A Geometric Approach for Cross-View Human Action Recognition using Deep Learning

Antonios Papadakis*, Eirini Mathe†‡, Evaggelos Spyrou† and Phivos Mylonas‡
*Department of Informatics and Telecommunications, University of Athens
Email: {sdi1400141@di.uoa.gr}
†Institute of Informatics and Telecommunications, National Center for Scientific Research - "Demokritos," Athens, Greece
Email: {emathe,espyrou}@iit.demokritos.gr
‡Department of Informatics, Ionian University, Corfu, Greece
Email: fmylonas@ionio.gr

***Abstract*—In this paper we present an approach for the recognition of human actions which is based on a deep Convolutional Neural Network architecture. More specifically, 3D skeletal joint information is used to create 2D (image) representations. To compensate for potential viewpoint changes, these images are pre-processed using geometric transformations. Then, they are transformed to the spectral domain using well-known transforms. We focus on actions that are close to activities of daily living (ADLs), yet we evaluate our approach using a large-scale action dataset. We cover single-view, cross-view and cross subject cases and thoroughly discuss experimental results and the potential of our approach.**

***Index Terms*—Human Activity Recognition, Convolutional Neural Networks**

I. Introduction

Understanding human activity from video has become one of the most challenging computer vision related tasks, during the last few years and a broad field of applications such as surveillance, assisted living, human-machine interaction, affective computing, etc. have significantly benefited. Open challenges include the representation, the analysis and the recognition of actions. Corresponding tasks are divided into gesture, action, interaction and group activity recognition [21]. They are differentiated based on their duration, body parts involved, number of persons and/or objects interacting. Gestures are instant, involving at most a couple of body parts, while actions require a significant amount of time and may involve more body parts. Interactions are defined between an individual and an object, or between two individuals. Group activities are combinations of the above.

Early research efforts were based on the extraction of hand-crafted features from raw visual data and training of traditional machine learning models [17]. Their main drawback is that their performance may significantly drop in large-scale datasets, while they are not robust to viewpoint changes, a common situation in real-life applications. These limitations may be surpassed using deep neural network architectures and exploitation of complementary modalities such as depth and skeletal data. With the availability of modern graphics processing units (GPUs), fast training of deep architectures [9] has been enabled, allowing their application in real-life problems. Note that in such approaches a feature extraction step is obsolete; instead features are "learnt" within the network. Moreover, it has been demonstrated that their accuracy may significantly increase with larger datasets.

Also, due to recent development of low-cost RGB cameras that also extract depth information, such as the Microsoft Kinect[1] or the Asus Xtion,[2] human motion analysis efforts have incorporated this extra modality. This, is due to a) the insensitivity of depth to illumination changes; and b) offering of an enhanced 3D structural information regarding a scene. Moreover, the combination of RGB and depth data has allowed for the extraction and tracking of 3D position of human joints [19]. Recently, datasets comprised of large numbers of training video and depth sequences collected by such sensors have become available [11], [18], allowing for training and evaluation of deep architectures.

In this work, we present a human action recognition approach, which is applied on segmented video data. In this case, each video segment contains only the action to be recognized. This means that any frame before/after the action, i.e., which does not depict a part of the action, has been removed. Note that recent research efforts in this case are typically based either on deep Convolutional Neural Networks (CNNs) or on deep Recurrent Neural Networks (RNNs) [21]. We adopt a deep CNN model which is trained on 3D skeletal data. Therefore, an intermediate representation of skeletal data sequences is used, capturing both spatial and temporal information regarding the motion of joints which is reflected to color and texture properties of the representation. Of course, a hand-crafted feature extraction step is not necessary.

We build on previous works [13], [14] which propose the use of visual representations of human actions from skeletal data, based on well-known 2D image transformations. The novelty of this work lies in the application of a geometric transform for augmentation of the available dataset and for compensation of viewpoint changes. We evaluate the proposed approach using a challenging large-scale dataset and present

[1] https://developer.microsoft.com/en-us/windows/kinect

[2] https://www.asus.com/gr/3D-Sensor/Xtion_PRO/

results for single-view, cross-subject and cross-view cases.

The rest of this paper is organized as follows: Section II presents related research efforts in the field of human action recognition, that are also based on 3D skeletal motion and make use of deep learning architectures. Section III presents the proposed approach for human motion recognition and the proposed geometric correction step for cross-view motion recognition. Experimental results are presented and discussed in Section IV, while conclusions are drawn and plans of future work are presented in Section V.

