Human Activity Recognition under Partial Occlusion

Ioannis-Aris Kostis[1], Eirini Mathe[2], Evaggelos Spyrou[1,3], and Phivos Mylonas[2]

[1] Department of Computer Science and Telecommunications, University of Thessaly, Lamia, Greece
ya.kostis@gmail.com, espyrou@uth.gr
[2] Department of Informatics, Ionian University, Corfu, Greece
{cmath17, fmylonas}@ionio,gr
[3] Institute of Informatics and Telecommunications, National Center for Scientific Research – "Demokritos," Athens, Greece

Abstract. One of the major challenges in Human Activity Recognition (HAR) using cameras is occlusion of one or more body parts. However, this problem is often underestimated in contemporary research works, wherein training and evaluation is based on datasets shot under laboratory conditions, i.e., without some kind of occlusion. In this work we propose an approach for HAR in the presence of partial occlusion, i.e., in case of up to two occluded body parts. We solve this problem using regression, performed by a deep neural network. That is, given an occluded sample, we attempt to reconstruct the missing information regarding the motion of the occluded part(s). We evaluate our approach using a publicly available human motion dataset. Our experimental results indicate a significant increase of performance, when compared to a baseline approach, wherein a network that has been trained using non-occluded samples is evaluated using occluded samples. To the best of our knowledge, this is the first research work that tackles the problem of HAR under occlusion as a regression problem.

Keywords: human activity recognition · deep learning · regression .

1 Introduction

Human activity recognition (HAR) still remains one of the most challenging computer vision-related problems. It may be defined as the recognition of some human behaviour within an image or a video sequence. An activity (or "action") may be defined as a type of motion performed by a single human, taking place within a relatively short time period (however, not instant) and involving multiple body parts [23]. This informal definition differentiates activities from gestures; the latter are typically instant and involve at most a couple of body parts. Similarly, interactions may involve either a human and an object or two humans and group activities involve more than one humans. Typical HAR applications include, yet are not limited to video surveillance, human-computer/robot interaction, augmented reality (AR), ambient assisted environments, health monitoring, intelligent driving, gaming and immersion, animation, etc. [23,20,3].

There exist several HAR approaches that are based on either wearable sensors or sensors installed within the subject's environment.In the former case, the most popular ones include smartwatches, hand/body worn sensors, smartphones, etc. Moreover, in the latter case, typical sensors include video/thermal cameras microphones, infrared, pressure, magnetic, RFID sensors [5] etc. However, it has been shown that wearable sensors are not preferred by the users, while their usability is below average [18,12]. Moreover, overloading the users' environment with a plethora of sensors may be an expensive task, requiring in some cases many interventions in home furniture and/or appliances, e.g., in case of a home environment. Therefore, several low-cost solutions tend to be based solely on cameras, detecting activities using the subjects' motion. Although such approaches are low-cost and demonstrate more than satisfactory performance in laboratory conditions, in real-life situations they suffer from viewpoint and illumination changes and occlusion.

In previous work [19] we dealt with the problem of viewpoint invariance and demonstrated that the decrease of accuracy due to viewpoint changes may be limited when using more than one cameras. Also, recent advances in technology have allowed for camera sensors that also capture depth information and perform significantly better in low-light conditions. Therefore, from the three aforementioned problems, occlusion is the one that introduces most limitations. Also in previous work [7] we assessed how partial occlusion of the subject affects the accuracy of recognition. We simulated occlusion by removing parts of captured visual data and showed that partial occlusion of the subject, in certain cases significantly affected the accuracy of recognition. To tackle this limitation, in this work we aim to reconstruct occluded data, upon formulating this problem as a regression task. We use a deep neural network approach, whose input is a human skeleton, with one or more body parts removed, so as to simulate occlusion. The network is trained to output the skeleton upon estimating the missing parts. We demonstrate that this approach is effective and may significantly increase accuracy.

The rest of this paper is organized as follows: In Section 2 we present research works that aim to assess or even tackle the effect of occlusion in HAR-related scenarios. Then, in section 3 we present the proposed regression methodology. Experimental results of are presented in section 4. Finally, conclusions are drawn in section 5, wherein plans for future work are also presented.

2 Related Work

During the last few years, a plethora of research works focusing on HAR, based on 2D representations of skeletal data have been presented [6,22,9,14,15,11]. Moreover, a may be found in [23]. However, although it is widely accepted that occlusion consists one of the most important factors that compromise the performance of HAR approaches [10], resulting to poor or even unusable results, few are those works that focus either on studied its effects on the performance of recognition or even attempt to overcome them.

