IET Computer Vision

Research Article

IET Journals
The Institution of Engineering and Technology

Learning to recognise 3D human action from a new skeleton-based representation using deep convolutional neural networks

ISSN 1751-9632
Received on 24th January 2018
Revised 11th September 2018
Accepted on 30th October 2018
E-First on 4th March 2019
doi: 10.1049/iet-cvi.2018.5014
www.ietdl.org

Huy-Hieu Pham[1,2] ✉, Louahdi Khoudour[1], Alain Crouzil[2], Pablo Zegers[3], Sergio A. Velastin[4,5,6]

[1]*Cerema, Equipe-projet STI, 1 Avenue du Colonel Roche, 31400, Toulouse, France*
[2]*Institut de Recherche en Informatique de Toulouse (IRIT), Université de Toulouse, UPS, 31062 Toulouse, France*
[3]*Aparnix, La Gioconda 4355, 10B, Las Condes, Santiago, Chile*
[4]*Department of Computer Science, Applied Artificial Intelligence Research Group, University Carlos III de Madrid, 28270 Madrid, Spain*
[5]*School of Electronic Engineering and Computer Science, Queen Mary University of London, UK*
[6]*Cortexica Vision Systems Ltd., London, UK*
✉ *E-mail: huy-hieu.pham@cerema.fr*

Abstract: Recognising human actions in untrimmed videos is an important challenging task. An effective three-dimensional (3D) motion representation and a powerful learning model are two key factors influencing recognition performance. In this study, the authors introduce a new skeleton-based representation for 3D action recognition in videos. The key idea of the proposed representation is to transform 3D joint coordinates of the human body carried in skeleton sequences into RGB images via a colour encoding process. By normalising the 3D joint coordinates and dividing each skeleton frame into five parts, where the joints are concatenated according to the order of their physical connections, the colour-coded representation is able to represent spatio-temporal evolutions of complex 3D motions, independently of the length of each sequence. They then design and train different deep convolutional neural networks based on the residual network architecture on the obtained image-based representations to learn 3D motion features and classify them into classes. Their proposed method is evaluated on two widely used action recognition benchmarks: MSR Action3D and NTU-RGB+D, a very large-scale dataset for 3D human action recognition. The experimental results demonstrate that the proposed method outperforms previous state-of-the-art approaches while requiring less computation for training and prediction.

1 Introduction

Human action recognition (HAR) [1] is an important topic in machine vision. HAR methods have been widely applied in building electronic systems based on machine intelligence such as intelligent surveillance [2, 3], human–machine interaction [4, 5], health care [6], and so on. Although significant progress has been made in the last few years, HAR is still a challenging task due to many obstacles, e.g. camera viewpoint, occlusions, or large intra-class variations [7].

There are two main types of intelligence that are artificial intelligence and machine intelligence. HAR can be considered an electronic system based on machine intelligence. The first stage of any HAR system is data acquisition. Nowadays, a variety of electronic imaging systems can be used for this task such as traditional optical cameras, RGB-D sensors, thermal infrared (IR) sensors, and synthetic aperture radar (SAR) sensors. For instance, IR sensors have been exploited in HAR [8]. These sensors are able to generate images based on the heat radiated by the human body and to work independently of lighting conditions. SAR imaging has also been widely used for detecting and analysing human activities [9]. This type of sensor is able to operate far away from potential targets and functions during the daytime as well as night-time, under all weather conditions. In particular, SAR image segmentation techniques [10–12] can segment humans from other components (e.g. objects and background) in the scene. As a result, the use of SAR sensors offers big advantages for HAR, in particular for human target detection and identification in a complex environment such as in military applications.

Traditional studies on HAR mainly focus on the use of RGB sequences provided by 2D cameras. These approaches typically recognise actions based on hand-crafted local features such as cuboids [13], histogram of oriented gradients (HOG)/histogram of optical flow [14], HOG-3D [15], which are extracted from the appearance and movements of human body parts. However, it is very hard to fully capture and model the spatial–temporal information of human action due to the absence of 3D structure from the scene. Therefore, RGB-D cameras are becoming one of the most commonly-used sensors for HAR [16–18]. Recently, the rapid development of cost-effective and easy-to-use depth cameras, e.g. Microsoft Kinect™ sensor [19, 20], ASUS Xtion PRO [21], or Intel® RealSense™ [22] has helped dealing with problems related to 3D action recognition. In general, depth sensors are able to provide detailed information about the 3D structure of the human body. Therefore, many approaches have been proposed for 3D HAR based on data provided by depth sensors such as RGB sequences, depth, or combining these two data types (RGB-D). Furthermore, these devices have integrated real-time skeleton tracking algorithms [23] that provide high-level information on human motions in a 3D space. Thus, exploiting skeletal data for 3D HAR is an active research topic in computer vision [24–28].

In recent years, approaches based on convolutional neural networks (CNNs) have achieved outstanding results in many image recognition tasks [29–31]. After the success of AlexNet [29], a new direction of research has been opened for designing and optimising higher performing CNN architectures. As a result, there is substantial evidence that seems to show that the learning performance of CNNs may be significantly improved by increasing the number of hidden layers [32–34]. In the literature of HAR, many studies have indicated that CNNs have the ability to learn complex motion features better than hand-crafted approaches. However, most previous works have just focused on exploring simple CNNs such as AlexNet [29] and have not exploited the potential of recent state-of-the-art very deep CNNs (D-CNNs), e.g. residual networks (ResNet) [35]. In addition, most existing CNN-based approaches limit themselves to using RGB-D sequences as the input to learning models. Although RGB-D images are informative for understanding human action, the computation complexity of learning models increases rapidly when the

dimension of the input features is large. Therefore, representation learning models based on RGB-D modality become more complex, slower and less practical for solving large-scale problems as well as real-time applications.

Different from previous works, to take full advantages of 3D skeletal data and the learning capacity of D-CNNs, this study proposes an end-to-end deep learning framework for 3D HAR from skeleton sequences. We focus on solving two main issues: first, using a simple skeleton-to-image encoding method to transform the 3D coordinates of the skeletal joints into RGB images. The encoding method needs to ensure that the image-based representation of skeleton sequences is able to effectively represent the spatial structure and temporal dynamics of the human action. Second, we design and train different D-CNNs based on ResNets [35] – a recent state-of-the-art CNN for image recognition, to learn and classify actions from the obtained image-coded representations.

This paper is an extended version of our work published in the eighth International Conference of Pattern Recognition Systems (ICPRS 2017) [36]. It is part of our research project of investigating and developing a low-cost system based on RGB-D data provided by depth sensors for understanding human actions and analysing their behaviours in indoor environments such as inside home, offices, residential buildings, buses, trains etc. In general, three observations motivate our exploration of using ResNet [35] for 3D HAR from skeleton sequences, including: (i) human actions can be correctly represented as the movements of skeletons [27] and the spatial–temporal dynamics of skeletons can be transformed into 2D image structure, which can be effectively learned by D-CNNs; (ii) skeletal data is high-level information with much less complexity than RGB-D sequences, this advantage makes our action learning model much simpler and faster; (iii) many pieces of evidence [32, 33] show that the deeper convolutional model, especially ResNet [35] can boost the learning accuracy in image recognition tasks.

We evaluate the proposed method on two benchmark skeleton datasets (i.e. MSR Action3D [37] and NTU-RGB+D [38]). Experimental results confirmed the above statements since our method achieved state-of-the-art performance compared with the existing results using the same evaluation protocols. Furthermore, we also indicate the effectiveness of our learning framework in terms of computational complexity.

In summary, two main contributions of this study include:

- First, we introduce a new skeleton-based representation (Section 3.1) and an end-to-end learning framework [Codes and pre-trained models will be shared to the community at https://github.com/huyhieupham/after publication.] (Section 3.2) based on D-CNNs to learn the spatio-temporal evolutions of 3D motions and then to recognise human actions;
- Second, we show the effectiveness of our method on HAR tasks by achieving state-of-the-art performance on two benchmark datasets, including the most challenging skeleton benchmark currently available for 3D HAR [38] (Section 5).

The advantage of the proposed method is that it has high-computational efficiency (Section 5.4). Moreover, this approach is general and can be easily applied to other time-series problem, e.g. recognising human actions with mobile devices with integrated inertial sensors.

The rest of the paper is organised as follows: Section 2 discusses related works. In Section 3, we present the details of our proposed method, including the colour-encoding process from skeletons to RGB images and deep learning networks. Datasets and experiments are described in Section 4. Experimental results are reported in Section 5 with a detailed analysis of computational efficiency. Finally, Section 6 concludes the paper with a discussion on future work.

2 Related work

Skeleton-based action recognition (SBAR) using depth sensors has been widely studied in recent years. In this section, we briefly review existing SBAR methods, including two main categories that are directly related to our work. The first category is approaches based on the hand-crafted local features of skeletons. The second is approaches based on deep learning networks, especially recurrent neural networks with long short-term memory (RNN-LSTMs).

