems, 2014, pp. 2204–2212.
100. Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhudinov, R.; Zemel, R.; Bengio, Y. Show, attend and tell: Neural image caption generation with visual attention. International Conference on Machine Learning (ICML), 2015, pp. 2048–2057.
101. Luong, M.T.; Pham, H.; Manning, C.D. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025* **2015**.
102. Zang, J.; Wang, L.; Liu, Z.; Zhang, Q.; Hua, G.; Zheng, N. Attention-based temporal weighted convolutional neural network for action recognition. International Conference on Artificial Intelligence Applications and Innovations (IFIP), 2018, pp. 97–108.
103. Glorot, X.; Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. International Conference on Artificial Intelligence and Statistics (AISTATS), 2010, pp. 249–256.
104. LeCun, Y.; Bottou, L.; Orr, G.B.; Müller, K.R. Efficient backprop. In *Neural Networks: Tricks of the trade*; Springer, 1998; pp. 9–50.
105. Krizhevsky, A.; Hinton, G. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.