To begin with, in the work of Iosifidis et al. [10], a multi-camera setup, surrounding the subject was used for HAR. In order to simulate occlusion, they first trained their algorithm using data from all available cameras and then evaluate using a randomly chosen subset. More specifically, they made the assumption that due to occlusion, not all cameras were simultaneously able to capture the subject's motion. However, we should note that in all cases more than one cameras were able to capture the whole body of the subjects. Also, recognition of a given activity took place upon combining results only from those cameras that are not affected at any means by occlusion. In the work of Gu et al. [8], randomly generated occlusion masks were used in both training and evaluation. Note that each mask caused the occlusion of more than one 2D skeletal joints. Then, and in order to reconstruct the skeleton, they used a regression network. Liu et al. [17] studied two augmentation strategies for modelling the effect of occlusion. The first discarded independent keypoints, while the second discarded structured sets of keypoints, i.e., those composing main body parts. Note that in this work occluded samples were included in the training process. Moreover, the authors herein made the assumption that the torso and the hips were always visible. Their recognition approach was based on learning view-invariant, occlusion-robust probabilistic embeddings. Similarly, Angelini et al. [2] also included artificially occluded samples within the training process. In that case, samples were created by randomly removing body landmarks according to a binary Bernoulli distribution. Their recognition approach was based on pose libraries which included several pose prototypes. When dealing with missing body parts, they exploited the aforementioned libraries either by matching occluded sequences to pre-defined prototypes, based on high-level features, or by filling missing parts upon searching through the pose libraries. In case of short-time occlusions, they used an interpolation approach.

Finally, in previous work [7] we performed a study, wherein our main goal was to assess the effect of occlusion of body parts, within a HAR approach. We created artificial occluded activity samples, by manually removing one or two body parts (i.e., upon removing subsets of skeleton joints). We made the following assumption: occlusion was continuous during the whole duration of activity and concerned the same part(s). For HAR, we used a deep neural network, that had been trained using only non-occluded samples, i.e., contrary to [8], [17], [2]. Also, in our study the whole skeleton was never "visible" as it was in the work presented in [10]. Finally, Gu et al. [8] proposed a regression-based approach which was limited to pose estimation.

3 Methodology

3.1 Skeletal Data

As in previous work [19], [7], the proposed approach uses as input 3D trajectories of human skeletons. In 3D HAR problems, subjects perform actions in space and over time. We consider skeleton representations as sets of 3D joints. We use skeleton data that have been captured using the Microsoft Kinect v2 RGB/depth camera.[4] A human skeleton comprises of 25 joints, organized as a graph; each node corresponds to a body part such as arms, legs, head, neck etc., while edges follow the body structure, appropriately connecting pairs of joints. In Fig. 1 we illustrate a skeleton extracted using Kinect. Note that joints are shown as being grouped; each group corresponds to a body part, i.e., an arm, a leg or the torso. In the context of this work, an activity is considered to be a temporal sequence of 3D skeleton representations. For the sake of explanation, a visual example of an activity is illustrated in Fig. 2.

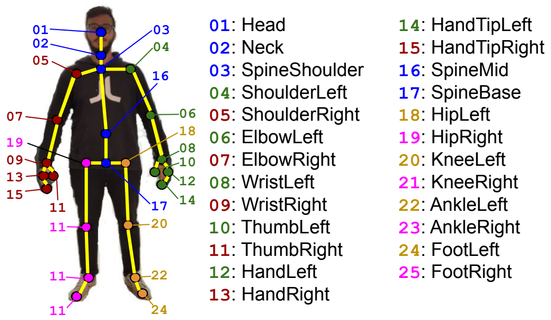


Fig. 1: The 25 skeletal joints extracted by Microsoft Kinect, divided into five main body parts – blue: torso, red: left hand, green: right hand, magenta: left leg, orange: right leg.

3.2 Occlusion

As it has already been mentioned in Section 1, occlusion may compromise the performance of HAR, in real-life scenarios. Within the context of several applications such as ambient assisted environments, AR environments etc., occlusion typically occurs due to e.g., activities taking place behind furniture, or e.g., due to the presence of more than one people in the same room. Of course, it should be obvious that occlusion of e.g., the legs when the subject performs the action “kicking” results to a significant loss of visual information, which in turn may result to failure of recognition. Although the aforementioned example is quite extreme, it is common sense that partial occlusion may hinder the effectiveness of HAR approaches. We should herein note that most large-scale public motion-based datasets such as the PKU-MMD dataset [16] have been created under ideal laboratory conditions, thus occlusion is prevented. Thus, since the creation of a large scale dataset is a time consuming task, we decided to follow an approach such as the one of Gu et al. [8]. More specifically, we manually discard subsets of joints that correspond to body parts, assuming that the these parts remain