II. Related Work

During the last few years, a plethora of human action recognition approaches have been proposed. Herein, we attempt to present state-of-the-art approaches that similarly to this work, attempt to solve the problem by proposing some visual representation of joints when performing actions, which is then used to train CNNs.

Du et al. [5] proposed a pseudo-colored image representation. The set of joints was split into five subsets, i.e., arms, legs, trunk. They corresponded spatial coordinates to color components, i.e., x, y, z to R, G, B, accordingly. In an effort to preserve the temporal information, they chronologically arranged spatial representations. In the work of Wang et al. [21], a representation called "joint trajectory maps," is proposed, so that motion magnitude changes are reflected to texture changes, by appropriately setting saturation and brightness values. Moreover, in the work of Hou et al., [6] a representation called "joint skeleton spectra" is proposed. Therein, temporal variation of skeletal motion are reflected to changes of hue values within the representation. Another representation called "joint distance maps" has been proposed by Li et al. [10]. They encoded the pair-wise joint distances as they vary when performing actions, using color components. Ke et al. [8] extracted translation, rotation and scale invariant features by subsets of joints, as well as cosine distances and normalized magnitudes from vector representations generated from pairwise relative positions between joints, which were then concatenated to form a 2D image representation.

However, only few research efforts have addressed the problem of viewpoint invariance. A notable example is the work of Liu et al. [12], who applied transformations to skeletal sequences. A joint was represented by its 3D space coordinates, while time and joint label were also added to create a 5D representation. Upon projection to a 2D image using two of the aforementioned dimensions, the remaining three were used as R, G, B values to form pseudo-colored images. Our approach has been partially inspired by the one of Zhang et al. [22], who proposed a view adaptive RNN, which aimed to transform skeletons of several views to more consistent viewpoints.

III. Methodology

A. 3D Skeletal Information

As we have already mentioned in Section I, the proposed approach is based on 3D skeletal motion information. More specifically, it uses as input 3D trajectories of a set of human skeletal joints that are automatically extracted from RGB and depth data. More specifically, human motion during the performance of an action is captured using the Microsoft Kinect v2. The latter is an RGB and depth camera and is accompanied by a powerful SDK that allows for real-time extraction and tracking of skeletal joint positions. In other words, x, y and z coordinates are provided for a set of 25 joints. As it may be seen in Fig. 1, a human skeleton corresponds to a graph; nodes correspond to joints (i.e., body parts such as arms, legs, head etc.), while edges connect joints following the body structure. "SPINEBASE" is considered as the root of the graph, thus parent-child relationship among the several joints is implied. For example, "SPINESHOULDER" is the parent of "SHOULDERLEFT," while "SHOULDERLEFT" is the parent of "ELBOWLEFT" and so on.

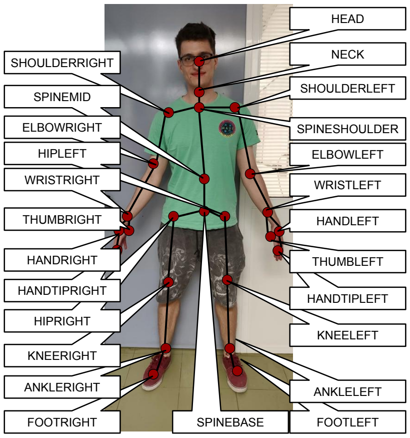

Figure 1. Extracted human skeleton 3D joints using the Kinect SDK.

B. Signal Images

The first step of the proposed approach consists of the creation of an image using the 3D skeletal information. At the following, we will refer to this image as "signal image." The motion of each joint coordinate may be considered as an 1D signal, changing over time. Since 25 joints are available, each having 3 coordinates, from each video sequence 75 such signals are available. Of course, it should be intuitive that different actions may typically have significantly different duration. Also, it should be expected that the same action, when performed by different subjects, may have different duration. Of course, even instances that correspond to the same action, when performed by the same subject are expected to have similar, yet not the exactly the same duration. To address the aforementioned problem of temporal variability between actions and between users, interpolation is necessary. Therefore, by imposing a linear interpolation step, we set the duration of all action instances to $T_a = 159$. Upon concatenation of the 75 interpolated joint coordinate signals, the size of the resulting signal image is fixed and equal to 159×75. A signal image is illustrated in Fig. 3.