Approaches based on hand-crafted local features: Human motion can be considered as a spatio–temporal pattern. Many researchers have built hand-crafted feature representations for SBAR and then used temporal models for modelling 3D human motion. For instance, Wang *et al.* [24] represented the human motion from skeletal data by means of the pairwise relative positions of the key joints. The authors then employed a hidden Markov model (HMM) [39] for modelling temporal dynamics of actions. A similar approach has been presented by Lv *et al.* [40] where the 3D joint position trajectories have been mapped into feature spaces. The dynamics of actions was then learned by one continuous HMM. Xia *et al.* [25] proposed the use of a histogram-based representation of human pose, and then actions were classified into classes by a discrete HMM. Wu and Shao [41] also exploited HMM for recognising actions from high-level features of skeletons. Vemulapalli *et al.* [27] represented the 3D geometric interactions of the body parts in a Lie Group. These elements were processed with a Fourier temporal pyramid (FTP) transformation for modelling temporal evolutions of the original motions. Different from the studies above, Luo *et al.* [42] proposed a new dictionary learning model to learn the spatio–temporal information from skeleton sequences. Although promising results have been achieved from approaches based on hand-crafted local features and probabilistic graphical models, they have some limitations that are very difficult to overcome, e.g. most of these approaches are data-dependent and require a lot of hand-designing features. HMM-based methods require preprocessing input data in which the skeleton sequences need to be segmented or aligned. Meanwhile, FTP-based approaches can only utilise limited contextual information of actions.

Approaches based on deep learning: RNN-LSTMs [43, 44] are able to model the contextual information of the temporal sequences as skeleton data. Thus, many authors have explored RNN-LSTMs for SAR. For instance, Du *et al.* [45] proposed an end-to-end hierarchical RNN-LSTM for modelling local motions of a body part in which all skeleton frames were divided into five parts according to the human physical structure. Each part of a skeleton was then fed into an independent RNN and then fused to be the inputs of higher layers. The final representation was used for the classification task. Zhu *et al.* [46] proposed an LSTM network with a mixed-norm regularisation term to cost functions in order to learn the co-occurrence of discriminative features from skeletal data. Liu *et al.* [47] introduced a new gating mechanism with an LSTM network to analyse the reliability of input skeleton sequences. In another study, Shahroudy *et al.* [38] used a part-aware LSTM network in which the memory-cell was divided into sub-cells such that each sub-cell can model long-term contextual representation of a body part. Finally, all these sub-cells were concatenated to the final output. Although RNN-LSTMs are able to model the long-term temporal of motion and experimental results provided that RNN-LSTMs have outperformed many other approaches, RNN-based approaches just consider skeleton sequences as a kind of low-level feature by feeding directly the raw 3D joint coordinates carried in skeletons into the network input. The huge number of input parameters may make RNNs become very complex and easily lead to overfitting due to the insufficient training data. Moreover, many RNN-LSTMs act as a classifier and cannot extract high-level features [48] for recognition tasks.

Several authors have exploited the feature learning ability of CNNs on skeletal data [49, 50]. However, such studies mainly focus on the use of complex encoding methods for finding good skeletal representations and learning geometric features carried in skeleton sequences with simple CNN architectures. In contrast, in this study, we concentrate on proposing a new and simple skeleton-based representation and exploiting the power of D-CNNs for action recognition. Our experiments on two public datasets, including the MSR Action3D dataset [37] and the NTU-RGB+D dataset [38] show state-of-the-art performance on both datasets.

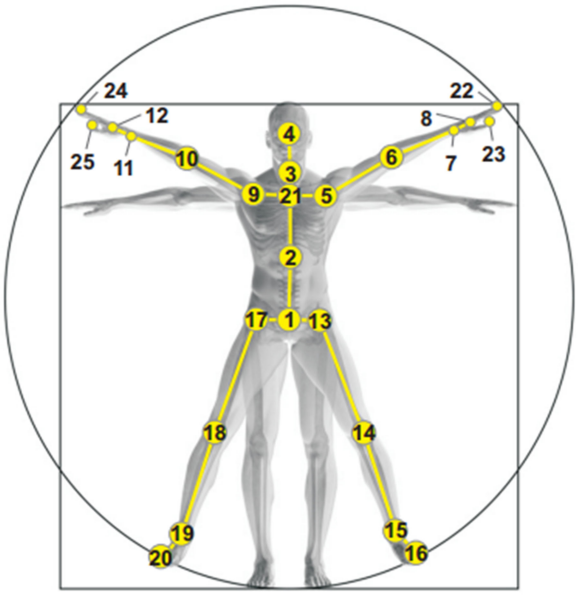

Fig. 1 *Position of 25 joints in the human body extracted by Kinect v2 sensor [38]. Skeleton sequences can be recorded frame-by-frame at 30 frames per second*

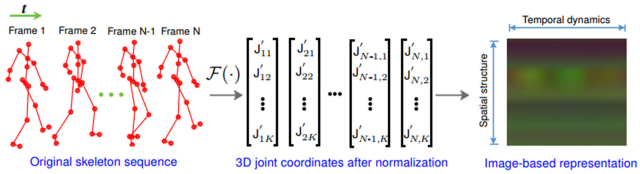

Fig. 2 *Illustration of the data transformation process. Each skeleton sequence is encoded into a single colour image that represents the spatio-temporal evolutions of the motion. Here, N denotes the number of frames in each sequence while K denotes the number of joints in each frame. The value of K depends on each RGB+D dataset, e.g. K is equal to 20 for MSR Action3D dataset [37]. Meanwhile, K is equal to 25 for NTU-RGB + D dataset [38]*

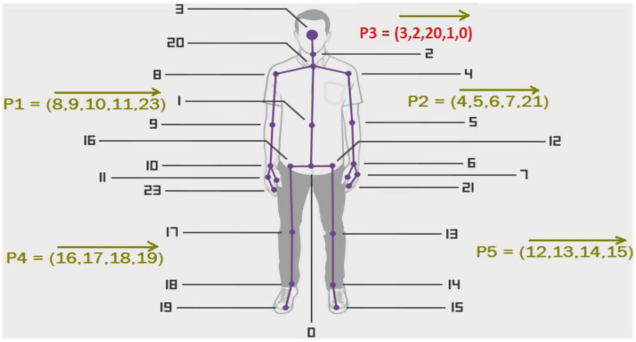

Fig. 3 *Map of joints in each part from P_1 to P_5 on a skeleton provided by Microsoft Kinect v2™ sensor [55]*

Moreover, in terms of computational cost, our measurement and analyses show that the proposed model is fast enough for many real-time applications.

3 Proposed method

This section presents the proposed method. We first describe the colour-encoding process that allows us to represent the spatial structure and temporal dynamics of skeleton sequences as the static structure of a 2D colour image. We then review the main idea of ResNet [35] and propose five different ResNet configurations, including 20-layer, 32-layer, 44-layer, 56-layer, and 110-layer networks to learn motion features carried in the image-based representations and perform action classification.

3.1 Building image-based representation from skeleton sequences

Currently, skeletal data that contain 3D joint coordinates can be obtained from depth cameras via real-time skeleton estimation algorithms [51, 52]. This technology allows extracting the position of the key joints in the body, which is suitable for 3D HAR problems, e.g. the latest version of the Microsoft Kinect™ sensor [19, 20] (Kinect v2 sensor) can track the main 25 joints of the human body in real-time speed as shown in Fig. 1.

In this study, instead of feeding raw 3D joint coordinates directly to RNN-LSTMs as in many previous works [38, 45–47], we propose the use of D-CNNs for learning motion features from skeleton sequences. The first step is to find a skeleton-based representation that can effectively capture the spatio-temporal evolutions of skeletons and that can also be easily learned by learning method as D-CNNs. One solution to this problem is to transform each skeleton sequence into a single colour image (e.g. RGB image) in which the pixel values must have the ability to represent the movement of skeletons.

