Information Sciences 513 (2020) 112–126

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Image representation of pose-transition feature for 3D skeleton-based action recognition

Thien Huynh-The[a,*], Cam-Hao Hua[c], Trung-Thanh Ngo[b], Dong-Seong Kim[a,b,*]

[a] ICT Convergence Research Center, Kumoh National Institute of Technology, 61, Daehak-ro, Gumi-si, Gyeongsangbuk-do 39177, Republic of Korea

[b] Department of IT Convergence Engineering, Kumoh National Institute of Technology, 61, Daehak-ro, Gumi-si, Gyeongsangbuk-do 39177, Republic of Korea

[c] Department of Computer Science & Engineering, Kyung Hee University (Global Campus), 1732 Deokyoungdae-ro, Giheung-gu, Yongin-si, Gyeonggi-do 446-701, Republic of Korea

ARTICLE INFO

Article history:
Received 29 December 2018
Revised 9 August 2019
Accepted 25 October 2019
Available online 5 November 2019

Keywords:
Pose-transition feature to image (PoT2I) encoding technique
Depth camera
Human action recognition
Deep convolutional neural networks

ABSTRACT

Recently, skeleton-based human action recognition has received more interest from industrial and research communities for many practical applications thanks to the popularity of depth sensors. A large number of conventional approaches, which have exploited handcrafted features with traditional classifiers, cannot learn high-level spatiotemporal features to precisely recognize complex human actions. In this paper, we introduce a novel encoding technique, namely Pose-Transition Feature to Image (PoT2I), to transform skeleton information to image-based representation for deep convolutional neural networks (CNNs). The spatial joint correlations and temporal pose dynamics of an action are exhaustively depicted by an encoded color image. For learning action models, we fine-tune end-to-end a pre-trained network to thoroughly capture multiple high-level features at multi-scale action representation. The proposed method is benchmarked on several challenging 3D action recognition datasets (e.g., UTKinect-Action3D, SBU-Kinect Interaction, and NTU RGB+D) with different parameter configurations for performance analysis. Outstanding experimental results with the highest accuracy of 90.33% on the most challenging NTU RGB+D dataset demonstrate that our action recognition method with PoT2I outperforms state-of-the-art approaches.

1. Introduction

Human action understanding and analysis have received considerable attention from research and industrial communities for multiple areas of applications as daily living assistant, human-robot interaction, video-based monitoring, and surveillance. In the last decade, although a large number of advanced works of human action recognition has been studied for performance improvement (wherein meaningful action information extracted from color video sequences is analyzed to understand the body joint relations and action dynamics in the space-time domain), several principal challenges in the computer vision domain are still not addressed thoroughly (for instance, illumination change, variations of background environment

* Corresponding author.
E-mail addresses: thienht@kumoh.ac.kr, thien.huynh-the.2016@ieee.org (T. Huynh-The), hao.hua@oslab.khu.ac.kr (C.-H. Hua), thanh-ngo-ua@naver.com (T.-T. Ngo), dskim@kumoh.ac.kr (D.-S. Kim).

https://doi.org/10.1016/j.ins.2019.10.047
0020-0255/

[30] and subject appearance [39]). Besides that, conventional visual feature descriptors, such as Histogram of Gradient (HoG), Scale-Invariant Feature Transform (SIFT), restricted the capability of informative representation in the practical environment. Recently, with the innovation of depth sensor technologies, human action recognition has achieved several remarkable results of accuracy, wherein essential advantages of depth camera (for example, accurate object detection and human pose estimation) are fully exploited [7,9]. Obviously, thanks to the convenience of data retrieval and the robustness of 3D skeleton information, the approaches of action analysis and understanding are modernized completely.

Based on the skeleton information, an action can be recognized by the analysis of spatial joint relations and temporal posture dynamics. Most of existing conventional approaches have adopted handcrafted features with traditional classifiers for recognizing simple actions, but nevertheless their performance is trivial due to shallow features leading to weak discrimination. For example, the joint trajectory extracted from the skeleton data for formulating the posture movement in the temporal dimension cannot capture the latent features of an action comprehensively. Furthermore, conventional classification techniques (e.g., Decision Tree, k-Nearest Neighbor, Support Vector Machine) seem to be bounded with large-scale challenging actions in the practical environment (for instance, the variation of single action, human-object interaction, and human-human interaction under the performance of various subjects). As a primary weakness, they cannot deal with a huge amount of data effectively. Lately, numerous approaches based on recurrent neural networks (RNNs) and long short-term memory (LSTM) network have been introduced for learning hierarchical features from the raw skeleton data. Despite remarkable efforts for accuracy improvement, RNNs- and LSTM-based learning models have to face with the increment of network complexity (i.e., a vast amount number of input features) and network overfitting besides an unable learning high-level spatiotemporal features.

In this paper, we present an efficient method of action recognition using 3D skeleton data of depth camera, wherein a deep convolutional neural network (DCNN) is exploited to learn the comprehensive information of human pose and action dynamic. As the major contribution, the proposed action recognition method has two components: (i) feature to image encoding and (ii) action learning with deep model. In concrete, we firstly introduce a novel encoding technique, called Pose-Transition Feature to Image (PoT2I), for transforming skeleton information to image data. Instead of converting skeleton coordinates to grayscale or color images directly as in several existing works, we consider higher relative features which are capable of thoroughly representing spatial posture and temporal motions. To this end, the joint-joint distance and orientation feature of two arbitrary joints within a frame and between two consecutive frames are extracted by an intensity conversion function for encoding to color pixels. Correspondingly, an action image, which is generated from a skeleton sequence by gathering all features and arranging them in the temporal dimension, is capable of portraying a whole action appearance. We then end-to-end fine-tune a pre-trained Inception-v3 [28] network model for learning multiple high-level features of posture and motion. The spatiotemporal correlations of action appearance are profoundly captured by implementing convolution and pooling operations at multi-level feature maps. Compared with the state-of-the-art approaches, including the handcrafted feature-based, RNNs-based, and CNNs-based, our method with the proposed encoding technique achieves significant improvement of recognition accuracy.

In brief, main contributions of this paper are summarized as follows

- An efficient 3D skeleton-based action recognition method is presented by studying a novel feature encoding technique coupled with deep learning models.
- Pose-Transition Feature to Image (PoT2I), a novel encoding technique is proposed to transform the high-level relative features of human pose and transition, which are extracted from skeletal coordinate data, to color pixels. Accordingly, all meaningful information extracted in a sequence is presented by an encoded action image.
- A pre-trained Inception-v3 model is fine-tuned end-to-end to gain deep visual features of action images for action recognition. Thanks to the powerful and informative features obtained from a large-scale color image set, adopting a pre-trained network with transfer learning is much better than training a network from scratch in terms of simplicity and training convergence.
- We benchmark the proposed method on three well-known challenging datasets (UTKinect-Action3D, SBU-Kinect Interaction, and NTU RGB+D), wherein our method mostly outperforms other state-of-the-art skeleton-based action recognition approaches in terms of accuracy. In addition, the detailed analysis of performance under different experimental configurations is further given.

The organization of the rest of this paper is as follows. State-of-the-art skeleton-based action recognition methods are summarized in Section 2. Section 3 presents the proposed method, where the efficient encoding technique PoT2I is completely described. The experiments for performance evaluation are provided in Section 4, where the analysis and comparison with other methods are comprehensively conducted. The conclusion and future work are finally given in Section 5.

2. Related works

Instead of covering all ideas of visual-based human action recognition, we pay attention to existing work that exploits 3D skeleton information in the perspectives of feature engineering and action model learning. Accordingly, several papers, which have studied handcrafted feature descriptors from conventional to advanced schemes, are discussed intensively due to the association with our contribution of feature encoding. Additionally, many state-of-the-art deep learning-based approaches are carefully studied, wherein several modern deep models are developed to enrich high-level features and classify actions.