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

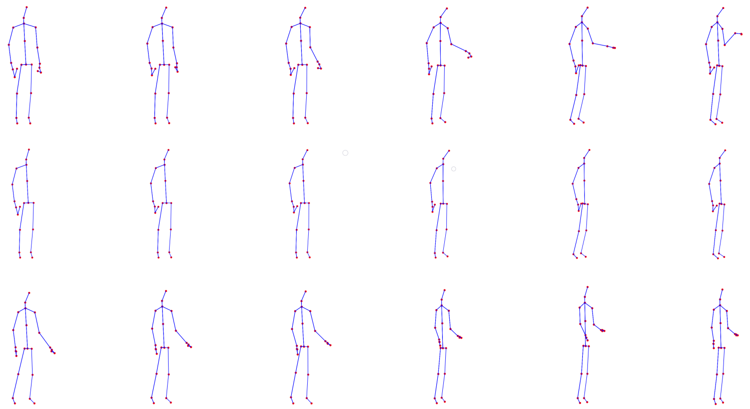


Fig. 2: Example skeleton sequences of the activity *handshaking*. First row: skeletons include all 25 joints; Second row: joints corresponding to left arm have been discarded; Third row: joints corresponding left arm have been reconstructed.

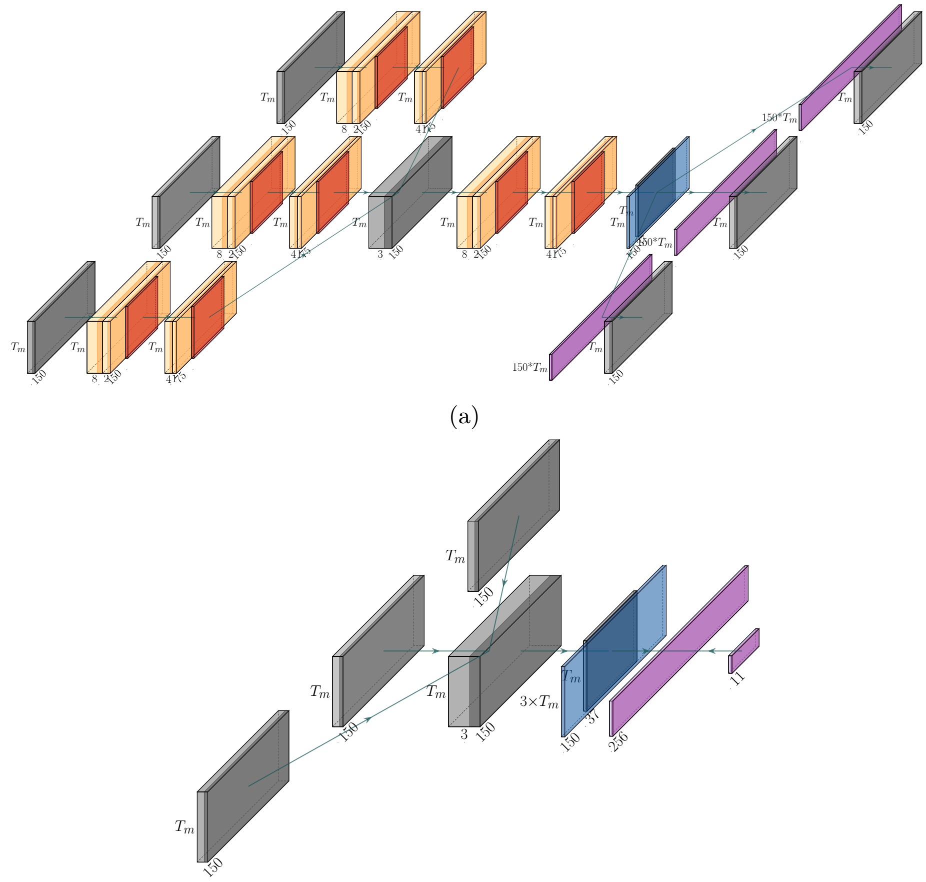
occluded during the whole action. For the sake of explanation, a visual example of an activity upon occlusion is illustrated in Fig. 2.