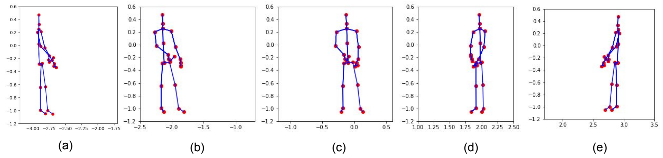

Figure 2. A skeleton rotated by angle θ: a) $\theta = 90°$, b) $\theta = 45°$, c) $\theta = 0°$ (raw skeleton), d) $\theta = -45°$, e) $\theta = -90°$. For illustrative purposes, depth information, i.e., z-coordinate has been discarded.

C. Geometric Transformation and Data Augmentation

In this work, our main goals are to assess the effects of a) data augmentation; and b) viewpoint alignment to the classification accuracy in a human action recognition task. To these goals, we opted for rotations of captured skeletons. We considered a camera setup that is composed of three cameras. One camera is placed directly at the front of the test subject, while the other two are placed at its left and right, respectively. Under the assumption that all cameras are placed at the same distance to the test subject, i.e., at the perimeter of an imaginary circle, each camera may be aligned to any other with a simple rotation transformation. Such a transformation by an angle θ is described by [20]:

$$\mathbf{R}_y(\theta) = \begin{bmatrix} cos\theta & 0 & sin\theta \\ 0 & 1 & 0 \\ -sin\theta & 0 & cos\theta \end{bmatrix} . \quad (1)$$

Therefore, to align any two given skeletons captured from different cameras, simultaneously, we may rotate each 3D joint by an angle θ, about the y-axis, complying to the Cartesian 3D coordinate system used by Kinect v2. Of course, the angle θ of the rotation, depends on the initial camera position setup and on which two cameras are used for training and testing. For example, to align a camera placed at the subject's left, at an angle θ_L and a camera placed at the subject's right, at an angle θ_R, we should apply $\mathbf{R}_y(\theta_R - \theta_L)$.

For data augmentation, our goal is to assist training of our network, by providing rotated instances of samples taken by a given camera. For viewpoint alignment, we aim to test the classification accuracy of our approach by providing images rotated towards different angles. More details regarding the evaluation protocol we followed are presented in subsection IV-B. Moreover, in Fig. 2 we illustrate a skeleton which is rotated by all angles used throughout our experiments.

D. Activity Images

From each signal image we create an "activity" image, by applying the 2D Discrete Sine Transform (DST), which in our previous work [16] showed the best performance among four of the most popular transforms. Note that we preserve only the magnitude of the transformation, while we discard its phase. Note that DST is further processed by normalizing using the orthonorm. Obviously, in all cases the result is a 2D signal spectrum. In Fig. 3 we illustrate an example signal image and the corresponding activity image. Moreover, indicative activity images for a set of 11 classes are depicted in Fig. 4.

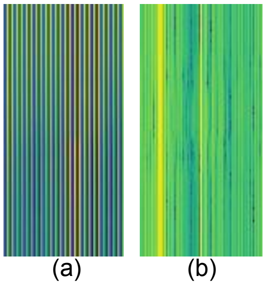

Figure 3. a) A signal image; b) the corresponding activity image. Figure best viewed in color.

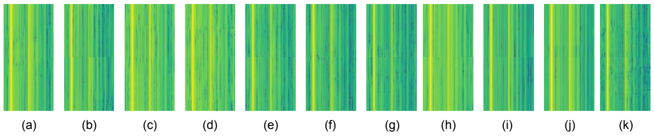

Figure 4. Examples of activity images from 11 classes for the DST transform. a) eat meal/snack; b) falling; c) handshaking; d) hugging other person; e) make a phone call/answer phone; f) playing with phone/tablet; g) reading; h) sitting down; i) standing up; j) typing on a keyboard; k) wear jacket. Figure best viewed in color.