2.1. Handcrafted feature-based action recognition

In the last decade, a considerable amount of works has been proposed for 3D human action analysis and recognition by exploiting handcrafted features with traditional classification techniques [38,41]. One of the most widely used features for representing action dynamics is skeleton trajectory. By modeling body movements as the elements of an extraordinary Euclidean group SE(3), Vemulapalli et al. [32] learned each human action as a curve of a Lie group for classification using Support Vector Machine (SVM). As the weakness of this study, the interactions between specific categories of body parts are abandoned to characterize some multi-person interactive actions. In another work [1], skeleton trajectory was comprehensively depicted in a Kendalls shape manifold to effectively address the data corruption issue caused by the variation of execution rates within and across subjects. Transforming trajectories into the transported square-root vector fields tackles the variation of action execution rates within a subject for single actions and across multi-subjects of interactions. Despite conducting some performance improvements by this approach, three critical drawbacks have to be taken into account: (i) fragility under noise of skeleton data, (ii) restriction to short-term actions, and (iii) confusion between similar actions. Besides the trajectory feature, several existing works exploited other advanced features, such as Skeleton Quads [5], EigenJoint [40], structured streaming skeleton [44], multimodal multipart [22], and pose-transition structures [8,10]. For example, Huynh-The et al. [8] presented an action by a feature set of joint-joint, joint-plane distance, and orientation. All these geometric features are categorized into the pose class (features extracted within a frame for describing the current human posture) and the transition class (features captured in two adjacent frames for representing the action movement) before they are encoded separately by k-means clustering using two codebooks. Although the action recognizer in this work is exhaustively learned by Pachinko Allocation Model, an advanced hierarchical topic model, it confuses in discriminating overlapping-in-time interactive actions, where the beginning and ending parts of two different actions are similar (e.g., *approaching* vs. *departing*). In addition, there are some other ways to delineate human action from 3D joint data, for example, encoding sparse skeleton features to another representative space using Grassmann manifold [24], joint spatial graph [16], Markov random field [2], and mapping skeleton data to such multiple geometric viewpoints [3]. Based on several advantages of easy implementation and obvious visualization, handcrafted features are usually combined along with conventional machine learning methods for small-scale datasets, but nevertheless their shallowness cannot discriminate complex actions in large-scale datasets.

2.2. Deep learning-based action recognition

Recently, deep learning has been studied for many applications of the image processing and computer vision areas due to its impressive power compared with traditional machine learning techniques. RNN and LSTM network (a.k.a., an extension of RNN) have demonstrated their strength for skeleton-based action recognition in numerous existing work. A hierarchical RNN architecture [4] is developed for learning skeleton sequences temporally, wherein five subnets are correspondingly designed for learning the motion of five body parts. Wang et al. [34] captured the spatiotemporal contextual information from skeleton data by a novel two-stream RNNs architecture, wherein a stacked RNN and a hierarchical RNN are built for learning the spatial-dependency of joints. To address the problem of incompetently learning spatiotemporal dynamics, Veerial et al. [31] introduced a differential gating scheme for LSTM networks, in which the informative change of salient motions between two and more successive frames are exposed sufficiently. Similarly, Shahroudy et al. [21] developed a part-ware LSTM model to take advantages of long-term context correlations between features of different body parts. With a novel gating mechanism, a spatiotemporal LSTM network presented by Liu et al. [17] is able to learn the reliable skeleton data sequentially and also to update the long-term context information selectively. Liu et al. [18] designed Global Context-Aware Attention LSTM (GCA-LSTM) to selectively learn the most discriminative joints from the global contextual information for accuracy improvement. Wang et al. [33] handled the variety of action appearance by incorporating three specific layers (i.e., for beginning joint projection, viewpoint transformation, and spatial dropping out) into a standard RNN. This research was extended in [35] by learning primitive geometric features (e.g., joint, edge, and surface) instead of raw skeleton data. In [42], Zhang et al. combined two LSTM networks in series for human action analysis, in which the first one is developed for adaptively transforming viewpoints to address the variety of action appearance. Ensemble Temporal Sliding LSTM (TS-LSTM) [14], a multiple-term LSTM networks architecture, has the ability to apprehend various temporal dependencies among multiple body parts. Some other approaches have considered different kinds of multiple LSTM networks architecture for acquiring discriminative features [25], fusing multi-geometric feature learning models [36,43], and maintaining temporal information [29]. Although both RNNs and LSTM are able to model short- and long-term actions with performance improvement compared with traditional learning approaches, they directly work with raw skeleton coordinates as an input, that conduct shallow distinctness of action classification. Additionally, the high dimension of input feature rapidly amplifies the network complexity and may cause overfitting as well.

Lately, several approaches have taken advantage of CNNs for 3D human action recognition. CNNs are designed to proficiently work with highly structured modalities [26] (a.k.a., highly dimensional data such as image or video), therefore, most of the skeleton-based approaches study an efficient encoding technique for data transformation (i.e., from time serial data to image form). For example, the input of Residual Temporal CNN (Res-TCN) [13] is the image which is transformed from a concatenation of all frame-wise features. Liu et al. [19] encoded skeleton information (characterized by coordinates, time label, and joint label) to color pixels before learning action model by a two-stream deep CNNs architecture. In [15],

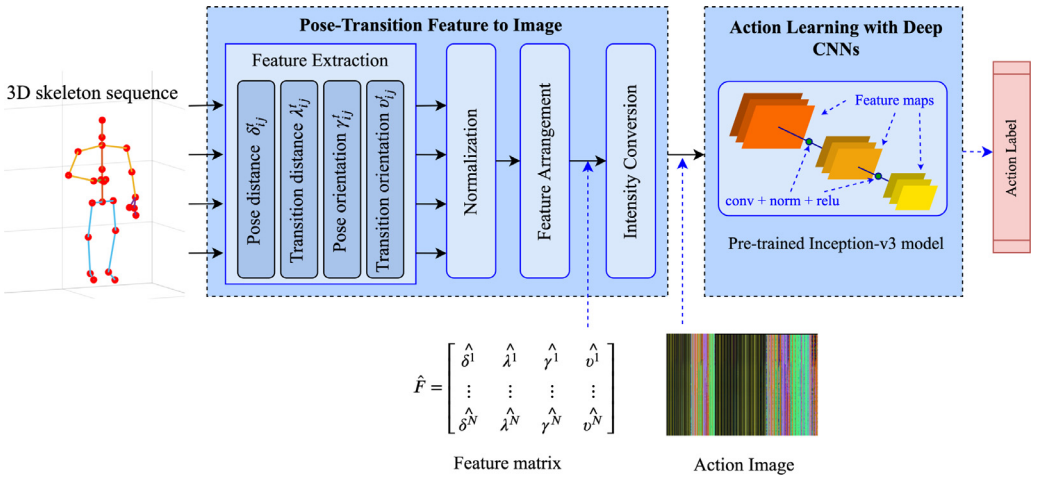

Fig. 1. The overview of our proposed method for 3D action recognition with an encoder for transforming skeleton data to image-form data.

the new coordinates of all body joints, obtained by joint reference and projection, are transformed to an image for fine-tuning the pre-trained VGG-19 model [23]. Ke et al. [11] converted such robust cosine distance to grayscale image for action model learning with a simple CNN (where parameters in the network are initialized from a pre-trained VGG model). By encoding 3D joint coordinates of an action sample following an axis channel scheme into fixed-length clips [12], the spatiotemporal correlations within a frame and also across multiple frames are extensively acquired. Additionally, a multitask convolutional neural network (MTCNN), which has several flows of convolutional stacks, is designed with the pre-trained weights of VGG-19. By representing low-level skeleton feature under image form, Pham et al. [20] achieved a remarkable recognition accuracy with Inception-ResNet [27]. Similarly, the joint trajectories and posture dynamics (e.g., joint motion direction, motion magnitude, and selective body parts) are transformed into 2D color image [37]. Accordingly, a multiple ConvNets-based architecture is fine-tuned from the pre-trained AlexNet model for classification task. Based on the capability of multiple high-level features at multi-scale representations, most of CNN-based action recognition approaches reach an impressive performance in terms of accuracy. However, encoding directly raw skeleton data to image is fragile due to the variation of action appearance.

3. Methodology

Very recently, some 3D action recognition methods have studied mechanisms of depicting skeleton information by image-based representation for modern CNNs-based learning. Compared with LSTM-based approaches, CNNs take advantage of high-level features learning for overall performance improvement. Inspired by the success of CNNs for classification tasks, we introduce an efficient 3D action recognition method, in which a novel technique for encoding pose and transition features to color images is proposed to gain highly distinctive joint correlations. For learning classification model from action images, we fine-tune a pre-trained Inception-v3 network model completely. The overall architecture of our proposed action recognition method is shown in Fig. 1.

3.1. Pose-transition feature to image encoding technique

Different from other techniques where joint coordinates are mapped directly into grayscale values, we introduce a novel encoding technique, namely Pose-Transition Feature to Image (PoT2I), which includes the feature extraction, feature arrangement, and action image generation processes.