To this end, we transform the 3D joint coordinates of each skeleton into a new space by normalising their coordinates using a transformation function. Specifically, given a skeleton sequence $\mathcal{S}$ with $\boldsymbol{N}$ frames $[\boldsymbol{F}_1, \boldsymbol{F}_2, \ldots, \boldsymbol{F}_N]$, let (x_i, y_i, z_i) be the 3D coordinates of each joint in frame $\{\boldsymbol{F}_n\} \in \mathcal{S}, n \in [1, N]$. A normalisation function $\mathscr{F}(\cdot)$ is used to transform all 3D joint coordinates to the range of [0, 255] as follows:

$$(x_i', y_i', z_i') = \mathscr{F}(x_i, y_i, z_i), \tag{1}$$

$$x_i' = 255 \times \frac{(x_i - \min\{\mathscr{C}\})}{\max\{\mathscr{C}\} - \min\{\mathscr{C}\}}, \tag{2}$$

$$y_i' = 255 \times \frac{(y_i - \min\{\mathscr{C}\})}{\max\{\mathscr{C}\} - \min\{\mathscr{C}\}}, \tag{3}$$

$$z_i' = 255 \times \frac{(z_i - \min\{\mathscr{C}\})}{\max\{\mathscr{C}\} - \min\{\mathscr{C}\}}, \tag{4}$$

where (x_i', y_i', z_i') is the 3D joint coordinate in the normalised space $\mathcal{S}'$. The $\max\{\mathscr{C}\}$ and $\min\{\mathscr{C}\}$ are the maximum and minimum values of all coordinates, respectively. To preserve the spatio-temporal information of skeleton movements, we stack all normalised frames according to the temporal order $\mathcal{S}' = [\boldsymbol{F}'_1, \boldsymbol{F}'_2, \ldots, \boldsymbol{F}'_N]$ to represent the whole action sequence. These elements are quantified to RGB colour space and can be stored as RGB images. In this way, we convert the skeletal data to 3D tensors that will then be fed into deep learning networks as the input for feature leaning and classification. Fig. 2 illustrates the skeleton-to-image transformation process.

Naturally, the human body is structured as four limbs and one trunk (see Fig. 1). Simple actions can be performed through the movement of a limb, e.g. hand-waving, kicking forward etc. More complex actions combine the movements of a group of limbs or the whole body, e.g. running or swimming. To keep the local characteristics of the human action [53, 54], we divide each skeleton into five parts, including two arms (P_1, P_2), two legs (P_4, P_5), and one trunk (P_3). In each part, the joints are concatenated according to their physical connections as shown in Fig. 3. These parts are then connected in a sequential order: $P_1 \rightarrow P_2 \rightarrow P_3 \rightarrow P_4 \rightarrow P_5$. The whole process of rearranging all frames in a sequence can be done by rearranging the order of rows of pixels in the colour-based representation. This process is illustrated in Fig. 4.

Like that, we have encoded skeleton sequences into RGB images. Fig. 5 shows some examples of RGB images obtained from input sequences of MSR Action3D dataset [37]. These images will be learned and classified by D-CNN models. This way, the original skeleton sequence will be recognised via the corresponding image.

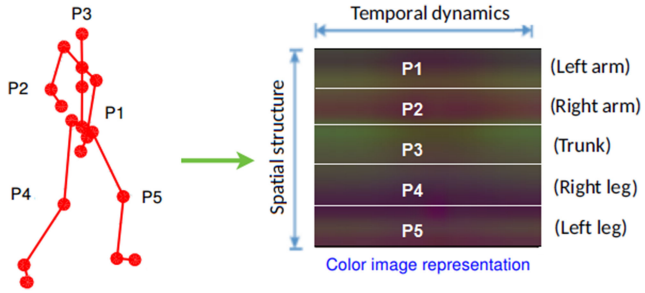

Fig. 4 *Rearranging the order of joints according to the human body physical structure*

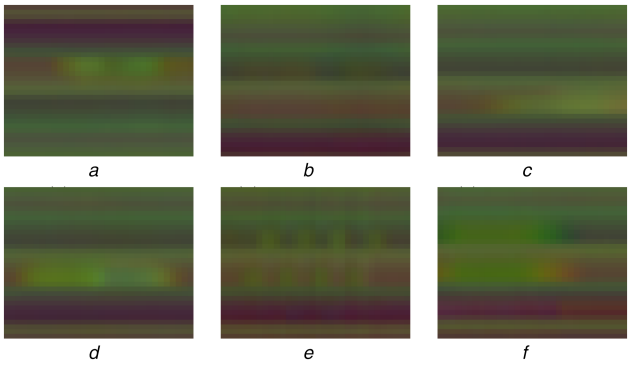

Fig. 5 *Image-based representations obtained from several samples of MSR Action3D dataset [37]. In our experiments, all images are resized to* 32×32 *pixels. Best viewed in colour*

(a) Draw X, ***(b)*** Forward kick, ***(c)*** Hand catch, ***(d)*** High throw, ***(e)*** Jogging, ***(f)*** Two-hand wave

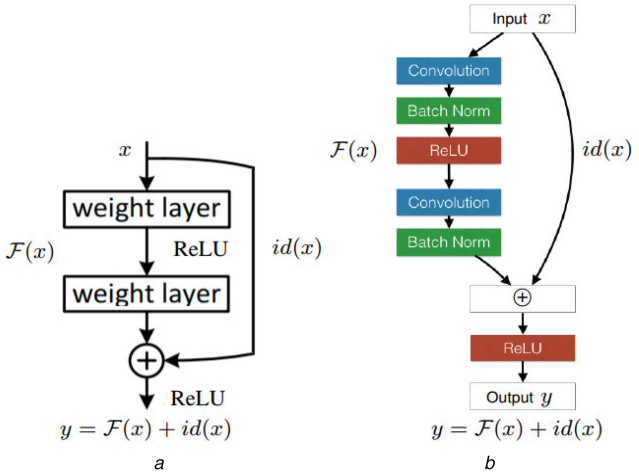

Fig. 6 *Building block of ResNet*

(a) Information flow in a ResNet building block, ***(b)*** Implementation details of a ResNet block introduced in the original paper [35]. The symbol ⊕ denotes element-wise addition

3.2 Deep ResNets for skeleton-based HAR

In this section, we introduce our network designs for recognising human action from image-based representation obtained in Section 3.1. Before that, in order to put our method into context, we will briefly review the key idea of ResNet [35].

3.2.1 Residual learning method: Very deep neural networks, especially D-CNNs have demonstrated to have a high performance on many visual-related tasks [32, 34]. However, D-CNNs are very difficult to optimise due to the vanishing gradients problem. If the network is deep enough, the error signal from the output layer can be completely attenuated on its way back to the input layer. In addition, another problem called 'the degradation phenomenon [56]' also impedes the convergence of deeper networks. More specially, adding more layers to a deep network can lead to higher training/testing error. To overcome these challenges, ResNet [35] has been introduced. The key idea behind ResNet architectures is the presence of shortcut connections between input and output of each ResNet building block. These shortcut connections are implemented by identity mappings and are able to provide a path for gradients to back propagate to early layers in the network. This idea improves the information flow in ResNets and helps them to learn faster. Not only that, experimental results on standard datasets such as CIFAR-10 [57], and ImageNet [58] for image classification tasks showed that the use of the shortcut connections in ResNet architectures can boost recognition performance. Fig. 6 shows information flow in a ResNet building block and their implementation details.

Mathematically, a layer or a series of layers in a typical CNN model will try to learn a mapping function $y = \mathcal{F}(x)$ from input feature x. However, a ResNet building block will learn the function $y = \mathcal{F}(x) + \mathrm{id}(x)$, where $\mathrm{id}(x)$ is an identity function $\mathrm{id}(x) = x$. The additional element $\mathrm{id}(x)$ is the key factor that improves the information flow through layers during training ResNets.

3.2.2 Network design: A deep ResNet can be constructed from multiple ResNet building blocks that are serially connected to each other. To search for the best recognition performance, we suggest different configurations of ResNet with 20, 32, 44, 56, and 110 layers. For further details, see the Appendix. These networks are denoted as ResNet-20, ResNet-32, ResNet-44, ResNet-56, and ResNet-110, respectively. In our implementations, a ResNet building block performs the learning of a function $y = \mathcal{F}(x) + \mathrm{id}(x)$ where $\mathrm{id}(x) = x$ and $\mathcal{F}(x)$ is implemented by a sequence of layers: *Conv-BN-ReLU-Dropout-Conv-BN*. Specifically, each ResNet block uses the convolutional layers (*Conv*) with 3×3 filters. The batch normalisation (BN) [59] and non-linear activation function rectified linear unit (ReLU) [60] are applied after each *Conv*. To reduce possible overfitting, we add a dropout layer [61] with a rate of 0.5 into each ResNet block, located between two *Conv* layers and after *ReLU*. Finally, another *ReLU* layer [60] is used after the element-wise addition. The proposed ResNet unit is shown in Fig. 7. We refer the interested reader to the Supplementary Materials to see details of the proposed network architectures.

In this learning framework, all networks are designed for accepting images with size 32×32 pixels and trained in an end-to-end manner using stochastic gradient descent algorithm [62] from scratch. The last fully-connected layer of the network represents the action class scores and its size can be changed corresponding to the number of action classes.