E. CNN Architecture

The CNN that we use for experiments is trained using the activity images and its architecture has been proposed and evaluated in our previous work [16]. It is presented in detail in Fig. 5. The first convolutional layer filters the 159×75 input activity image with 32 kernels of size 3×3. The first pooling layer uses max-pooling to perform 2×2 subsampling. Then, the second convolutional layer filters the resulting 76×34 image with 64 kernels of size 3×3, followed by a second pooling layer, which also uses max-pooling to perform 2×2 sub-sampling. A third convolutional layer filters the resulting 36×15 image with 128 kernels of size 3×3 and a third pooling layer uses max-pooling to perform 2×2 sub-sampling. Then, a flatten layer transforms the output of the last pooling to a vector, which is then used as input to a dense layer using dropout. Finally, a second dense layer produces the output of the network.

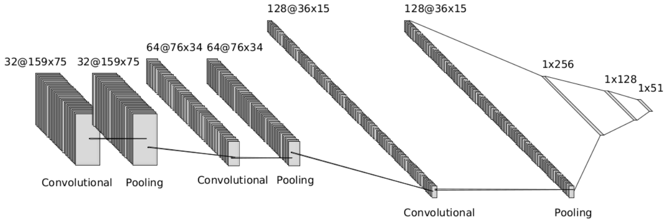

Figure 5. The CNN architecture used for classification of the activity images.

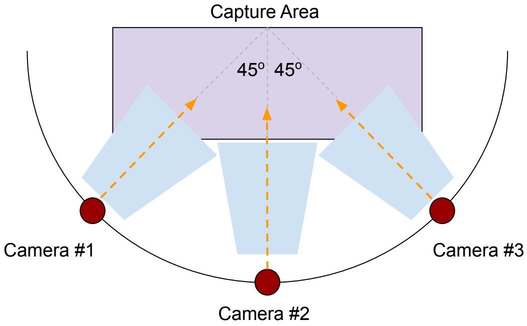

Figure 6. Camera setup of the PKU-MMD dataset. Cameras #1, #2, #3 correspond to L, M, R, respectively.

IV. Experiments

A. Data Set

For the experimental evaluation of our approach we used the PKU-MMD dataset [11]. It is a large-scale benchmark dataset that focuses on human action understanding. It contains approx. 20K action instances from 51 action categories, spanning into 5.4M video frames. For the data collection, 66 human subjects have been involved. Moreover, each action has been recorded by 3 camera views, namely L (left), M (middle) and R (right). As illustrated in Fig. 6 fixed angles are used, i.e., $-45°$, $0°$ and $+45°$. Note, that the height of all cameras is the same and remains fixed, while the area, within which users perform actions is pre-determined. The Microsoft Kinect v2 camera was used for all recordings, and for each action instance the following where provided: a) raw RGB video sequences depicting one or more test subjects performing an action; b) depth sequences, i.e., depth information of the aforementioned RGB sequences; c) infrared radiation sequences of the aforementioned sequences; and d) extracted 3D positions of human skeleton joints.

B. Evaluation Protocol

Our experiments are divided into two phases and each phase is divided into two parts. To begin with, in Phase 1, we create activity images simply by transforming the raw signal image by applying the DST transform. In Phase 2 we include the extra step of rotation for aligning skeletal information prior to applying a transform. Note that during Phase 2, due to data augmentation we use up to 5 times more data for training and testing. In Phase 2, we also include many different angles in our training and evaluation process in order to determine in practice the best rotation angle for maximum accuracy optimization.

Also, in the first part of each phase our goal has been to assess whether the proposed approach may be used for ambient assistive living scenarios and more specifically for the recognition of ADLs. Therefore, we selected 11 out of the 51 classes of PKU-MMD, which we believe are the most close to ADLs or events in such a scenario. The selected classes are: *eat meal snack*, *falling*, *handshaking*, *hugging other person*, *make a phone call*, *answer phone*, *playing with phone-tablet*, *reading*, *sitting down*, *standing up*, *typing on a keyboard* and *wear jacket*. In the second part, we performed experiments with the whole dataset, i.e., with all 51 classes. In both parts, we worked only based on the skeletal data, discarding RGB, depth and infrared information.

1) Phase 1: The evaluation protocol we followed, during Phase 1 of our experiments, is as follows:

i. *Experiments per camera position (single-view)*: in this case both training and testing sets derived from the same viewpoint e.g., L used for training and testing purposes.
ii. *Cross-view experiments*: at least two different camera viewpoints were used for training and only one viewpoint for testing or 2 different viewpoints were used, one for training and one for testing, e.g., L used for training, M or R used for testing. The goal of these experiments was to test the robustness of the proposed approach in terms of transformation (i.e., a rotation), which could simulate a real-life case of abrupt viewpoint changes, typically occurring in assistive living environments.
iii. *Cross-subject experiments*: subjects were split in training and testing groups, i.e., each one was a member of exactly one of these groups. The goal of these experiments was to test the robustness of our approach into intra-class variations. In real-life situations this is expected to happen when a system is trained e.g., within a laboratory environment and is deployed into a real ambient-assistive living environment.