3.1.1. Pose and transition feature extraction

In this research, we calculate two geometric features (i.e., joint-joint distance and joint-joint orientation), in which the features extracted from two arbitrary joints within a frame are called pose features and the features extracted from two arbitrary joints of two consecutive frames are called transition features. Given a skeleton frame F consisting of m 3D joints, where each joint is defined by 3D coordinate (x, y, z) in space $\Re^3$. Let p^t_i and p^t_j refer to as the 3D coordinates of the ith and jth joints of a skeleton in frame F^t, respectively. The joint-joint Euclidean distance feature δ^t_{ij} between p^t_i and p^t_j are determined by projecting 3D points on three planes Oxy, Oyz, and Ozx, where $z = 0$, $x = 0$, and $y = 0$, respectively. The joint-joint distance is calculated for pose feature on the plane $z = 0$ by

$$\delta^{t,z=0}_{ij} = \left\| p(x,y)^t_i - p(x,y)^t_j \right\|, \tag{1}$$

on the plane $x = 0$ by

$$\delta_{ij}^{t,x=0} = \left\| p(y,z)_i^t - p(y,z)_j^t \right\|, \tag{2}$$

and on the plane $y = 0$ by

$$\delta_{ij}^{t,y=0} = \left\| p(z,x)_i^t - p(z,x)_j^t \right\|. \tag{3}$$

The pose distance feature Δ of the ith and jth joints ($i \neq j$) is formed as $\Delta_{ij}^t = \left[\delta_{ij}^{t,z=0}, \delta_{ij}^{t,x=0}, \delta_{ij}^{t,y=0}\right]$. The joint-joint distance is calculated for transition feature on the plane $z = 0$ by

$$\lambda_{ij}^{t,z=0} = \left\| p(x,y)_i^t - p(x,y)_j^{t-1} \right\|, \tag{4}$$

on the plane $x = 0$ by

$$\lambda_{ij}^{t,x=0} = \left\| p(y,z)_i^t - p(y,z)_j^{t-1} \right\|, \tag{5}$$

and on the plane $y = 0$ by

$$\lambda_{ij}^{t,y=0} = \left\| p(z,x)_i^t - p(z,x)_j^{t-1} \right\|. \tag{6}$$

The transition distance feature Λ of the ith and jth joints is formed as $\Lambda_{ij}^t = \left[\lambda_{ij}^{t,z=0}, \lambda_{ij}^{t,x=0}, \lambda_{ij}^{t,y=0}\right]$.

Another geometric feature is the joint-joint orientation between the joint-joint vector $\overrightarrow{u^t}$ versus three unit vectors of three corresponding original axes (particularly, the horizontal axis $\overrightarrow{Ox} = \langle 1,0,0 \rangle$, the vertical axis $\overrightarrow{Oy} = \langle 0,1,0 \rangle$, and the depth axis $\overrightarrow{Oz} = \langle 0,0,1 \rangle$). The joint-joint pose orientation is determined as follows

$$\begin{aligned}
\gamma_{ij}^{t,\overrightarrow{Ox}} &= \measuredangle\left(\overrightarrow{u^t}, \overrightarrow{Ox}\right) = \cos^{-1}\left(\frac{\overrightarrow{u^t} \cdot \overrightarrow{Ox}}{\left\|\overrightarrow{u^t}\right\| \times \left\|\overrightarrow{Ox}\right\|}\right), \\
\gamma_{ij}^{t,\overrightarrow{Oy}} &= \measuredangle\left(\overrightarrow{u^t}, \overrightarrow{Oy}\right) = \cos^{-1}\left(\frac{\overrightarrow{u^t} \cdot \overrightarrow{Oy}}{\left\|\overrightarrow{u^t}\right\| \times \left\|\overrightarrow{Oy}\right\|}\right), \\
\gamma_{ij}^{t,\overrightarrow{Oz}} &= \measuredangle\left(\overrightarrow{u^t}, \overrightarrow{Oz}\right) = \cos^{-1}\left(\frac{\overrightarrow{u^t} \cdot \overrightarrow{Oz}}{\left\|\overrightarrow{u^t}\right\| \times \left\|\overrightarrow{Oz}\right\|}\right),
\end{aligned} \tag{7}$$

where $\overrightarrow{u^t} = \left\langle x_i^t - x_j^t, y_i^t - y_j^t, z_i^t - z_j^t \right\rangle$. The pose orientation feature Γ is arranged as $\Gamma_{ij}^t = \left[\gamma_{ij}^{t,\overrightarrow{Ox}}, \gamma_{ij}^{t,\overrightarrow{Oy}}, \gamma_{ij}^{t,\overrightarrow{Oz}}\right]$, where $(i \neq j)$. Similar to distance feature, the joint-joint orientation for portraying pose dynamics is also estimated on as follows

$$\begin{aligned}
\upsilon_{ij}^{t,\overrightarrow{Ox}} &= \measuredangle\left(\overrightarrow{v^t}, \overrightarrow{Ox}\right) = \cos^{-1}\left(\frac{\overrightarrow{v^t} \cdot \overrightarrow{Ox}}{\left\|\overrightarrow{v^t}\right\| \times \left\|\overrightarrow{Ox}\right\|}\right), \\
\upsilon_{ij}^{t,\overrightarrow{Oy}} &= \measuredangle\left(\overrightarrow{v^t}, \overrightarrow{Oy}\right) = \cos^{-1}\left(\frac{\overrightarrow{v^t} \cdot \overrightarrow{Oy}}{\left\|\overrightarrow{v^t}\right\| \times \left\|\overrightarrow{Oy}\right\|}\right), \\
\upsilon_{ij}^{t,\overrightarrow{Oz}} &= \measuredangle\left(\overrightarrow{v^t}, \overrightarrow{Oz}\right) = \cos^{-1}\left(\frac{\overrightarrow{v^t} \cdot \overrightarrow{Oz}}{\left\|\overrightarrow{v^t}\right\| \times \left\|\overrightarrow{Oz}\right\|}\right),
\end{aligned} \tag{8}$$

where $\overrightarrow{v^t} = \left\langle x_i^t - x_j^{t-1}, y_i^t - y_j^{t-1}, z_i^t - z_j^{t-1} \right\rangle$. The transition orientation feature Υ is formed as $\Upsilon_{ij}^t = \left[\upsilon_{ij}^{t,\overrightarrow{Ox}}, \upsilon_{ij}^{t,\overrightarrow{Oy}}, \upsilon_{ij}^{t,\overrightarrow{Oz}}\right]$. Typically, all distance and orientation features are extracted from one skeleton that represents single actions performed by one actor. For the person-person interactions, we extract the features of the ith and jth joints of two different person-belonging skeletons (i.e., $i \in F_X^t$ and $j \in F_Y^t$ for the case of pose features, and $i \in F_X^t$ and $j \in F_Y^{t-1}$ for the case of transition features, where X and Y refer to as two different persons). Because the distance and orientation features are different in unit, value normalization should be done before transforming pose-transition features to chromatic RGB values. This normalization processing is applied separately for distance and orientation features by the following equations

$$\widehat{D_{ij}^t} = \frac{D_{ij}^t - \min(D)}{\max(D) - \min(D)}, \tag{9}$$

$$\widehat{O_{ij}^t} = \frac{O_{ij}^t - \min(O)}{\max(O) - \min(O)}, \tag{10}$$

where $D_{ij}^t = \left\{\Delta_{ij}^t, \Lambda_{ij}^t\right\}$ refers to as the set of pose and transition distance features; and $O_{ij}^t = \left\{\Gamma_{ij}^t, \Upsilon_{ij}^t\right\}$ refers to as the set of pose and transition orientation features. Correspondingly, the normalized features are formed as follows $\widehat{D_{ij}^t} = \left\{\widehat{\Delta_{ij}^t}, \widehat{\Lambda_{ij}^t}\right\}$ and $\widehat{O_{ij}^t} = \left\{\widehat{\Gamma_{ij}^t}, \widehat{\Upsilon_{ij}^t}\right\}$.

3.1.2. Feature arrangement

All normalized pose-transition distance and orientation features of a skeleton frame are concatenated into a one-row matrix, called mixture feature, as follows

$$\widehat{F^t} = \begin{bmatrix}\widehat{\Delta_{ij}^t} & \widehat{\Lambda_{ij}^t} & \widehat{\Gamma_{ij}^t} & \widehat{\Upsilon_{ij}^t}\end{bmatrix}. \tag{11}$$

It should be noted that $\widehat{F^t}$ has the depth size of three, corresponding to the size of feature components. By stacking the mixture feature $\widehat{F}$ of all skeleton frames in the sequence S, a full feature matrix $\widehat{\mathbf{F}}$ is constructed as follows

$$\widehat{\mathbf{F}} = \begin{bmatrix}\widehat{F^1} \\ \vdots \\ \widehat{F^N}\end{bmatrix}, \tag{12}$$

where N refers to as the number of frames in the skeleton sequence S. The information of action performance is herein entirely described by the 3D feature matrix $\widehat{\mathbf{F}}$. Furthermore, it should be noticed that the values of features in the full normalized matrix $\widehat{\mathbf{F}}$ are distributed in the range [0,1].