3.2.3 Learning to recognise actions from skeleton-based representations: In order to recognise an action from a given skeleton sequence $\mathcal{S}$ with N frames $[\boldsymbol{F}_1, \boldsymbol{F}_2, \ldots, \boldsymbol{F}_N]$, we firstly encode all frames of $\mathcal{S}$ into a single RGB image I_{RGB} as mentioned in Section 3.1

$$I_{\mathrm{RGB}} = [\boldsymbol{F}'_1, \boldsymbol{F}'_2, \ldots, \boldsymbol{F}'_N], \tag{5}$$

where all elements of $\boldsymbol{F}'_{i|i \in [1,N]}$ were normalised to the range of [0, 255] as colour pixels and rearranged according to their physical structure (see Fig. 4). We then propose the use of ResNets to learn and classify the obtained colour images. During the training phase, we minimise the cross-entropy loss function $\mathscr{L}_{\mathscr{X}}(\boldsymbol{y}, \hat{\boldsymbol{y}})$ between the action labels $\boldsymbol{y}$ of I_{RGB} and the predicted action labels y by the network over the training set $\mathscr{X}$. In other words, the network will be trained to solve the following problem:

$$\mathbf{Arg} \min_{\mathscr{W}}(\mathscr{L}_{\mathscr{X}}(\boldsymbol{y}, \hat{\boldsymbol{y}})) = \mathbf{Arg} \min_{\mathscr{W}}\left(-\frac{1}{M}\sum_{i=1}^{M}\sum_{j=1}^{C}\boldsymbol{y}_{ij}\log\hat{\boldsymbol{y}}_{ij}\right), \tag{6}$$

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

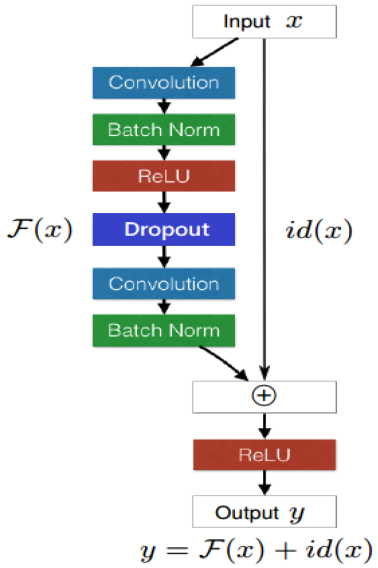

Fig. 7 *Proposed ResNet building block. Here, the symbol ⊕ denotes element-wise addition*

Table 1 Three subsets AS1, AS2, and AS3 of the MSR Action3D dataset [37]

AS1	AS2	AS3
horizontal arm wave	high arm wave	high throw
hammer	hand catch	forward kick
forward punch	draw x	side kick
high throw	draw tick	jogging
hand clap	draw circle	tennis swing
Bend	two hand wave	tennis serve
tennis serve	forward kick	golf swing
pickup & throw	side-boxing	pickup & throw

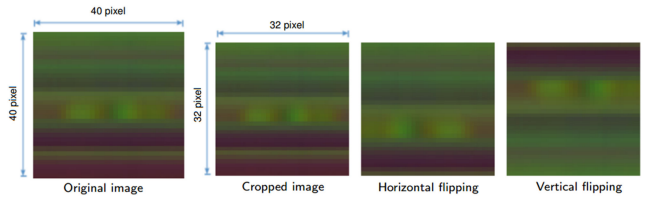

Fig. 8 *Illustration of data augmentation methods applied to MSR Action3D dataset [37]*

4 Experiments

In this section, we evaluate the effectiveness of the proposed method on two benchmark datasets: MSR Action3D [37] and NTU-RGB+D [38]. To have a fair comparison with state-of-the-art approaches in the literature, we follow the evaluation protocols as provided in the original papers [37, 38]. Recognition performance will be measured by average classification accuracy.

4.1 Datasets and experimental protocols

4.1.1 MSR Action3D dataset [37]: This dataset [the MSR Action3D dataset can be obtained at https://www.uow.edu.au/~wanqing/#Datasets] was collected with the Microsoft Kinect v1™ sensor. It contains 20 different actions. Each action was performed by ten subjects for two or three times. The experiments were conducted on 557 valid sequences of the dataset after removing the defective sequences. We followed the same experimental protocol as many other authors, in which the whole data is divided into three subsets named as *AS1*, *AS2*, and *AS3*. For each subset, five subjects (with IDs: 1, 3, 5, 7, 9) are selected for training and the rest (with IDs: 2, 4, 6, 8, 10) for the test. More details of the action classes in each subset can be found in Table 1.

4.1.2 NTU-RGB+D dataset [38]: The NTU-RGB+D dataset [obtained at http://rose1.ntu.edu.sg/Datasets/actionRecognition.asp] is a very large-scale RGB+D dataset. To the best of our knowledge, the NTU-RGB+D is currently the largest dataset that contains skeletal data for HAR. It is a very challenging dataset due to the large intra-class variations and multiple viewpoints. Specifically, the NTU-RGB+D provides more than 56,000 videos, collected from 40 subjects for 60 action classes. The list of action classes in NTU RGB+D dataset includes: *drinking, eating, brushing teeth, brushing hair, dropping, picking up, throwing, sitting down, standing up (from sitting position), 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 (on watch), rubbing two hands together, bowing, shaking head, wiping face, saluting, putting palms together, crossing hands in front, sneezing/coughing, staggering, falling down, touching head (headache), touching chest (stomach-ache/heart pain), touching back (back-pain), touching neck (neck-ache), vomiting, fanning self, punching/slapping other person, kicking other person, pushing other person, patting other's back, pointing to the other person, hugging, giving something to other person, touching other person's pocket, handshaking, walking towards each other, and walking apart from each other*.

The NTU RGB+D data was collected by using the Microsoft Kinect v2™ sensor. Each skeleton frame provides the 3D coordinates of 25 body joints. Two different evaluation criteria have been suggested, including *cross-subject* and *cross-view*. In the *cross-subject* settings, 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 are used for test. For *cross-view* settings, the sequences provided by cameras 2 and 3 are used for training and sequences provided by camera 1 are used for the test.

4.2 Data augmentation

Very deep neural networks such as ResNets [35] require a lot of data to optimise. D-CNNs cannot learn and classify images effectively when training with an insufficient data amount. Unfortunately, we have only 557 skeleton sequences (<30 images per class) on MSR Action3D dataset [37]. Therefore, data augmentation has been applied to prevent ResNets from overfitting. We used random cropping, flip horizontally and vertically techniques to generate more training samples. More specifically, 8× cropping has been applied on 40×40 images to create 32×32 images. Then, their horizontally and vertically flipped images are also created. Fig. 8 illustrates some data augmentation techniques on an image sample obtained from the MSR Action3D dataset [37]. For the NTU-RGB+D dataset [38], we do not apply any data augmentation method due to its very large-scale.

4.3 Implementation details

The image-based representations are computed directly from the original skeletons without any data preprocessing technique, e.g. transforming the coordinate system, using a fixed number of frames, or removing noise. In this project, the MatConvNet toolbox [63] was used to design and train our deep learning networks. During the training phase, we use mini-batches of 128 images for ResNet-20, ResNet-32, ResNet-44, and ResNet-56 networks. For ResNet-110 network, we use mini-batches of 64 images. We initialise the weights randomly and train all networks for 120 epochs on a computer using Geforce GTX 1080 Ti GPU with 11 GB RAM from scratch. The initial learning rate is set to 0.01 and is decreased to 0.001 at epoch 75. The last 45 epochs use a learning rate of 0.0001. The weight decay is set at 0.0001 and the momentum is 0.9.

Table 2 Test accuracies (%) of our proposed ResNets on AS1, AS2, and AS3 subsets. The best results and configuration are highlighted in bold

Network	AS1, %	AS2, %	AS3, %	Aver., %
ResNet-20	99.60	98.90	100.0	99.50
ResNet-32	**99.80**	**99.00**	**99.80**	**99.53**
ResNet-44	99.50	98.90	100.0	99.48
ResNet-56	99.40	98.70	99.50	98.20
ResNet-110	99.00	96.50	99.30	99.28

5 Experimental results and analysis

This section reports our experimental results on the MSR Action3D dataset [37] and the NTU-RGB+D dataset [38]. We compare the obtained classification rates with state-of-the-art approaches in the literature [25, 27, 37, 38, 42, 45, 47, 64–74]. In addition, we also provide a detailed analysis of the computational efficiency of the proposed learning framework.

5.1 Result on MSR Action3D dataset

Experimental results on MSR Action3D dataset [37] are shown in Table 2. We achieved the best classification accuracy with the ResNet-32 model. More specifically, classification accuracies are 99.80% on AS1, 99.00% on AS2, and 99.80% on AS3. We obtained a total average accuracy of 99.53%. Our result outperforms many previous works [25, 37, 42, 45, 64–79] (see Table 3). Moreover, this result also indicates that the proposed ResNet architecture learns image features better than the original ResNet architecture [36]. The learning curves of all proposed ResNets on AS1, AS2, and AS3 are shown in Fig. 9*a*–*c*, respectively.