2) Phase 2: The evaluation protocol of Phase 2 of our experiments was similar to the one of Phase 1. In this case, experiments included cross-view and cross-subject cases and were conducted in 2 parts, i.e., for the aforementioned handpicked 11 classes and for the whole set of 51 classes. However, the difference lies in the training and testing sets used. Based on the camera setting of the PKU-MMD data, and the coordinate system used by the Microsoft Kinect v2 camera, we opted for signal image rotations as discussed in subsection III-C. More specifically, we performed rotations using $\theta \in \{\pm 45°, \pm 90°\}$. This means that L signal images rotated by $-45°$ and $-90°$, align to M, R, respectively, M signal images $45°$ and $-45°$ align to L, R, respectively, while R signal images rotated by $45°$, $90°$ align to M, L, respectively. Obviously, the number of available images is multiplied by 5.

Our goal is to use this augmented data set, so as to accomplish an increase in performance, when comparing with Phase 1. In other words, our goal is given a certain camera viewpoint, to provide more reliable recognition of actions by training with an augmented data set, which includes all aligned images resulting from the aforementioned rotations. However, the following obvious question rises from the aforementioned statement: What if the given camera setting is *unknown*, while we have no knowledge regarding which camera viewpoint(s) has(have) been used for training and from which viewpoint originates a given testing sequence? Bearing this question in mind, our decision was to expand our protocol by applying all the aforementioned rotations to all the original signal images. More specifically, our experiments were organized as follows:

Table I
PHASE 1 SINGLE-VIEW RESULTS. L. R, M DENOTE LEFT, RIGHT AND MIDDLE CAMERA ANGLES, RESPECTIVELY. NUMBERS DENOTE CLASSIFICATION ACCURACY.

Part	L	R	M
11	0.84	0.87	0.86
51	0.68	0.76	0.72

we used the augmented set of samples deriving from a single or multiple cameras for training. For testing, we performed separate experiments using original samples and augmented ones for the remaining cameras.

C. Results

In Table I we present single-view results from Phase 1. Moreover, in Tables II and III we present cross-view and cross-subject results of Phases 1 and 2, respectively. For each experiment, we calculated the classification accuracy. Note that each experiment was performed 10 times, and in all Tables we depict mean accuracy for each.

Regarding Phase 1 results, we may observe that single-view accuracy achieved is satisfactory in both parts. However, a significant drop of performance is observed when one view is used for training and another for testing. Moreover, a smaller drop of performance is observed when two views are used for training and the remaining one for testing. Notably, when L, R are used for training and M for testing, performance is comparable to the single-view case.

Regarding Phase 2 results, we should emphasize the following: a) in every case, a significant boost of performance is achieved, compared to Phase 1 results; b) in most cases, best accuracy is achieved when the testing viewpoint is "aligned" with the training one, e.g., when L is used for training, while M_{+45° for testing; c) a significant boost of performance is also observed in cross-subject case. Therefore, we may argue that experiments justify our expectations regarding both data augmentation and viewpoint alignment.

D. Implementation Details

For the implementation of the CNN we have used Python 3.6 and more specifically, the Keras framework [4] running on top of Tensorflow [1]. All data pre-processing and processing steps have been implemented in Python 3.6 using NumPy 6 [15], SciPy 7 [7] and OpenCV 3.8 [3].

V. CONCLUSIONS AND FUTURE WORK

In this paper we presented a novel approach which aims to recognize human actions in video sequences. Our approach is based on a representation of skeletal 3D motion, using spectral image transformations, while classification is performed by a CNN. More specifically, the model was trained on images that resulted upon a) concatenation of raw 1D signals corresponding to 3D motion of skeletal joints; and b) application of a transform to the created image. A further pre-processing, i.e., a geometric rotation was imposed and significantly improved the accuracy while providing robustness to our approach. Our approach was evaluated using a state-of-the-art, challenging human action dataset, and for single-view, cross-view and cross-subject cases, so as to assess the robustness of our approach. Our experimental results indicate that the proposed approach may be successfully applied to real-like, monitoring problems, such as the ones presented in assistive living environments. We also experimentally verified that the proposed alignment step leads to an increase of accuracy. More specifically, we demonstrated a boost of performance in the cross-view and single-view cases, achieving results comparable to when the same camera viewpoint is used for training and testing.