3.1.3. Action image generation

To explain the human pose by image-based representation, we encode each normalized feature in $\widehat{\mathbf{F}}$ as a color pixel in the RGB space. Indeed, three values of each feature are correspondingly encoded to the values of three color channels of a pixel by a general intensity conversion function as follows

$$\mathbf{I} = \widehat{\mathbf{F}} \times (g_{\max} - g_{\min}) = \begin{bmatrix}\mathbf{g}_{\delta^t}^1 & \mathbf{g}_{\lambda^t}^1 & \mathbf{g}_{\gamma^t}^1 & \mathbf{g}_{\upsilon^t}^1 \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{g}_{\delta^t}^N & \mathbf{g}_{\lambda^t}^N & \mathbf{g}_{\gamma^t}^N & \mathbf{g}_{\upsilon^t}^N\end{bmatrix}, \tag{13}$$

where $\mathbf{I}$ refers to as the output action image which is represented by a 3D matrix of intensity values; $\mathbf{g}_\delta$, $\mathbf{g}_\lambda$, $\mathbf{g}_\gamma$, and $\mathbf{g}_\upsilon$ refer to as the color pixels that are converted from pose distance, transition distance pose orientation, and transition orientation features, respectively; and $[g_{\min}, g_{\max}]$ is the range of grayscale value, usually $g_{\min} = 0$ and $g_{\max} = 255$ for a full-scale conversion to 24-bits color image. The structure of output image $\mathbf{I}$ can be portrayed by three individual color channels as follows

$$\mathbf{I} = \begin{bmatrix}\mathbf{R} & \mathbf{G} & \mathbf{B}\end{bmatrix}_{depth}, \tag{14}$$

where three color channels $\mathbf{R}$, $\mathbf{G}$, and $\mathbf{B}$ are constructed as

$$\mathbf{R} = \begin{bmatrix} g_\delta^{1,z=0} & g_\lambda^{1,z=0} & g_\gamma^{1,\vec{Ox}} & g_\upsilon^{1,\vec{Ox}} \\ \vdots & \vdots & \vdots & \vdots \\ g_\delta^{N,z=0} & g_\lambda^{N,z=0} & g_\gamma^{N,\vec{Ox}} & g_\upsilon^{N,\vec{Ox}} \end{bmatrix},$$

$$\mathbf{G} = \begin{bmatrix} g_\delta^{1,x=0} & g_\lambda^{1,x=0} & g_\gamma^{1,\vec{Oy}} & g_\upsilon^{1,\vec{Oy}} \\ \vdots & \vdots & \vdots & \vdots \\ g_\delta^{N,x=0} & g_\lambda^{N,x=0} & g_\gamma^{N,\vec{Oy}} & g_\upsilon^{N,\vec{Oy}} \end{bmatrix},$$

$$\mathbf{B} = \begin{bmatrix} g_\delta^{1,y=0} & g_\lambda^{1,y=0} & g_\gamma^{1,\vec{Oz}} & g_\upsilon^{1,\vec{Oz}} \\ \vdots & \vdots & \vdots & \vdots \\ g_\delta^{N,y=0} & g_\lambda^{N,y=0} & g_\gamma^{N,\vec{Oz}} & g_\upsilon^{N,\vec{Oz}} \end{bmatrix}. \tag{15}$$

It is realized that the color pixels $\mathbf{g}_\delta$, $\mathbf{g}_\lambda$, $\mathbf{g}_\gamma$, and $\mathbf{g}_\upsilon$ in (13) are formed by

$$\mathbf{g}_\delta = \left[g_\delta^{z=0}, g_\delta^{x=0}, g_\delta^{y=0}\right]^{\mathbb{N}},$$

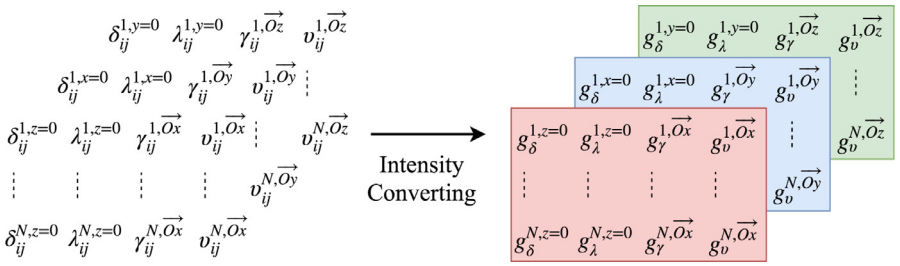

Fig. 2. The illustration of intensity conversion and pixel arrangement for generating color action images.

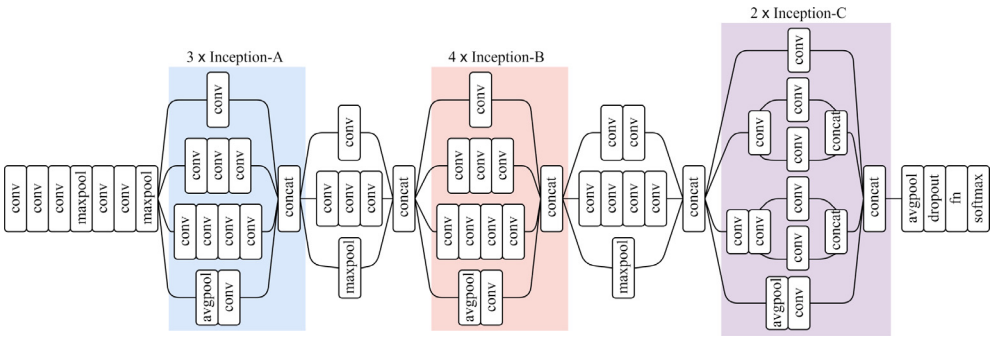

Fig. 3. The overview architecture of Inception-v3 with several principal Inception-A, Inception-B, and Inception-C modules.

$$\begin{aligned}
\mathbf{g}_{\lambda} &= \left[g_{\lambda}^{z=0}, g_{\lambda}^{x=0}, g_{\lambda}^{y=0}\right]^{\mathbb{N}},\\
\mathbf{g}_{\gamma} &= \left[g_{\gamma}^{\overrightarrow{Ox}}, g_{\gamma}^{\overrightarrow{Oy}}, g_{\gamma}^{\overrightarrow{Oz}}\right]^{\mathbb{N}},\\
\mathbf{g}_{\upsilon} &= \left[g_{\upsilon}^{\overrightarrow{Ox}}, g_{\upsilon}^{\overrightarrow{Oy}}, g_{\upsilon}^{\overrightarrow{Oz}}\right]^{\mathbb{N}},
\end{aligned} \tag{16}$$

where $\mathbb{N}$ refers to as the depth dimension of color images. The illustration of converting from feature values to intensity values is presented in Fig. 2 with the arrangement of three color channels. The image size is same with the size of feature matrix $\widehat{\mathbf{F}}$. Corresponding to a skeleton sequence depicting an action appearance, the encoding technique generates an action image, wherein the joint-joint correlations within a frame and across two consecutive frames are comprehensively represented by color pixels.

3.2. Action learning with deep CNNs

Instead of training a deep network from scratch with randomly initialized weights, fine-tuning a pre-trained network is more advantageous (i.e., saving training time with inheriting valuable feature set learned from a huge dataset in advance). Thanks to the power of pre-trained networks and the usage convenience, most of the CNNs-based action recognition approaches either exploit a pre-trained model to learn new patterns or develop a network with weights of convolution layers initialized by a pre-trained model. In this paper, we fine-tune the pre-trained Inception-v3 model, a state-of-the-art deep CNNs, by using the generated action image dataset beforehand. Inception-v3, which is fundamentally developed for the image classification task in computer vision, has achieved a very impressive performance in terms of classification accuracy. Compared with several modern networks (e.g., VGG-16, VGG-19, ResNet-101, and etc.), the computational cost of Inception-v3 is more efficient. Regarding to the network architecture (see Fig. 3), Inception-v3 has 3 Inception-A modules (where a sequence of 3×3 convolutional layers is adopted to reduce the number of parameters), 4 Inception-B modules (where several asymmetric convolutional layers 1×7 and 7×1 are connected to significantly save computational cost), and 2 Inception-C modules (where filter bank outputs are expanded to generate high dimensional sparse representation). By passing the asymmetric filters along the input (known as the output of previous layer) vertically, the network learns the temporal information of skeleton dynamics thoroughly. In order to guarantee the network compatibility, the size of input images must be rescaled to the pre-defined size of corresponding network. Furthermore, the last fully connected layer (a.k.a., the output layer) is modified to fit the number of action classes.

4. Experiments

In this section, we evaluate and compare the proposed method with several state-of-the-art 3D action recognition approaches on such common datasets as UTKinect-Action3D [38], SBU-Kinect Interaction [41], and NTU RGB+D [21] datasets. Furthermore, we analyze the effectiveness of our approach on different parameter configurations.

4.1. Datasets

UTKinect-Action3D: The UTKinect-Action3D dataset, which is collected by using a single stationary Kinect sensor, has 200 sequences of 10 human-object interactive action classes performed by 10 participants. Each action is captured twice for every subject and each skeleton detected in a frame has 20 joints. As a challenge, the coordinates of some body parts are tracked unsuccessful due to occlusion. We follow the standard leave-one-out cross validation configuration, where one sequence is used for testing and reaming samples are for training.