3.3 Regression of Skeletal Data

The input of our approach consists of temporal sequences of 3D skeleton data, i.e., as described in subsection 3.1. Upon imposing a linear interpolation step between consecutive timeframes so as to address temporal variability of activities, we set the length of all activity examples equal to T_m, i.e., to the size of the longest one in duration. Note that if the desired length is not reached upon one interpolation step, the process is repeated until the desired length is reached. As we will mention in Section 4, we use a dataset that has been captured using 3 cameras. Therefore, as we wish to exploit all possible information, we use the corresponding 3 skeleton sequences as input. We also assume that in every case of occlusion, the same missing body part(s) is (are) occluded in all 3 sequences.

The core philosophy of our approach is that since occlusion practically causes missing values (i.e., in our case some of joints of the skeleton are removed), we may formulate the problem of "reconstructing" those missing values as a regression task. More specifically, let $\mathbf{X}$ denote the original skeleton sequence and $\mathbf{X_o}$ the sequence resulting upon occlusion. The goal of regression is ideally to estimate a function f, so that $\mathbf{X}_r = f(\mathbf{X_o}) + \epsilon$, where $\mathbf{X}_r$ denotes the reconstructed skeleton sequence and ϵ is some error value, to be minimized.

To this goal, we use a Convolutional Recurrent Neural Network (CRNN) model, whose aim is to estimate the missing (occluded) data (joints). Its architecture is illustrated in Fig. 3a and is described in short as follows: The input of the network constitutes of sequential data from 3 cameras. Each camera provides a skeletal sequence under a different viewpoint. Given that in every sequence up



(a)

(b)

Fig. 3: (a) The CRNN that has been used for regression of skeletal joint sequences; (b) The RNN that has been used for classification. Layers have been colored as follows: Gray: input/output, concatenated, Light Orange: 2D convolutional, orange: max pooling, Light Blue: (input layer of) LSTM, Purple: fully connected (dense). Figure best viewed in color.

to 2 skeletons are included (i.e., in case of interactions between 2 subjects), and each skeleton comprises 25 3-D joints, and the duration of the sequence is T_m, input layer size is $T_m \times 150$. Those three input branches are each filtered by a stack of 2 2-D convolutional layer, followed by a max-pooling layer that performs 1×2 sub-sampling. This process repeats after the three branches are concatenated into a single tensor. This single tensor is again filtered by a stack of 2 2-D convolutional layer, followed by a max-pooling layer that performs 1×2 sub-sampling. The output of this layer constitutes the input to an LSTM layer, whose goal is to harness temporal information of skeletal data. Then, 3 parallel dense layers of size $T_m \times 150$ follow. They are ultimately reshaped to three $T_m \times 150$ output layers. For loss computation, the Mean Square Error (MSE) has been used.

At this point we would like to note that the reason for the use of an asymmetrical kernel (i.e., 1×2) is that while being sub-sampled this way, information is compressed only along the spatial coordinates' axes, leaving temporal information intact. We experimentally verified that this kernel choice led to a significant improvement of the performance of the network.

The occluded data $\mathbf{X}_o$ are given as input in both training and testing phases of the network. Also, the targets of the network are the non-occluded data $\mathbf{X}$; these data are to be estimated by the network, i.e., its output are reconstructed data $\mathbf{X}_r$. Thus, the network is train to learn f, while minimizing ϵ. As we mentioned in subsection 3.1, each skeleton joint has its own id. Therefore, in a real-life application, we could easily identify missing (occluded) joints. Bearing this in mind, we opted to train one network per occlusion case, ending up with 8 different networks. Therefore, given an input skeletal sequence, it is fed to the appropriate network, upon identifying missing joints.

At this point, the trained network serves as a mean to reconstruct missing skeletal data of a given skeletal sequence. For the sake of explanation, a visual example of an activity upon reconstruction is illustrated in Fig. 2. Thus, we are able to proceed with its classification into one of the pre-defined classes. This is performed using a second network, whose architecture is based on an LSTM layer and is illustrated in Fig. 3b. As expected, data collected from three cameras constitute again the input of this network. The three branches are concatenated into a single tensor, serving as input to the LSTM layer. The latter is followed by another dense layer of size 11, i.e., equal to the number of classes and constitutes the output layer of the network. During training, the non-occluded data $\mathbf{X}$ serve as input data to the network, thus no occlusion information is used. During testing, its input is a reconstructed skeletal sequence $\mathbf{X}_r$.