5.2 Result on NTU-RGB+D dataset

As shown in Table 4, the proposed learning framework reaches competitive results with an accuracy of 73.40% on cross-subject evaluation and an accuracy of 80.40% on cross-view evaluation. The obtained results indicate that our method can deal with large intra-class variations and multiple viewpoints dataset as NTU-RGB +D [38]. Table 5 provides a comparison with published studies [25, 37, 64–71, 73, 74, 80, 81]. It is clear that our method surpasses many previous approaches in the same experimental conditions. The learning curves of all ResNet configurations on two evaluation settings are shown in Figs. 9*d* and *e*.

5.3 Convergence rate analysis

Fig. 9 shows the learning curves of the five proposed deep learning networks on the MSR Action3D [37] and the NTU-RGB+D [38] datasets. We can find that the training convergence rate of the networks is different in the two datasets. More specifically, on MSR Action3D dataset [37], the proposed networks exhibit rapid convergence after 50 epochs at the beginning of the training process. Meanwhile, these networks can only start to converge after near 80 epochs on the NTU-RGB+D dataset [38]. This phenomenon can be explained by the complexity of feature spaces generated by the two datasets. With more than 56,000 videos collected from 40 subjects for 60 action classes, it is clear that the NTU-RGB+D dataset [38] is more complex than the MSR Action3D dataset [37]. This leads to deep learning networks needing more iteration to start convergence.

As shown in Tables 2 and 4, the best recognition accuracy and configuration are highlighted in bold. More specifically, our experimental result on the MSR Action3D indicated that the baseline ResNet-32 achieved the best overall accuracy (99.53%). On the NTU RGB+D dataset, the ResNet-32 network was the best version on the Cross-View setting (80.40%). However, it worked worse than the ResNet-20 on the cross-subject setting. This is a weak aspect of the proposed approach. This limitation could be overcome by using ensemble learning techniques [89].

Table 3 Comparison with the state-of-the-art approaches on MSR Action3D dataset [37]. The best performances are in bold

Method	AS1, %	AS2, %	AS3, %	Aver., %
bag of 3D points [37]	72.90	71.90	79.20	74.67
motion trail model [70]	73.70	81.50	81.60	78.93
histograms of 3D joints [25]	87.98	85.48	63.46	78.97
motion and shape features [69]	81.00	79.00	82.00	80.66
spatial-temporal features [80]	77.36	73.45	91.96	80.92
space-time occupancy [64]	84.70	81.30	88.40	84.80
improved space-time Occ. [74]	91.70	72.20	98.60	87.50
HON4D [81]	N/A	N/A	N/A	88.89
skeletal quads [71]	88.39	86.61	94.59	89.86
multi-modality information [73]	92.00	85.00	93.00	90.00
depth motion maps [66]	96.20	83.20	92.00	90.47
covariance descriptors [68]	88.04	89.29	94.29	90.53
histogram of oriented displacements [67]	92.39	90.18	91.43	91.26
moving pose [82]	N/A	N/A	N/A	91.70
skeletal and silhouette fusion [65]	92.38	86.61	96.40	91.80
improved key poses [78]	91.53	90.23	97.06	92.94
pose-based representation [72]	91.23	90.09	99.50	93.61
hierarchical recurrent neural network (H-RNN) [45]	93.33	94.64	95.50	94.49
local binary patterns [75]	98.10	92.00	94.60	94.90
spatio-temporal pyramid [76]	99.10	92.90	96.40	96.10
depth motion maps [77]	99.10	92.30	98.20	96.50
features combination [79]	N/A	N/A	N/A	97.10
group sparsity [42]	97.20	95.50	99.10	97.26
our best configuration	**99.80**	**99.00**	**99.80**	**99.53**

5.4 Analysis of training and prediction time

We take the AS1 subset of the MSR Action3D dataset [37] and the proposed ResNet-32 network for illustrating the computational efficiency of our deep learning framework. As shown in Fig. 10, the proposed framework contains three main stages, including the encoding process from skeleton sequences into RGB images (stage A), the supervised training stage (stage B), and the prediction stage (stage C). With the implementation in Matlab using MatconvNet toolbox [63] [MatconvNet is an open source library and can be downloaded at address: http://www.vlfeat.org/matconvnet/] on a single NVIDIA GeForce GTX 1080 Ti GPU [more details about the GPU specification, please refer to https://www.nvidia.com/en-us/geforce/products/10series/geforce-gtx-1080-ti/], without parallel processing, we take on average 4.15×10^{-3} s per sequence during training stage. While the prediction time, including the time for encoding skeletons into RGB images and classification by the pre-trained ResNet, takes on average 21.84×10^{-3} s per sequence (see Table 6). These results verify the effectiveness of our proposed method, not only in terms of accuracy but also in terms of computational efficiency.

6 Conclusion and future work

In this study, we have proposed a new 3D motion representation and an end-to-end deep learning framework based on ResNets [35] for HAR from skeleton sequences. To this end, we transformed the 3D joint coordinates carried in skeleton sequences into RGB images via a colour encoding process. We then designed and trained different D-CNNs based on the ResNet architecture for learning the spatial and temporal dynamics of human motion from image-coded representations. The experimental results on two well-established datasets, including the largest RGB-D dataset currently available, demonstrated that our method can achieve state-of-the-art performance whilst requiring a low-computation cost for the training and prediction stages. For future work, we aim to extend this research by investigating a new skeleton encoding

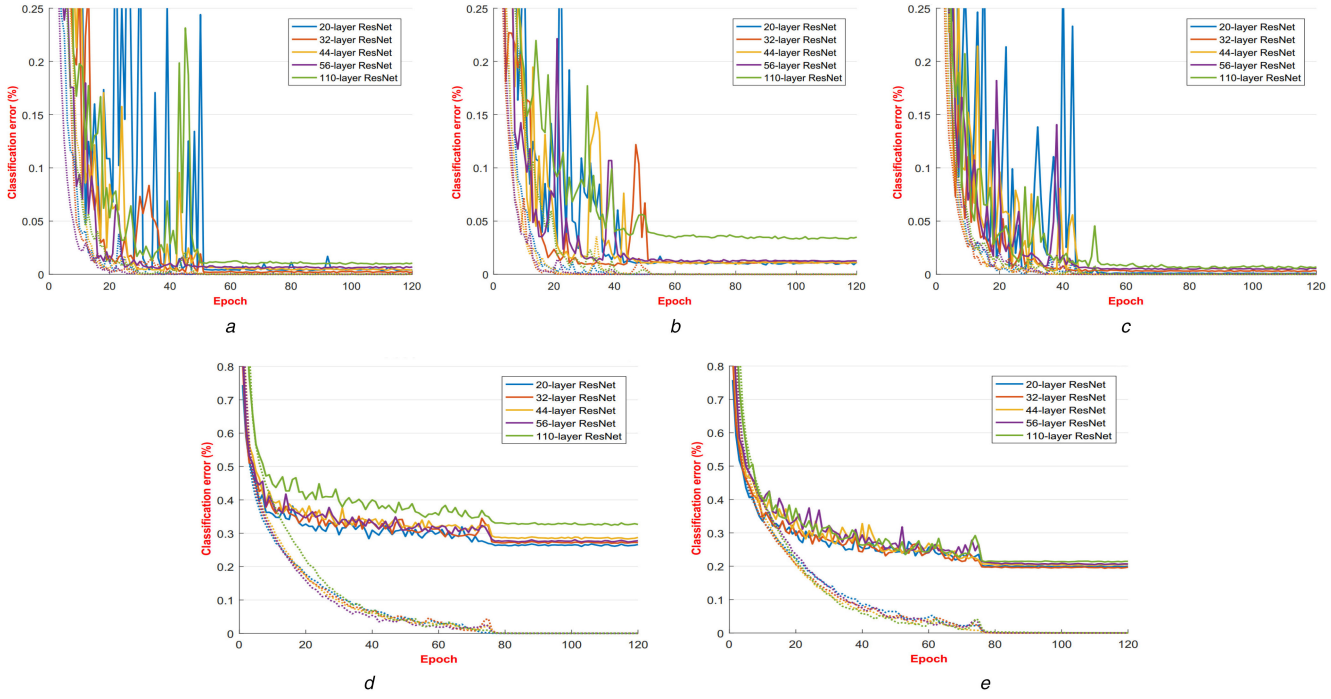

Fig. 9 *Learning curves of 20-layer, 32-layer, 44-layer, 56-layer, and 110-layer ResNets on the MSR Action3D [37] and the NTU-RGB+D [38] datasets. Dashed lines denote training errors while bold lines denote test errors. Best viewed on a computer with zoomed-in and in colour*
(a) Learning on an AS1 subset, ***(b)*** Learning on an AS2 subset, ***(c)*** Learning on an AS3 subset, ***(d)*** Learning on NTU-RGB + D dataset (view-cross settings), ***(e)*** Learning on NTU-RGB + D dataset (cross-subject settings)

Table 4 Test accuracies (%) of our proposed networks on cross-subject and cross-view settings [38]. The best results and configuration are in bold