SBU-Kinect Interaction: This dataset describes the human-human interactive actions, wherein one person acts and another does the reaction. The entire dataset has 282 skeleton sequences corresponding to 8 interaction classes performed by 7 subjects. Each person is detected by a 15-joints skeleton. Some interactions depicting very similar postures at the beginning and ending of appearance is challenging to recognize precisely. A standard benchmark configuration is provided with 5-fold cross validation.

NTU RGB+D: This dataset is recorded by Kinect sensor v2 with more upgrades of hardware and software to estimate pose more accurately. For details, there are totally 56,880 skeleton sequences covering 60 action classes, including single person actions, human-object interactions, and human-human interactions. Each skeleton is depicted by 25 joints. As our knowledge, NTU RGB+D is the most challenging dataset currently due to its large intra-class and view-point variations. Due to the large amount of samples, NTU RGB+D is pretty suitable to deep learning-based action recognition method. The dataset provides two evaluation protocols that are cross-subject (i.e., subjects with IDs 3, 6, 7, 10, 11, 12, 20, 21, 22, 23, 24, 26, 29, 30, 32, 33, 36, 37, 39, 40 are for test and remaining subjects are for training) and cross-view (i.e., samples with camera IDs 1 are for testing while remaining samples with camera IDs 2 and 3 are for training).

Compared with NTU RGB+D, both the UTKinect-Action3D and SBU-Kinect Interaction datasets are significantly small with 200 and 282 action images generated by our PoT2I encoding technique, respectively, that potentially leads to a risk of network overfitting. One of the popular solution to reduce overfitting influence on CNNs model is data augmentation by increasing the amount of training action images. In this research, for each original action image of UTKinect-Action3D and SBU-Kinect Interaction, we generate multiple mirror samples by adding some skeleton frames randomly selected from a corresponding action sequence. By this data augmentation mechanism, the primary action flow is almost preserved.

4.2. Experimental setup

Regarding the hyper-parameter setting for end-to-end fine-tuning the Inception-v3, we utilize the stochastic gradient descent with momentum (SGDM) optimizer, where the mini-batch size is set to 128. The network is fine-tuned in 30 epochs with the learning rate initialized at 0.01 and reduced 90% after each 10 epochs. All experiments are implemented on a system using two NVIDIA GeForce GTX 1080Ti cards. There are three experiments briefly described as follows:

- In the first experiment, we evaluates the proposed method on various well-known datasets and further compares with other existing action recognition approaches in terms of accuracy.
- The second experiment analyzes the performance sensitivity of our method under different feature types and also various pre-trained CNNs models.
- The last experiment aims to measure the method complexity based on measuring the processing time, including two procedures of feature to image encoding and action recognition.

4.3. Experimental results

4.3.1. Method comparison

The proposed 3D action recognition method, denoted PoT2I + Inception-v3, is evaluated and compared with state-of-the-art methods on several common benchmark datasets for analysis and discussion.

UTKinect-Action3D Dataset: The proposed PoT2I + Inception-v3 achieves the recognition accuracy of 98.5%, the runner-up score (less than GCA-LSTM (stepwise) [18], Multi-Stream CNN [19], and RotClips + MTCNN [12] by 0.5%), as reported in Table 1. The classification confusion matrix is presented in Fig. 4, where some samples of *throw* and *push* actions are misclassified to each other. As the effort of performance improvement, the strategy of dual-stream networks for jointly learning joint-level and body part-level information is recommended in GCA-LSTM (stepwise) [18]. Compared with Multi-Stream CNN (wherein a multiple-CNN architecture is developed for learning deep features of many encoded color images representing for an action sample) and RotClips + MTCNN (wherein encoding clips transformed from a skeleton sequence are fed into a multiple-convolutional-stacks CNNs model), PoT2I + Inception-v3 (with a single CNN architecture) is much less complicated. Moreover, the final action is given by incorporating class scores of multiple CNNs by an averaged-fusion

Table 1
Accuracy comparison on UTKinect-Action3D dataset.

Method	Accuracy (%)
MLSTM + Weight Fusion [43]	96.0
Ensemble TS-LSTM v2 [14]	97.0
ST-LSTM [17]	97.0
PAM + Pose-Transition Feture [8]	97.0
Lie Group [32]	97.1
PoT2I + Inception-v3	**98.5**
GCA-LSTM (stepwise) [18]	99.0
Multi-Stream CNN [19]	99.0
RotClips + MTCNN [12]	99.0

Table 2
Accuracy comparison on SBU-Kinect interaction dataset.

Method	Accuracy (%)
Multiple Instance Learning [41]	80.3
HBRNN-L [4]	80.4
STA-LSTM [25]	91.5
ST-LSTM [17]	93.3
SkeletonNet [11]	93.5
RotClips + MTCNN [12]	94.2
Two-stream RNN [34]	94.8
GCA-LSTM (stepwise) [18]	94.9
PoT2I + Inception-v3	**95.9**
VA-LSTM [42]	97.2
MLSTM + Weight Fusion [43]	99.3

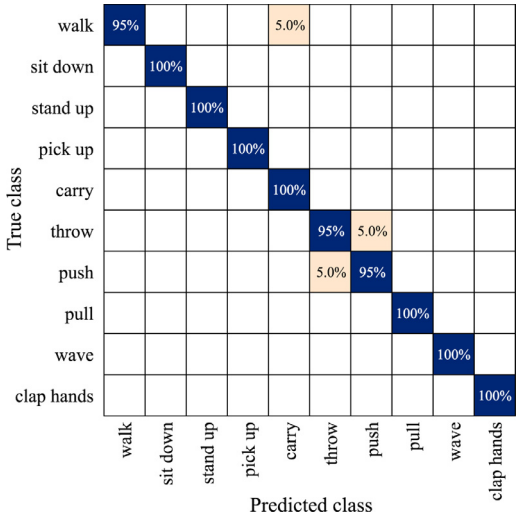

Fig. 4. The confusion matrix of the UTKinect-Action3D dataset.

(RotClips + MTCNN) or a weighted-fusion (Multi-Stream CNN) which can be integrated inside the network or designed as a post-processing component. The accuracy of PoT2I + Inception-v3 is better than the results of numerous previous state-of-the-art methods, including handcrafted feature-based (e.g., PAM + Pose-Transition Feture [8] and Lie Group [32]) and LSTM-based (e.g., MLSTM + Weight Fusion [43], Ensemble TS-LSTM v2 [14], and ST-LSTM [17]).

SBU-Kinect Interaction Dataset: For the two-person interactive action challenge presented in this dataset, the proposed PoT2I + Inception-v3 reaches the recognition accuracy of 95.9% as reported in Table 2, besides the detailed confusion matrix shown in Fig. 5. Compared with GCA-LSTM (stepwise) [18] and RotClips + MTCNN [12], PoT2I + Inception-v3 outperforms by 1.0% and 1.7%, respectively. In these methods, because the raw skeleton coordinate data is either fed into a network or basically encoded to image-based representation, the spatiotemporal correlations between two persons cannot be captured explicitly. Meanwhile, PoT2I encodes an action image by exploiting relative pose and transition features for portraying spatial and temporal human-human interactions. Compared with VA-LSTM [42] and MLSTM + Weight Fusion [43] results, the accuracy of PoT2I + Inception-v3 is worse than 1.3% and 3.4%, respectively. By rotating original coordinates, VA-LSTM

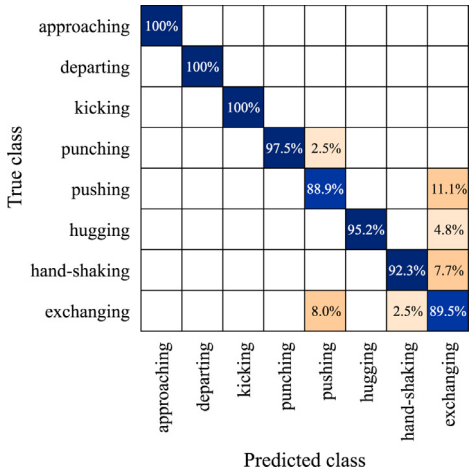

Fig. 5. The confusion matrix of the SBU-Kinect Interaction dataset.

Table 3
Accuracy comparison on NTU RGB+D dataset.