Among our plans for the future are the following: a) investigation on methods for creating the signal image. To this goal we could experiment with other types of sensors, e.g., wearable inertial measurement units etc. b) investigation on image processing methods both for processing and for transforming the signal image to the activity image; c) exploitation of other typically available modalities in the process. Typical data sets provide raw RGB footage, depth data or even infrared sequences, which could be used to improve accuracy; d) investigation of other geometric pre-processing steps, such as alignment on 3-D planes; e) experimentation with other deep architectures, such as Long Short-Term Memory (LSTM) recurrent neural networks. However, we feel that it is equally important to further evaluate our approach using other large-scale data sets and also within real-life environments.

ACKNOWLEDGMENT

This project has received funding from the Hellenic Foundation for Research and Innovation (HFRI) and the General Secretariat for Research and Technology (GSRT), under grant agreement No 273 (Funding Decision:$\Gamma\Gamma ET122785/I2/19$-07-2018)

REFERENCES

[1] M. Abadi, et al., *Tensorflow: A system for large-scale machine learning*. In Proc. of 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), 2016.

[2] S. Berretti, M. Daoudi, P. Turaga and A. Basu, *Representation, analysis, and recognition of 3D humans: A survey*. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), 14.1s:16, 2018.

[3] G. Bradski, *The OpenCV Library*. Dr. Dobb's Journal of Software Tools, 2000.

[4] F. Chollet, *keras-team/keras. [online] GitHub*. Available at: https://github.com/fchollet/keras [Accessed 17 June 2019].

[5] . Du, Y. Fu and L. Wang, *Skeleton based action recognition with convolutional neural network.* In Proc. of 3rd IAPR Asian Conference on Pattern Recognition (ACPR) (pp. 579-583). IEEE, 2015.

[6] Y. Hou, Z. Li, P. Wang and W. Li, *Skeleton optical spectra-based action recognition using convolutional neural networks.* IEEE Transactions on Circuits and Systems for Video Technology, 28(3), 807-811, 2018.

[7] E. Jones, E. Oliphant, P. Peterson, et al.,*SciPy: Open Source Scientific Tools for Python*. http://www.scipy.org/ [Accessed 20 May 2019]

[8] Q. Ke, S. An, M. Bennamoun, F. Sohel and F. Boussaid. *Skeletonnet: Mining deep part features for 3-d action recognition*. IEEE signal processing letters, 24(6), 731-735, 2017.

[9] A. Krizhevsky, I. Sutskever, and G. E Hinton, *Imagenet classi[U+FB01]cation with deep convolutional neural networks.* Advances in neural information processing systems, pp. 1097-1105, December 2012.

Table II
CROSS-VIEW RESULTS OF PHASES 1 AND 2. RESULTS DENOTE CLASSIFICATION ACCURACY. DIFFERENCES ARE CALCULATED BETWEEN THE ACCURACY OF PHASE 1 AND THE MEAN ACCURACY OF PHASE 2. NOTE THAT IN PHASE 1 THE ORIGINAL SAMPLES HAVE BEEN USED FOR TRAINING AND TESTING, WHILE IN PHASE 2 THE AUGMENTED SAMPLES HAVE BEEN USED FOR TRAINING AND TESTING.