Network	Cross-subject, %	Cross-view, %
ResNet-20	73.40	80.10
ResNet-32	**73.10**	**80.40**
ResNet-44	72.40	79.80
ResNet-56	73.00	79.80
ResNet-110	67.40	78.60

Table 5 Comparison with state-of-the-art methods on NTU-RGB+D dataset [38]. The best results and configuration are marked in bold

Method	Cross-subject, %	Cross-view, %
HON4D [81]	30.56	7.26
super normal vector [83]	31.82	13.61
joint angles + HOG2 [84]	32.24	22.27
skeletal quads [71]	38.62	41.36
shuffle and learn [85]	47.50	N/A
histograms of key poses [86]	48.90	57.70
lie group [27]	50.08	52.76
rolling rotations [87]	52.10	53.40
H-RNN [45] (reported in [38])	59.07	63.79
P-LSTM [38]	62.93	70.27
long-term motion [88]	66.22	N/A
spatio-temporal LSTM [47]	69.20	77.70
our best configuration	**73.40**	**80.40**

Table 6 Execution time of each component of the proposed learning framework

Stage	Average processing time
stage A	20.8×10^{-3} s per sequence (CPU time)
stage B	4.15×10^{-3} s per sequence (GPU time)
stage C	21.84×10^{-3} s per sequence (CPU time)

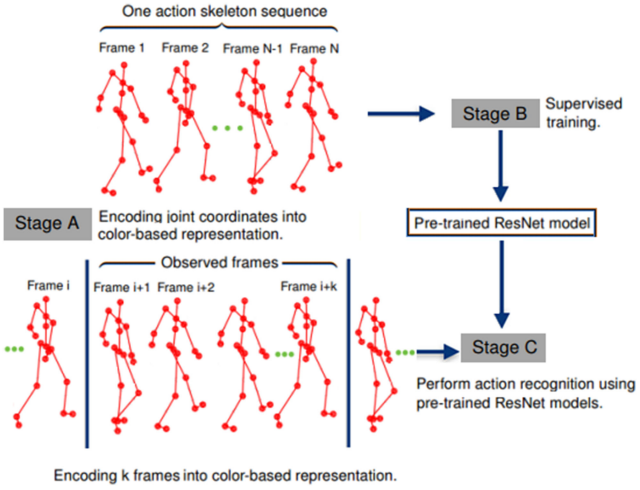

Fig. 10 *Three main stages of the proposed deep learning framework*

method, in which the Euclidean distances and orientation relations between joints will be exploited. In addition, we plan to build a better feature learning and classification framework by experimenting some new and potential architectures based on the idea of ResNet such as Inception-ResNets [90] or densely connected convolutional networks (DenseNet [91]). In order to overcome the limitations of kinect sensors when dealing with HAR in which the lighting conditions are available, we will also consider some new approaches for estimating human body key-points such as deep learning based approaches [92, 93], kurtosis wavelet energy (KWE) and skewness wavelet energy (SWE) methods [10]. We hope that our study will open a new door for the computer vision community on exploiting the potentials of very D-CNNs and skeletal data for 3D HAR.

7 Acknowledgments

This research was carried out at the Cerema Research Center (CEREMA) and Toulouse Institute of Computer Science Research (IRIT), Toulouse, France. 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 no. 600371, el Ministerio de Economia, Industria y Competitividad (COFUND2013-51509) el Ministerio de Educación, cultura y Deporte (CEI-15-17) and Banco Santander.

8 References

[1] Ranasinghe, S., Al Machot, F., Mayr, H.C.: 'A review on applications of activity recognition systems with regard to performance and evaluation', *Int. J. Distrib. Sens. Netw.*, 2016, **12**, (8), https://doi.org/10.1177/1550147716665520

[2] Niu, W., Long, J., Han, D., *et al.*: 'Human activity detection and recognition for video surveillance'. IEEE Int. Conf. on Multimedia and Expo (ICME), Taipei, Taiwan, 2004, vol. 1, pp. 719–722

[3] Lin, W., Sun, M.-T., Poovandran, R., *et al.*: 'Human activity recognition for video surveillance'. IEEE Int. Symp. on Circuits and Systems (ISCAS), Seattle, WA, USA, 2008, pp. 2737–2740

[4] Pickering, C.A., Burnham, K.J., Richardson, M.J.: 'A research study of hand gesture recognition technologies and applications for human vehicle interaction'. Institution of Engineering and Technology Conf. on Automotive Electronics, 2007, pp. 1–15

[5] Sonwalkar, P., Sakhare, T., Patil, A., *et al.*: 'Hand gesture recognition for real time human machine interaction system', *Int. J. Eng. Trends Technol.*, 2015, **19**, (5), pp. 262–264

[6] Pansiot, J., Stoyanov, D., McIlwraith, D., *et al.*: 'Ambient and wearable sensor fusion for activity recognition in healthcare monitoring systems'. Int. Workshop on Wearable and Implantable Body Sensor Networks (BSN 2007), Aachen, Germany, 2007, pp. 208–212

[7] Poppe, R.: 'A survey on vision-based human action recognition', *Image Vis. Comput.*, 2010, **28**, (6), pp. 976–990

[8] Akula, A., Shah, A.K., Ghosh, R.: 'Deep learning approach for human action recognition in infrared images', *Cogn. Syst. Res.*, 2018, **50**, pp. 146–154

[9] Gürbüz, S.Z., Melvin, W.L., Williams, D.B.: 'Detection and identification of human targets in radar data'. Signal Processing, Sensor Fusion, and Target Recognition XVI, Orlando, Florida, USA, 2007, vol. 6567, p. 65670I

[10] Akbarizadeh, G.: 'A new statistical-based kurtosis wavelet energy feature for texture recognition of SAR images', *IEEE Trans. Geosci. Remote Sens.*, 2012, **50**, (11), pp. 4358–4368

[11] Tirandaz, Z., Akbarizadeh, G.: 'Unsupervised texture-based SAR image segmentation using spectral regression and Gabor filter bank', *J. Indian Soc. Remote Sens.*, 2016, **44**, (2), pp. 177–186

[12] Farbod, M., Akbarizadeh, G., Kosarian, A., *et al.*: 'Optimized fuzzy cellular automata for synthetic aperture radar image edge detection', *J. Electron. Imaging*, 2018, **27**, (1), p. 013030

[13] Dollár, P., Rabaud, V., Cottrell, G., *et al.*: 'Behavior recognition via sparse spatio-temporal features'. Joint IEEE Int. Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, Beijing, 2005, pp. 65–72

[14] Laptev, I., Marszalek, M., Schmid, C., *et al.*: 'Learning realistic human actions from movies'. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Anchorage, Alaska, USA, 2008, pp. 1–8

[15] Klaser, A., Marszałek, M., Schmid, C.: 'A spatio-temporal descriptor based on 3D gradients'. BMVC 2008 - 19th British Machine Vision Conf., Leeds, UK., 2008, 2008, pp. 275:1–275:10

[16] Zhang, J., Li, W., Ogunbona, P.O., *et al.*: 'RGB-D-based action recognition datasets: A survey', *Pattern Recognit.*, 2016, **60**, pp. 86–105

[17] Ye, M., Zhang, Q., Wang, L., *et al.*: 'A survey on human motion analysis from depth data', in Grzegorzek, M., Theobalt, C., Koch, R., *et al.* (Eds.): '*Time-of-flight and depth imaging. Sensors, algorithms, and applications*' (Springer-Verlag, Berlin Heidelberg, 2013), pp. 149–187

[18] Ijjina, E.P., Chalavadi, K.M.: 'Human action recognition in RGB-D videos using motion sequence information and deep learning', *Pattern Recognit.*, 2017, **72**, pp. 504–516

[19] Cruz, L., Lucio, D., Velho, L.: 'Kinect and images: challenges and applications'. Conf. on Graphics, Patterns and Images Tutorials (SIBGRAPI-T), Ouro Preto, Minas Gerais, Brazil, 2012, pp. 36–49

[20] Han, J., Shao, L., Xu, D., *et al.*: 'Enhanced computer vision with Microsoft kinect sensor: a review', *IEEE Trans. Cybern.*, 2013, **43**, (5), pp. 1318–1334

[21] 'ASUS Xtion PRO'. Available at https://www.asus.com/3D-Sensor/Xtion_PRO/, accessed: 30 September 2010

[22] 'Intel RealSense'. Available at https://www.intel.com/content/www/us/en/architecture-and-technology/realsense-overview.html, accessed: 30 September 2010

[23] Shotton, J., Sharp, T., Kipman, A., *et al.*: 'Real-time human pose recognition in parts from single depth images', *Commun. ACM*, 2013, **56**, (1), pp. 116–124

[24] Wang, J., Liu, Z., Wu, Y., *et al.*: 'Mining action let ensemble for action recognition with depth cameras'. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Providence, Rhode Island, USA, 2012, pp. 1290–1297