Method	Accuracy (%)	
	C-S	C-V
Skeleton Quads [5]	38.60	41.40
Lie Group [32]	50.08	52.76
Deep RNN [21]	56.29	64.09
HBRNN-L [4]	59.07	63.97
Deep LSTM [21]	60.69	67.29
P-LSTM [21]	62.93	70.27
ST-LSTM [17]	69.20	77.70
Two-stream RNN [34]	71.30	79.50
End-to-End RNN [33]	72.40	84.60
PAM + Pose-Transition Feature [8]	72.77	77.42
STA-LSTM [25]	73.40	81.20
Res-TCN [13]	74.30	83.10
Ensemble TS-LSTM v2 [14]	74.60	82.56
Image Generation + VGG-19 [15]	75.20	82.10
SkeletonNet [11]	75.94	81.16
GCA-LSTM (stepwise) [18]	76.10	84.00
JTM + ConvNet [37]	76.32	81.08
MLSTM + Weight Fusion [43]	76.43	87.69
Deep LSTMZ [30]	78.14	85.73
SPMF Inception-ResNet-222 [20]	78.89	86.15
VA-LSTM [42]	79.40	87.60
Beyond Joints [35]	79.50	87.60
Deep LSTMZ + LSTM-AE [29]	80.63	88.56
Multi-Stream LSTM [36]	80.90	89.60
RotClips + MTCNN [12]	81.09	87.37
PoT2I + Inception-v3	**83.85**	**90.33**

can precisely recognize more complicated interactive actions for different viewpoints (for example, *hugging* and *approaching*, which share several posture at beginning, now can be differentiated in other transformed views). With MLSTM + Weight Fusion, action recognition is formulated to the decision-level fusion of multiple LSTM-based classifiers, where each one is trained on separated different geometric features (e.g., joint-plane distance, line-plane angle, plane-plane angle, and etc.), and the final label is then given by aggregating all model outputs via a weighted average fusing rule.

NTU RGB+D: From the numerical results provided in Table 3, the proposed PoT2I + Inception-v3 outperforms state-of-the-art approaches with the highest recognition accuracy, particularly 83.85% for the cross-subject protocol and 90.33% for the cross-view protocol. Accordingly, two confusion matrices of cross-subject and cross-view are depicted in Figs. 6 and 7, respectively. There exists a big gap of accuracy between deep learning-based approaches and handcrafted feature-based methods (e.g., Skeleton Quads [5] and Lie Group [32]). It can be seen that the combination of traditional classifiers and handcrafted feature descriptors is not effective with such a large and challenging dataset as NTU RGB+D. Therefore, most of the 3D action recognition methods reported in Table 3 are developed on RNNs, LSTM networks, and CNNs to significantly improve performance thanks to the benefits of deep learning in several classification tasks. Some approaches either expand

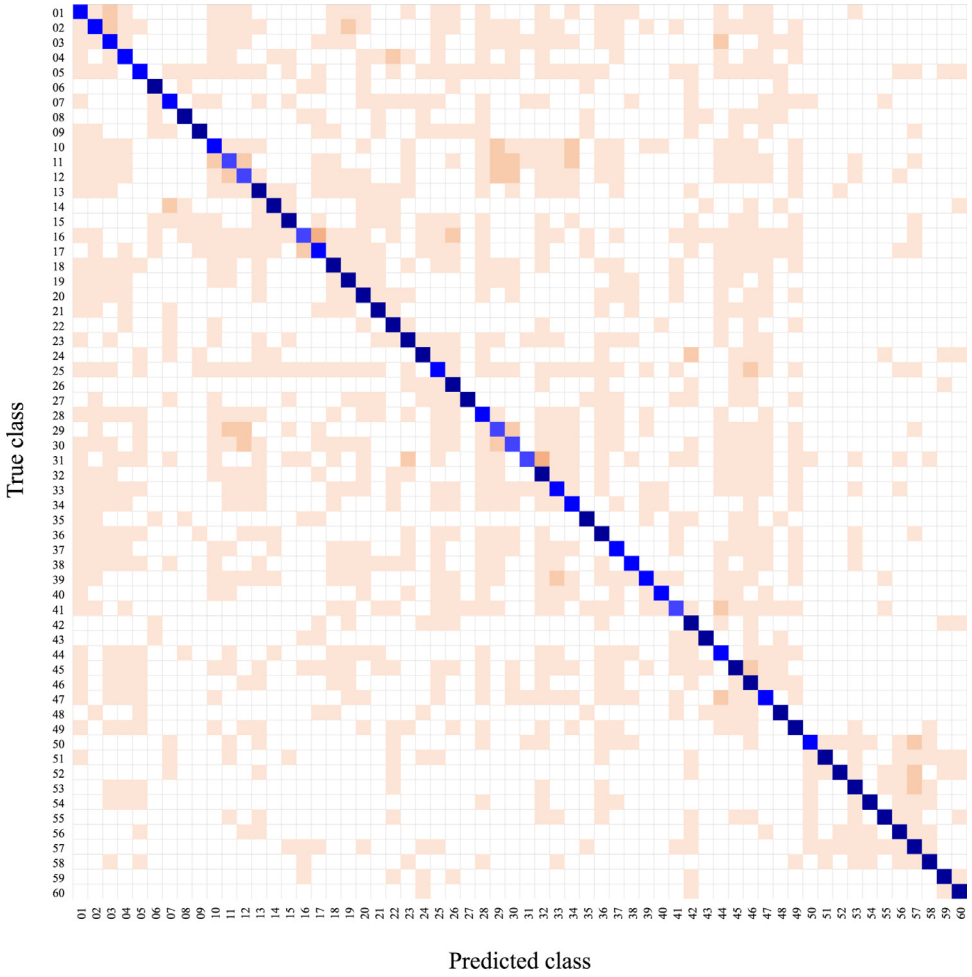

Fig. 6. The confusion matrix of the NTU RGB+D dataset with the cross-subject evaluation protocol. The classes from 01 to 60 standing for drink water, eat meal, brushing teeth, brushing hair, drop, pickup, throw, sitting down, standing up (from sitting), clapping, reading, writting, tear up paper, wear jacket, wear a shoe, take off a shoe, wear on glasses, take off glasses, put on a hat, take of a hat, cheer up, hand waving, kicking something, put something inside pocket/take something from pocket, hopping, jump up, make a phone call, playing with phone, typing on a keyboard, pointing to something, taking a selfie, check time, rub two hands together, nob head, shake head, wipe head, salute, put the palms together, cross hands in front, sneeze/cough, staggering, falling, touch head, touch chest, touch back, touch neck, nausea or vomiting condition, use a fan, punching, kicking, pushing, pat on back,pointing to someone, hugging, giving something, touch other person's pocket, handshaking, walking towards each other, walking apart each other.

a standard RNN with more deep layers (e.g., Deep RNN [21] and End-to-End RNN [33]) or develop a novel RNN architecture (e.g., HBRNN-L [4] and Two-stream RNN [34]). Compared with Lie Group [32], RNNs-based approaches conduct the average improvement of 14.68% and 31.64% for the cross-subject and cross-view evaluation settings, respectively. Besides that, LSTM network is widely used for modeling action pattern in many recent studies. Most of them make the contribution on enlarging the deep size of networks (e.g., Deep LSTM [21]), upgrading the activation function (e.g., ST-LSTM [17]), building a novel hierarchical network (e.g., Deep LSTMZ + LSTM-AE [30], P-LSTM [21], ST-LSTM [17], VA-LSTM [42], and STA-LSTM [25]), and developing a multi-stream network architecture with decision-level fusion (e.g., Beyond Joints [35], Ensemble TS-LSTM v2 [14,19], MLSTM + Weight Fusion [43], and Multi-Stream LSTM [36]). Generally, LSTM-based action recognition approaches yield remarkable performance, particularly, the average accuracy rate of 73.73% and 81.83% for the cross-subject and cross-view configurations, respectively. Lately, based on outstanding performance in general image classification tasks, CNNs have been studied for skeleton-based action recognition. Some approaches basically feed handcrafted or high-level features (e.g., Res-TCN [13]) to a deep network, but the recognition improvement is minor. Since CNNs are specialized for image-based format, several methods transform raw coordinate data (e.g., Image Generation + VGG-19 [15], Multi-Stream CNN [19], and RotClips + MTCNN [12]) and also low-level features (e.g., cosine distance in SkeletonNet [11], joint motion direction in JTM + ConvNet [37], and pose-motion feature in SPMF Inception-ResNet-222 [20]) to image-based representation. Convolutional neural networks are taken advantage in different ways for action learning, for example, either developing a novel network architecture (hierarchical or multi-stream design) with randomly initialized weights or end-to-end fine-tuning a pre-trained model. From the results in Table 3, the proposed PoT2I + Inception-v3 outperforms all approaches which exploit