Train	**Test**	**Part**	**Phase 1**	**Phase 2**						**Difference (%)**
				$-90°$	$-45°$	$0°$	$+45°$	$+90°$	**mean**	
L	**M**	**11**	0.72	0.82	0.86	0.93	0.93	0.86	0.88	+22.2
		51	0.57	0.56	0.61	0.72	0.76	0.66	0.66	+16.1
L	**R**	**11**	0.43	0.59	0.65	0.85	0.78	0.55	0.68	+59.07
		51	0.26	0.34	0.33	0.64	0.49	0.27	0.41	+59.23
M	**L**	**11**	0.64	0.84	0.84	0.85	0.90	0.91	0.87	+35.63
		51	0.49	0.59	0.58	0.60	0.72	0.73	0.64	+31.43
M	**R**	**11**	0.63	0.83	0.87	0.90	0.89	0.87	0.87	+38.41
		51	0.60	0.68	0.61	0.75	0.71	0.63	0.68	+12.67
R	**L**	**11**	0.39	0.59	0.53	0.69	0.83	0.86	0.70	+79.49
		51	0.23	0.28	0.25	0.38	0.52	0.65	0.42	+80.87
R	**M**	**11**	0.63	0.93	0.82	0.88	0.93	0.87	0.89	+40.63
		51	0.56	0.75	0.57	0.64	0.71	0.61	0.66	+17.14
L,R	**M**	**11**	0.82	0.93	0.94	0.94	0.95	0.96	0.94	+15.12
		51	0.76	0.81	0.78	0.82	0.84	0.81	0.81	+6.84
L,M	**R**	**11**	0.62	0.81	0.81	0.84	0.90	0.81	0.83	+34.52
		51	0.55	0.66	0.60	0.76	0.72	0.59	0.67	+21.10
R,M	**L**	**11**	0.60	0.78	0.74	0.81	0.88	0.88	0.82	+36.33
		51	0.52	0.59	0.57	0.64	0.73	0.75	0.66	+26.15

Table III
CROSS-SUBJECT RESULTS OF PHASES 1 AND 2. NUMBERS DENOTE CLASSIFICATION ACCURACY.

Part	**Phase 1**	**Phase 2**	**Difference**
11	0.81	0.95	+17.28
51	0.73	0.76	+4.11

[10] C. Li, Y. Hou, P. Wang an W. Li, *Joint distance maps based action recognition with convolutional neural networks*. IEEE Signal Processing Letters, 24(5), 624-628, 2017.

[11] C. Liu, Y. Hu, Y. Li, S. Song and J. Liu, *PKU-MMD: A large scale benchmark for continuous multi-modal human action understanding*. arXiv preprint arXiv:1703.07475, 2017.

[12] M. Liu, H. Liu and C. Chen, *Enhanced skeleton visualization for view invariant human action recognition.* Pattern Recognition, 68, 346-362, 2017.

[13] E. Mathe, A. Mitsou, E. Spyrou and P. Mylonas, *Arm Gesture Recognition using a Convolutional Neural Network*. In Proc. of Int'l Workshop on Semantic and Social Media Adaptation and Personalization (SMAP), 2018.

[14] E. Mathe, A. Maniatis, E. Spyrou and Ph. Mylonas, *A Deep Learning Approach for Human Action Recognition using Skeletal Information*. In Proc. of World Congress Genetics, Geriatrics and Neurodegenerative Diseases Research (GeNeDiS), 2018.

[15] T. E. Oliphant, *A guide to NumPy*.Trelgol Publishing, USA, 2006.

[16] A. Papadakis, E. Mathe, I. Vernikos, A. Maniatis, E. Spyrou and Ph. Mylonas, *Recognizing Human Actions using 3D Skeletal Information and CNNs*. In Proc. of Int'l Conf. on Engineering Applications of Neural Networks (EANN), 2019.

[17] C. Schuldt, I. Laptev and B. Caputo, *Recognizing human actions: a local SVM approach*. In Proc. of the 17th Int'l Conf. on Pattern Recognition (ICPR), 2004.

[18] A. Shahroudy, J. Liu, T. Ng and G. Wang, *NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis*. In Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2016.

[19] J. Shotton, A. W. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman and A. Blake, *Real-time human pose recognition in parts from single depth images*. In Proc. of Int'l Conf. on Computer Vision and Pattern Recognition, 2011.

[20] T. Theoharis, G. Papaioannou, N. Platis and N.M. Patrikalakis, *Graphics and visualization: principles & algorithms*. AK Peters/CRC Press, 2008.

[21] P. Wang, W. Li, C. Li, and Y. Hou, *Action recognition based on joint trajectory maps with convolutional neural networks*. Knowledge-Based Systems, vol 158, pp. 43-53, 2018.

[22] P. Zhang, C. Lan, J. Xing, W. Zeng, J. Xue and N. Zheng, *View adaptive recurrent neural networks for high performance human action recognition from skeleton data.* In Proc. of the IEEE Int'l Conference on Computer Vision (pp. 2117-2126), 2017.