[25] Xia, L., Chen, C.-C., Aggarwal, J.K.: 'View invariant human action recognition using histograms of 3D joints'. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW), Providence, Rhode Island, USA, 2012, pp. 20–27

[26] Chaudhry, R., Ofli, F., Kurillo, G., *et al.*: 'Bio-inspired dynamic 3D discriminative skeletal features for human action recognition'. IEEE Conf. on Computer Vision and Pattern Recognition Workshops (CVPR), Portland, Oregon, USA, 2013, pp. 471–478

[27] Vemulapalli, R., Arrate, F., Chellappa, R.: 'Human action recognition by representing 3D skeletons as points in a lie group'. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Columbus, Ohio, USA, 2014, pp. 588–595

[28] Ding, W., Liu, K., Fu, X., *et al.*: 'Profile HMMs for skeleton-based human action recognition', *Signal Process., Image Commun.*, 2016, **42**, pp. 109–119

[29] Krizhevsky, A., Sutskever, I., Hinton, G.E.: 'ImageNet classification with deep convolutional neural networks'. Advances in Neural Information Processing Systems (NIPS), Lake Tahoe, USA, 2012, pp. 1097–1105

[30] Karpathy, A., Toderici, G., Shetty, S., *et al.*: 'Large-scale video classification with convolutional neural networks'. Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Columbus, Ohio, USA, 2014, pp. 1725–1732

[31] Szegedy, C., Vanhoucke, V., Ioffe, S., *et al.*: 'Rethinking the inception architecture for computer vision'. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Las Vegas, USA, 2016, pp. 2818–2826

[32] Szegedy, C., Liu, W., Jia, Y., *et al.*: 'Going deeper with convolutions'. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1–9

[33] Telgarsky, M.: 'Benefits of depth in neural networks', arXiv preprint arXiv:1602.04485, 2016

[34] Simonyan, K., Zisserman, A.: 'Very deep convolutional networks for large-scale image recognition', CoRR, abs/1409.1556, 2014. Available at http://arxiv.org/abs/1409.1556

[35] He, K., Zhang, X., Ren, S., *et al.*: 'Deep residual learning for image recognition'. Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Las Vegas, USA, 2016, pp. 770–778

[36] Pham, H.H., Khoudour, L., Crouzil, A., *et al.*: 'Learning and recognizing human action from skeleton movement with deep residual neural networks'. Eighth Int. Conf. of Pattern Recognition Systems (ICPRS 2017), Madrid, Spain, 2017, pp. 1–6

[37] Li, W., Zhang, Z., Liu, Z.: 'Action recognition based on a bag of 3D points'. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW), San Francisco, USA, 2010, pp. 9–14

[38] Shahroudy, A., Liu, J., Ng, T.-T., *et al.*: 'NTU RGB+D: a large scale dataset for 3D human activity analysis'. Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Las Vegas, USA, 2016, pp. 1010–1019

[39] Yoon, B.-J.: 'Hidden Markov models and their applications in biological sequence analysis', *Curr. Genomics*, 2009, **10**, (6), pp. 402–415

[40] Lv, F., Nevatia, R.: 'Recognition and segmentation of 3-D human action using HMM and multi-class AdaBoost'. European Conf. on Computer Vision (ECCV), Graz, Austria, 2006, pp. 359–372

[41] Wu, D., Shao, L.: 'Leveraging hierarchical parametric networks for skeletal joints based action segmentation and recognition'. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Columbus, Ohio, USA, 2014, pp. 724–731

[42] Luo, J., Wang, W., Qi, H.: 'Group sparsity and geometry constrained dictionary learning for action recognition from depth maps'. Proc. IEEE Int. Conf. on Computer Vision (ICCV), Sidney, Australia, 2013, pp. 1809–1816

[43] Hochreiter, S., Schmidhuber, J.: 'Long short-term memory', *Neural Comput.*, 1997, **9**, (8), pp. 1735–1780

[44] Graves, A.: 'Supervised sequence labelling with recurrent neural networks', *Studies in Computational Intelligence*, **vol 385**, (Springer-Verlag, Berlin Heidelberg, 2008), 14 pp.

[45] Du, Y., Wang, W., Wang, L.: 'Hierarchical recurrent neural network for skeleton based action recognition'. Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Boston, Massachusetts, USA, 2015, pp. 1110–1118

[46] Zhu, W., Lan, C., Xing, J., *et al.*: 'Co-occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks', arXiv preprint arXiv:1603.07772, 2016

[47] Liu, J., Shahroudy, A., Xu, D., *et al.*: 'Spatio-temporal LSTM with trust gates for 3D human action recognition'. European Conf. on Computer Vision (ECCV), Amsterdam, The Netherlands, 2016, pp. 816–833

[48] Sainath, T.N., Vinyals, O., Senior, A.W., *et al.*: 'Convolutional, long short-term memory, fully connected deep neural networks'. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), Brisbane, Australia, 2015, pp. 4580–4584

[49] Wang, P., Li, Z., Hou, Y., *et al.*: 'Action recognition based on joint trajectory maps using convolutional neural networks'. Proc. 2016 ACM on Multimedia Conf., Amsterdam, The Netherlands, 2016, pp. 102–106

[50] Hou, Y., Li, Z., Wang, P., *et al.*: 'Skeleton optical spectra based action recognition using convolutional neural networks', *IEEE Trans. Circuits Syst. Video Technol.*, 2018, **28**, (3), pp. 807–811

[51] Davison, A.J.: 'Real-time simultaneous localisation and mapping with a single camera'. Proc. Ninth IEEE Int. Conf. on Computer Vision – Volume 2, ICCV '03, Nice, France, 2003

[52] Shotton, J., Fitzgibbon, A., Cook, M., *et al.*: 'Real-time human pose recognition in parts from single depth images'. Computer Vision and Pattern Recognition (CVPR), Colorado Springs, USA, 2011, pp. 1297–1304

[53] Yacoob, Y., Black, M.J.: 'Parameterized modeling and recognition of activities', *Comput. Vis. Image Underst.*, 1999, **73**, (2), pp. 232–247

[54] Chaudhry, R., Ofli, F., Kurillo, G., *et al.*: 'Bio-inspired dynamic 3d discriminative skeletal features for human action recognition'. Proc. IEEE Conf. on Computer Vision and Pattern Recognition Workshops (CVPR), Portland, Oregon, USA, 2013, pp. 471–478

[55] Microsoft: 'Kinect for windows – human interface guidelines v2.0'. Technical report, 2014

[56] He, K., Sun, J.: 'Convolutional neural networks at constrained time cost'. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Boston, Massachusetts, USA, 2015, pp. 5353–5360

[57] Krizhevsky, A.: 'Learning multiple layers of features from tiny images'. Technical report, 2009

[58] Russakovsky, O., Deng, J., Su, H., *et al.*: 'ImageNet large scale visual recognition challenge', *Int. J. Comput. Vis.*, 2015, **115**, (3), pp. 211–252