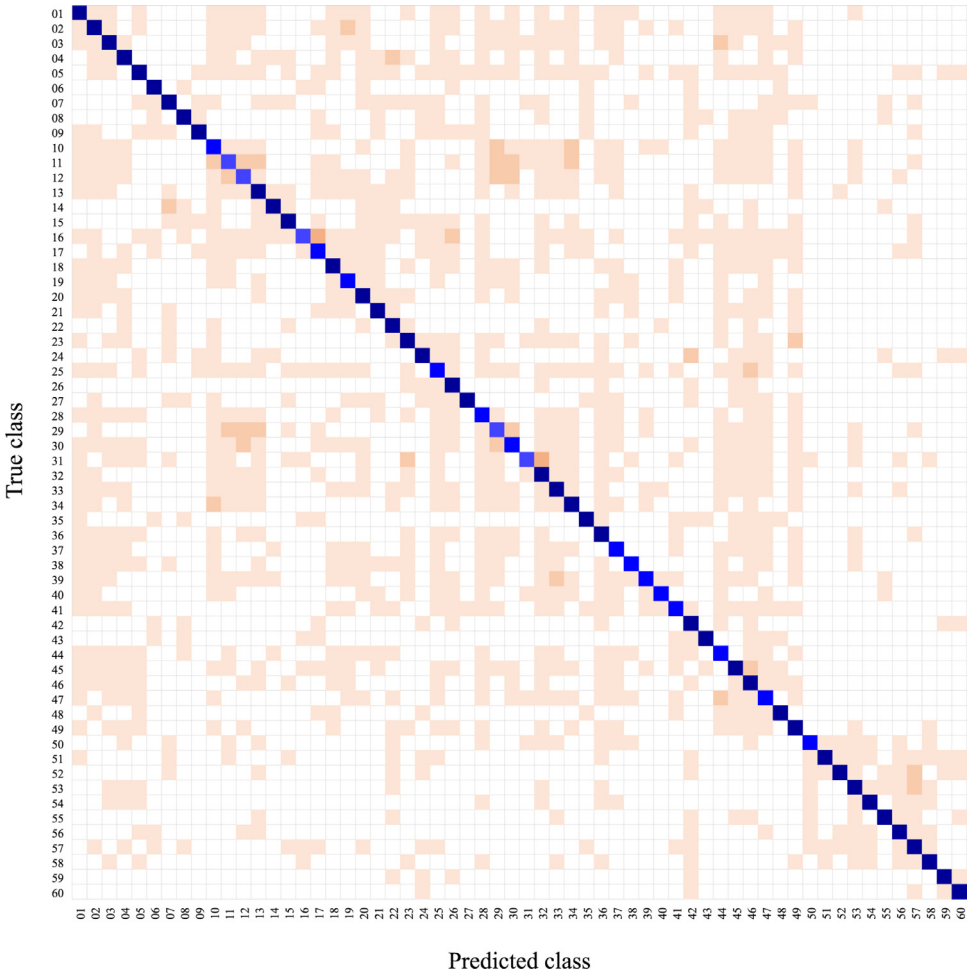

Fig. 7. The confusion matrix of the NTU RGB+D dataset with the cross-view evaluation protocol.

Table 4
Recognition accuracy with different feature categories on NTU RGB+D (BatchSize = 128).

Feature	Accuracy (%)	
	C-S	C-V
Pose-transition distance	82.87	88.61
Pose-transition orientation	80.64	85.50
Pose distance-orientation	81.67	87.79
Transition distance-orientation	82.30	88.15

skeleton-to-image transformation and CNNs-based learning. Compared with RotClips + MTCNN [12], besides recognizing actions more precisely, PoT2I + Inception-v3 presents a much more simple network (i.e., a multi-task CNNs architecture constructed with several streams of convolutional stacks is certainly more complicated than a single Inception-v3 network). Obviously, by an efficient encoding technique introduced in this work, PoT2I + Inception-v3 yields remarkable performance on several benchmark datasets and further outperforms state-of-the-art methods in terms of recognition accuracy.

4.3.2. Performance sensitivity analysis

In this experiment, we investigate the influence of feature type, used for data transformation, on the overall recognition accuracy. Concretely, four feature categories (e.g., pose-transition distance $\left\{\delta_{ij}^t, \lambda_{ij}^t\right\}$, pose-transition orientation $\left\{\gamma_{ij}^t, \upsilon_{ij}^t\right\}$, pose distance-orientation $\left\{\delta_{ij}^t, \gamma_{ij}^t\right\}$, and transition distance-orientation $\left\{\lambda_{ij}^t, \upsilon_{ij}^t\right\}$) are organized to fine-tune using the pre-trained Inception-v3 model. The numerical results are shown in Table 4. It can be seen that distance feature is more efficient than orientation feature by 2.67% approximately due to the variety of action performance in different viewpoints (i.e., some

Table 5
Recognition accuracy with different pre-trained CNNs Models On NTU RGB+D (Mini-Batch Size = 64).

Network	Accuracy (%)	
	C-S	C-V
VGG-16	79.41	85.42
VGG-19	79.52	86.08
GoogleNet	79.56	84.98
Inception-v3	83.05	88.80
ResNet101	82.36	88.10

Table 6
Method comparison in terms of processing time.

Method	Programming	Computing	Time (ms)
Joint Encoding + ConvNet [4]	n/a	Titan GK110 GPU	~2.3*
SOS + ConvNet [6]	Matlab + Python	Titan X GPU	~40
SPMF Inception-ResNet-222 [20]	Python	GTX Ti GPU	~128
PoT2I + Inception-v3	Matlab	2 × GTX 1080Ti GPU	~38

* : only the classification time without data encoding.

actions share a same relative joint-joint orientation if they are observed from different points of view). Under other conditions, the transition feature is slightly better than the pose feature by 0.50%.

Additionally, we benchmark the PoT2I encoding technique with different pre-trained network models, including VGG-16, VGG-19, GoogleNet, and ResNet101 besides Inception-v3. Due to the GPUs memory limitation when fine-tuning on ResNet101, we turn the mini-batch size to 64 for fair comparison of all networks. The recognition results on the NTU RGB+D dataset are summarized in Table 5, where the accuracy of cross-subject and cross-view protocol are reported. By using more convolutional layers, VGG-19 is more precise than VGG-16 but insignificant. Compared with VGG nets, GoogleNet drastically reduces the total number of parameters with a very comparable accuracy thanks to the inception modules. ResNet101 can reach remarkable accuracy with 101 convolutional layers (much more deeper than VGG nets) while still maintaining the complexity lower than VGG-16. By several improvements relating to convolution factorization, classifier regularization, and grid size reduction, Inception-v3 achieves an impressive performance in terms of recognition accuracy and complexity. Based on the results in Table 5, it is recognized that PoT2I can properly work with different networks to conduct a very competitive accuracy compared with other deep learning-based approaches.

4.3.3. Complexity measurement

For complexity analysis, we measure the processing time of PoT2I + Inception-v3 and further compare with other deep learning-based approaches. With a desktop computer equipped with a 3.7-Ghz AMD Ryzen 7 2700x CPU, 16GB RAM, and two NVIDIA GeForce GTX 1080Ti GPU, the systems takes approximately 35ms for feature calculation, intensity conversion, and image creation (implemented on CPU) plus 3ms for action classification (executed on GPUs). In addition, the numerical results of various existing CNN-based methods are summarized in Table 6. It is realized that conducting a comparison of deep learning-based methods, in terms of processing speed and the memory consumption, cannot be done fairly due to the diversity in use of programming languages (e.g., C++, Matlab, Python) and computing platforms (e.g., single machine vs. distributed system). From Table 6, PoT2I + Inception-v3 is faster than SPMF Inception-ResNet-222 thanks to the less operations of Inception-v3 compared with ResNet-222. Meanwhile, the gap between SOS + ConvNet and our method is insignificant despite the fact that AlexNet is more simple than Inception-v3.

5. Conclusion

In this paper, we have proposed an efficient encoding technique to represent the information of pose and motion under a color image for deep CNNs-based action recognition. Different from existing work encoding raw skeleton coordinates to image form, we exploit higher discriminative geometric features for data transformation. Particularly, the pose and transition features including joint-joint distance and joint-joint orientation of each skeleton frame are extracted for being encoded as color pixels. By gathering all frames of a sequence, an action image is generated, in which the joint correlations and posture dynamics are described entirely. The encoded images are then fine-tuned for a pre-trained CNNs model to learn spatial and temporal high-level features at multi-scale action representation. Our method achieves the remarkable performance on the NTU RGB+D dataset with accuracy rate of 83.85% and 90.33% for the cross-subject and cross-view protocol, respectively. In future, we attempt to consider a multiple stream CNNs architecture to specifically learn many kinds of geometric feature for performance improvement.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This research was financially supported by the National Research Foundation of Korea (NRF) through Creativity Challenge Research-based Project (2019R1I1A1A01063781), and in part by Priority Research Centers Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2018R1A6A1A03024003).

References

[1] B.B. Amor, J. Su, A. Srivastava, Action recognition using rate-invariant analysis of skeletal shape trajectories, IEEE Trans. Pattern Anal. Mach. Intell. 38 (1) (2016) 1–13.
[2] X. Cai, W. Zhou, L. Wu, J. Luo, H. Li, Effective active skeleton representation for low latency human action recognition, IEEE Trans. Multimedia 18 (2) (2016) 141–154.
[3] W. Chen, G. Guo, Triviews: a general framework to use 3d depth data effectively for action recognition, J. Vis. Commun. Image Represent. 26 (2015) 182–191.
[4] Y. Du, W. Wang, L. Wang, Hierarchical recurrent neural network for skeleton based action recognition, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1110–1118.
[5] G. Evangelidis, G. Singh, R. Horaud, Skeletal quads: Human action recognition using joint quadruples, in: 2014 22nd International Conference on Pattern Recognition, 2014, pp. 4513–4518.
[6] Y. Hou, Z. Li, P. Wang, W. Li, Skeleton optical spectra-based action recognition using convolutional neural networks, IEEE Trans. Circuits Syst. Video Technol. 28 (3) (2018) 807–811, doi:10.1109/TCSVT.2016.2628339.
[7] T. Huynh-The, C. Hua, D. Kim, Learning action images using deep convolutional neural networks for 3d action recognition, in: 2019 IEEE Sensors Applications Symposium (SAS), 2019, pp. 1–6, doi:10.1109/SAS.2019.8705977.
[8] T. Huynh-The, C.-H. Hua, N.A. Tu, T. Hur, J. Bang, D. Kim, M.B. Amin, B.H. Kang, H. Seung, S.-Y. Shin, E.-S. Kim, S. Lee, Hierarchical topic modeling with pose-transition feature for action recognition using 3d skeleton data, Inf. Sci. 444 (2018) 20–35.
[9] T. Huynh-The, H. Hua-Cam, D. Kim, Encoding pose features to images with data augmentation for 3d action recognition, IEEE Trans. Ind. Inf. (2019), doi:10.1109/TII.2019.2910876. 1–1.
[10] T. Huynh-The, B.-V. Le, S. Lee, Y. Yoon, Interactive activity recognition using pose-based spatiotemporal relation features and four-level pachinko allocation model, Inf. Sci. 369 (2016) 317–333.
[11] Q. Ke, S. An, M. Bennamoun, F. Sohel, F. Boussaid, Skeletonnet: mining deep part features for 3-d action recognition, IEEE Signal Process. Lett. 24 (6) (2017) 731–735.
[12] Q. Ke, M. Bennamoun, S. An, F. Sohel, F. Boussaid, Learning clip representations for skeleton-based 3d action recognition, IEEE Trans. Image Process. 27 (6) (2018) 2842–2855.
[13] T.S. Kim, A. Reiter, Interpretable 3d human action analysis with temporal convolutional networks, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2017, pp. 1623–1631.
[14] I. Lee, D. Kim, S. Kang, S. Lee, Ensemble deep learning for skeleton-based action recognition using temporal sliding lstm networks, in: 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 1012–1020.
[15] C. Li, S. Sun, X. Min, W. Lin, B. Nie, X. Zhang, End-to-end learning of deep convolutional neural network for 3d human action recognition, in: 2017 IEEE International Conference on Multimedia Expo Workshops (ICMEW), 2017, pp. 609–612.
[16] M. Li, H. Leung, Graph-based approach for 3d human skeletal action recognition, Pattern Recognit. Lett. 87 (2017) 195–202. Advances in Graph-based Pattern Recognition.
[17] J. Liu, A. Shahroudy, D. Xu, G. Wang, Spatio-temporal lstm with trust gates for 3d human action recognition, in: B. Leibe, J. Matas, N. Sebe, M. Welling (Eds.), Computer Vision–ECCV 2016, 2016.
[18] J. Liu, G. Wang, L. Duan, K. Abdiyeva, A.C. Kot, Skeleton-based human action recognition with global context-aware attention lstm networks, IEEE Trans. Image Process. 27 (4) (2018) 1586–1599.
[19] M. Liu, C. Chen, H. Liu, 3d action recognition using data visualization and convolutional neural networks, in: 2017 IEEE International Conference on Multimedia and Expo (ICME), 2017, pp. 925–930.
[20] H. Pham, L. Khoudour, A. Crouzil, P. Zegers, S.A. Velastin, Skeletal movement to color map: A novel representation for 3d action recognition with inception residual networks, in: 2018 25th IEEE International Conference on Image Processing (ICIP), 2018, pp. 3483–3487.
[21] A. Shahroudy, J. Liu, T. Ng, G. Wang, Ntu rgb+d: A large scale dataset for 3d human activity analysis, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 1010–1019.
[22] A. Shahroudy, T. Ng, Q. Yang, G. Wang, Multimodal multipart learning for action recognition in depth videos, IEEE Trans. Pattern Anal. Mach. Intell. 38 (10) (2016) 2123–2129.
[23] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, CoRR abs/1409.1556 (2014).
[24] R. Slama, H. Wannous, M. Daoudi, A. Srivastava, Accurate 3d action recognition using learning on the grassmann manifold, Pattern Recognit. 48 (2) (2015) 556–567.
[25] S. Song, C. Lan, J. Xing, W. Zeng, J. Liu, Spatio-temporal attention-based lstm networks for 3d action recognition and detection, IEEE Trans. Image Process. 27 (7) (2018) 3459–3471.
[26] V. Sze, Y. Chen, T. Yang, J.S. Emer, Efficient processing of deep neural networks: a tutorial and survey, Proc. IEEE 105 (12) (2017) 2295–2329.
[27] C. Szegedy, S. Ioffe, V. Vanhoucke, A. Alemi, Inception-v4, inception-resnet and the impact of residual connections on learning, in: in Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17), 2017.
[28] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2818–2826, doi:10.1109/CVPR.2016.308.
[29] J. Tu, H. Liu, F. Meng, M. Liu, R. Ding, Spatial-temporal data augmentation based on lstm autoencoder network for skeleton-based human action recognition, in: 2018 25th IEEE International Conference on Image Processing (ICIP), 2018, pp. 3478–3482.
[30] N.A. Tu, T. Huynh-The, K.U. Khan, Y. Lee, Ml-hdp: a hierarchical bayesian nonparametric model for recognizing human actions in video, IEEE Trans. Circuits Syst. Video Technol. 29 (3) (2019) 800–814, doi:10.1109/TCSVT.2018.2816960.
[31] V. Veeriah, N. Zhuang, G. Qi, Differential recurrent neural networks for action recognition, in: 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 4041–4049.
[32] R. Vemulapalli, F. Arrate, R. Chellappa, Human action recognition by representing 3d skeletons as points in a lie group, in: Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, in: CVPR '14, 2014, pp. 588–595.
[33] H. Wang, L. Wang, Learning robust representations using recurrent neural networks for skeleton based action classification and detection, in: 2017 IEEE International Conference on Multimedia Expo Workshops (ICMEW), 2017, pp. 591–596.

[34] H. Wang, L. Wang, Modeling temporal dynamics and spatial configurations of actions using two-stream recurrent neural networks, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 3633–3642.
[35] H. Wang, L. Wang, Beyond joints: learning representations from primitive geometries for skeleton-based action recognition and detection, IEEE Trans. Image Process. 27 (9) (2018) 4382–4394.
[36] L. Wang, X. Zhao, Y. Liu, Skeleton feature fusion based on multi-stream lstm for action recognition, IEEE Access 6 (2018) 50788–50800.
[37] P. Wang, W. Li, C. Li, Y. Hou, Action recognition based on joint trajectory maps with convolutional neural networks, Knowl. Based Syst. 158 (2018) 43–53.
[38] L. Xia, C. Chen, J.K. Aggarwal, View invariant human action recognition using histograms of 3d joints, in: 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2012, pp. 20–27.
[39] Y. Xiao, J. Chen, Y. Wang, Z. Cao, J.T. Zhou, X. Bai, Action recognition for depth video using multi-view dynamic images, Inf. Sci. 480 (2019) 287–304.
[40] X. Yang, Y. Tian, Effective 3d action recognition using eigenjoints, J. Vis. Commun. Image Represent. 25 (1) (2014) 2–11. Visual Understanding and Applications with RGB-D Cameras.
[41] K. Yun, J. Honorio, D. Chattopadhyay, T.L. Berg, D. Samaras, Two-person interaction detection using body-pose features and multiple instance learning, in: 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2012, pp. 28–35.
[42] P. Zhang, C. Lan, J. Xing, W. Zeng, J. Xue, N. Zheng, View adaptive recurrent neural networks for high performance human action recognition from skeleton data, in: 2017 IEEE International Conference on Computer Vision (ICCV), 2017.
[43] S. Zhang, Y. Yang, J. Xiao, X. Liu, Y. Yang, D. Xie, Y. Zhuang, Fusing geometric features for skeleton-based action recognition using multilayer lstm networks, IEEE Trans. Multimedia 20 (9) (2018) 2330–2343.
[44] X. Zhao, X. Li, C. Pang, Q.Z. Sheng, S. Wang, M. Ye, Structured streaming skeleton–a new feature for online human gesture recognition, ACM Trans. Multimedia Comput. Commun. Appl. 11 (1s) (2014) 22:1–22:18.