[59] Ioffe, S., Szegedy, C.: 'Batch normalization: accelerating deep network training by reducing internal covariate shift'. Int. Conf. on Machine Learning (ICML), Lille, France, 2015, pp. 448–456
[60] Nair, V., Hinton, G.E.: 'Rectified linear units improve restricted Boltzmann machines'. Proc. 27th Int. Conf. on machine learning (ICML-10), Haifa, Israel, 2010, pp. 807–814
[61] Srivastava, N., Hinton, G.E., Krizhevsky, A., *et al.*: 'Dropout: a simple way to prevent neural networks from overfitting', *J. Mach. Learn. Res.*, 2014, **15**, pp. 1929–1958
[62] Bottou, L.: '*Large-scale machine learning with stochastic gradient descent*' (Physica-Verlag HD, Heidelberg, 2010), pp. 177–186. Available at https://doi.org/10.1007/978-3-7908-2604-3_16
[63] Vedaldi, A., Lenc, K.: 'MatConvNet: convolutional neural networks for MATLAB'. Proc. 23rd ACM Int. Conf. on Multimedia, Brisbane, Australia, 2015, pp. 689–692
[64] Vieira, A.W., Nascimento, E.R., Oliveira, G.L., *et al.*: 'Stop: space-time occupancy patterns for 3D action recognition from depth map sequences'. Iberoamerican Congress on Pattern Recognition, Buenos Aires, Argentina, 2012, pp. 252–259
[65] Chaaraoui, A., Padilla-Lopez, J., Flórez-Revuelta, F.: 'Fusion of skeletal and silhouette-based features for human action recognition with RGB-D devices'. Proc. IEEE Int. Conf. on Computer Vision Workshops, Portland, Oregon, USA, 2013, pp. 91–97
[66] Chen, C., Liu, K., Kehtarnavaz, N.: 'Real-time human action recognition based on depth motion maps', *J. Real-Time Image Process.*, 2016, **12**, (1), pp. 155–163
[67] Gowayyed, M.A., Torki, M., Hussein, M.E., *et al.*: 'Histogram of oriented displacements (HOD): describing trajectories of human joints for action recognition'. Proc. 23rd Int. Joint Conf. on Artificial Intelligence, IJCAI '13, Beijing, China, 2013, pp. 1351–1357
[68] Hussein, M.E., Torki, M., Gowayyed, M.A., *et al.*: 'Human action recognition using a temporal hierarchy of covariance descriptors on 3D joint locations'. Proc. 23rd Int. Joint Conf. on Artificial Intelligence (IJCAI'13), Beijing, China, 2013, pp. 2466–2472
[69] Qin, S., Yang, Y., Jiang, Y.: 'Gesture recognition from depth images using motion and shape features'. Int. Symp. on Instrumentation and Measurement, Sensor Network and Automation (IMSNA), Toronto, Canada, 2013, pp. 172–175
[70] Liang, B., Zheng, L.: 'Three dimensional motion trail model for gesture recognition'. Proc. IEEE Int. Conf. on Computer Vision (ICCV) Workshops, Nice, France, 2013, pp. 684–691
[71] Evangelidis, G., Singh, G., Horaud, R.: 'Skeletal quads: human action recognition using joint quadruples'. Int. Conf. on Pattern Recognition (ICPR), Stockholm, Sweden, 2014, pp. 4513–4518
[72] Theodorakopoulos, I., Kastaniotis, D., Economou, G., *et al.*: 'Pose-based human action recognition via sparse representation in dissimilarity space', *J. Vis. Commun. Image Represent.*, 2014, **25**, (1), pp. 12–23
[73] Gao, Z., Song, J.-M., Zhang, H., *et al.*: 'Human action recognition via multi-modality information', *J. Electr. Eng. Technol.*, 2014, **9**, (2), pp. 739–748
[74] Vieira, A.W., Nascimento, E.R., Oliveira, G.L., *et al.*: 'On the improvement of human action recognition from depth map sequences using space–time occupancy patterns', *Pattern Recognit. Lett.*, 2014, **36**, pp. 221–227
[75] Chen, C., Jafari, R., Kehtarnavaz, N.: 'Action recognition from depth sequences using depth motion maps-based local binary patterns'. IEEE Winter Conf. on Applications of Computer Vision (WACV), Hawai, USA, 2015, pp. 1092–1099
[76] Xu, H., Chen, E., Liang, C., *et al.*: 'Spatio-temporal pyramid model based on depth maps for action recognition'. IEEE Int. Workshop on Multimedia Signal Processing (MMSP), Xiamen, China, 2015, pp. 1–6
[77] Jin, K., Min, J., Kong, J., *et al.*: 'Action recognition using vague division depth motion maps', *J. Eng.*, 2017, **1**, (1), pp. 77–84
[78] Li, X., Zhang, Y., Zhang, J.: 'Improved key poses model for skeleton-based action recognition', in Zeng, B., Huang, Q., El Saddik, A., *et al.* (Eds.): '*Advances in multimedia information processing – PCM 2017*' (Springer International Publishing, Cham, 2018), pp. 358–367
[79] Luvizon, D.C., Tabia, H., Picard, D.: 'Learning features combination for human action recognition from skeleton sequences', *Pattern Recognit. Lett.*, 2017, **99**, pp. 13–20
[80] Hbali, Y., Hbali, S., Ballihi, L., *et al.*: 'Skeleton-based human activity recognition for elderly monitoring systems', *IET Comput. Vis.*, 2017, **12**, (1), pp. 16–26
[81] Oreifej, O., Liu, Z.: 'HON4D: histogram of oriented 4D normals for activity recognition from depth sequences'. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Portland, Oregon, USA, 2013, pp. 716–723
[82] Zanfir, M., Leordeanu, M., Sminchisescu, C.: 'The moving pose: an efficient 3d kinematics descriptor for low-latency action recognition and detection'. Proc. IEEE Int. Conf. on Computer Vision, Portland, Oregon, USA, 2013, pp. 2752–2759
[83] Yang, X., Tian, Y.: 'Super normal vector for activity recognition using depth sequences'. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Columbus, Ohio, USA, 2014, pp. 804–811
[84] Ohn-Bar, E., Trivedi, M.M.: 'Joint angles similarities and HOG2 for action recognition'. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) Workshops, Portland, Oregon, USA, 2013, pp. 465–470
[85] Misra, I., Zitnick, C.L., Hebert, M.: 'Shuffle and learn: unsupervised learning using temporal order verification'. European Conf. on Computer Vision (ECCV), Amsterdam, The Netherlands, 2016, pp. 527–544
[86] Cippitelli, E., Gambi, E., Spinsante, S., *et al.*: 'Evaluation of a skeleton-based method for human activity recognition on a large-scale RGB-D dataset'. Int. Conf. on Technologies for Active and Assisted Living (TechAAL 2016), London, UK, 2016, pp. 1–6
[87] Vemulapalli, R., Chellappa, R.: 'Rolling rotations for recognizing human actions from 3D skeletal data'. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Las Vegas, USA, 2016, pp. 4471–4479
[88] Luo, Z., Peng, B., Huang, D.-A., *et al.*: 'Unsupervised learning of long-term motion dynamics for videos'. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Hawai, USA, 2017, pp. 7101–7110
[89] Dietterich, T.G.: 'Ensemble methods in machine learning'. Int. Workshop on Multiple Classifier Systems, Cagliari, Italy, 2000, pp. 1–15
[90] Szegedy, C., Ioffe, S., Vanhoucke, V., *et al.*: 'Inception-v4, inception-ResNet and the impact of residual connections on learning'. AAAI Conf. on Artificial Intelligence, San Francisco, USA, 2017, pp. 4278–4284
[91] Huang, G., Liu, Z., Weinberger, K.Q.: 'Densely connected convolutional networks'. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Hawai, USA, 2017, pp. 2261–2269
[92] Cao, Z., Simon, T., Wei, S.-E., *et al.*: 'Realtime multi-person 2D pose estimation using part affinity fields'. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Hawai, USA, 2017
[93] Kokkinos, I., Guler, R.A., Neverova, N.: 'DensePose: dense human pose estimation in the wild'. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, USA, 2018

9 Appendix

This section describes the network architectures in detail. To build 20-layer, 32-layer, 44-layer, 56-layer, and 110-layer networks, we stack the proposed ResNet building units as follows (see Tables 7–11).

Table 7 Baseline 20-layer ResNet architecture

3 × 3 Conv., 16 filters, BN, ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 16 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 16 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 16 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 32 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 32 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 32 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 64 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 64 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 64 filters-⊕-ReLU
Global mean pooling
fully convolutional (FC) layer with *n* units where *n* is equal the number of action class.
Softmax layer

Table 8 Baseline 32-layer ResNet architecture

3 × 3 Conv., 16 filters, BN, ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 16 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 16 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 16 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 16 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 16 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 32 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 32 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 32 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 32 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 32 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 64 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 64 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 64 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 64 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 64 filters-⊕-ReLU
Global mean pooling
FC layer with *n* units where *n* is equal the number of action class.
Softmax layer

Table 9 Baseline 44-layer ResNet architecture

3 × 3 Conv., 16 filters, BN, ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 16 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 16 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 16 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 16 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 16 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 16 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 16 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 32 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 32 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 32 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 32 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 32 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 32 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 32 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 64 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 64 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 64 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 64 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 64 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 64 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 64 filters-⊕-ReLU
Global mean pooling
FC layer with *n* units, where *n* is equal the number of action class.
Softmax layer

Table 10 Baseline 56-layer ResNet architecture

3 × 3 Conv., 16 filters, BN, ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 16 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 16 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 16 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 16 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 16 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 16 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 16 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 16 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 16 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 32 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 32 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 32 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 32 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 32 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 32 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 32 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 32 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 32 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 64 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 64 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 64 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 64 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 64 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 64 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 64 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 64 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 64 filters-⊕-ReLU
Global mean pooling
FC layer with *n* units, where *n* is equal the number of action class.
Softmax layer

Table 11 Baseline 110-layer ResNet architecture

3 × 3 Conv., 16 filters, BN, ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 16 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 16 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 16 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 16 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 16 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 16 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 16 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 16 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 16 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 16 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 16 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 16 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 16 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 16 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 16 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 16 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 16 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 16 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 32 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 32 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 32 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 32 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 32 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 32 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 32 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 32 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 32 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 32 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 32 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 32 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 32 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 32 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 32 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 32 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 32 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 32 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 64 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 64 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 64 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 64 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 64 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 64 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 64 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 64 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 64 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 64 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 64 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 64 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 64 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 64 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 64 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 64 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 64 filters-⊕-ReLU
Residual unit: Conv.-BN-ReLU-Dropout-Conv., 64 filters-⊕-ReLU
Global mean pooling
FC layer with *n* units, where *n* is equal the number of action class.
Softmax layer