4 Experiments and Results

4.1 Dataset

Since to the best of our knowledge such a large scale dataset consisting of 3D skeletal data does not exist, we used part of the PKU-MMD dataset [16]. Note that this dataset consists of activities that have been recorded using Microsoft Kinect v2 sensor. In order to produce results comparable to the ones of our previous work [7], we have selected the same 11 classes, i.e.: *eat meal snack* (10), *falling* (11), *handshaking* (14), *hugging other person* (16), *make a phone call answer phone* (20), *playing with phone tablet* (23), *reading* (30), *sitting down* (33), *standing up* (34), *typing on a keyboard* (46) and *wearing a jacket* (48). Numbers in parentheses denote the corresponding class ids and will be used at the remaining of this paper. A total number of 1100 samples has been used for training, while 400 samples have been used for testing.

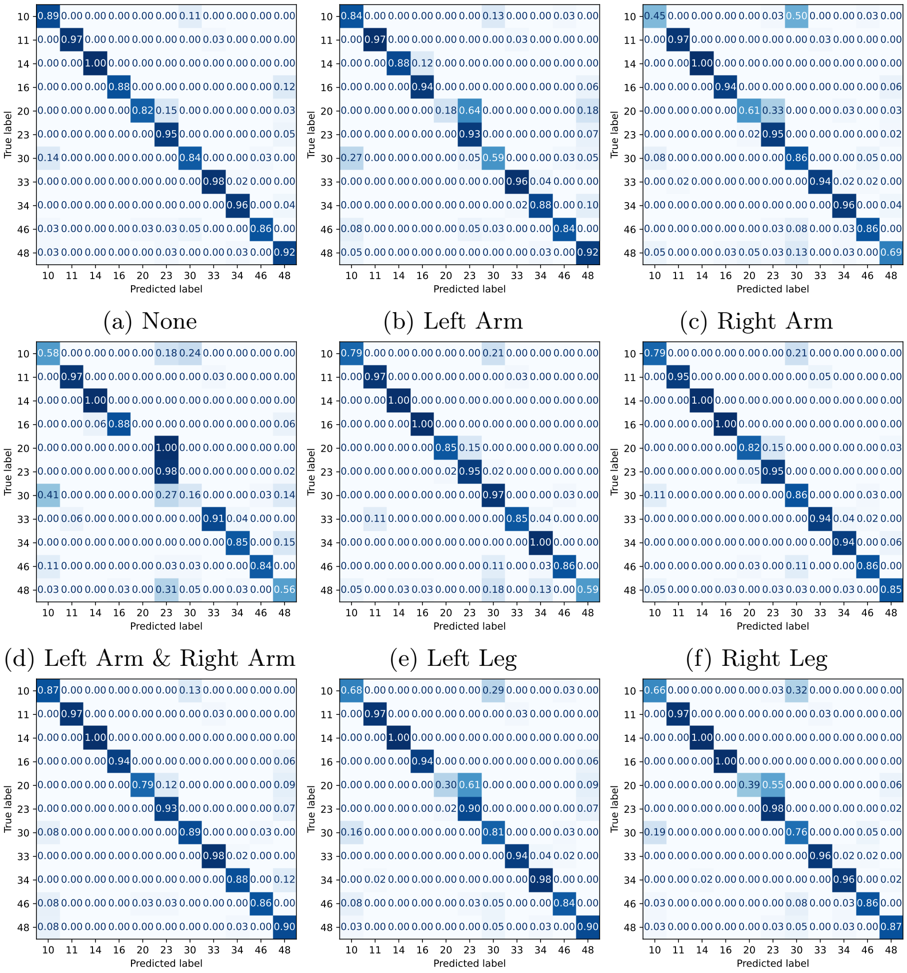
4.2 Experimental Setup and Network Training

Experiments were performed on a personal workstation with an Intel™i7 4770 4-core processor on 3.40 GHz and 16GB RAM, using NVIDIA™Geforce GTX 1050Ti GPU with 4 GB VRAM and Ubuntu 20.04 (64 bit). The deep architecture has been implemented in Python, using Keras 2.4.3 [4] with the Tensorflow 2.5 [1] backend. All data pre-processing and processing steps have been implemented in Python 3.9 using NumPy and SciPy. For the training of the estimator, we used the LeakyReLU activation function, except from the LSTM layer wherein the tanh function was used, and the last dense layer wherein linear activation function was used. For the training of the classifier, the LeakyReLU and tanh activation functions were used respectively, except from the last layer, wherein the softmax activation function was used. Moreover, we set the batch size to 5 and 10 for the training of the classifier and the estimator respectively. The Adam optimizer was utilized in both cases, the dropout was set to 0.3, set the learning rate to 0.001 and trained for 50 epochs, using the loss of the validation set calculated via MSE as an early stopping method, in order to avert overfitting. Moreover, since the duration of each activity was set to 150 frames, upon interpolation, the size of the input data was 3×150×150.

4.3 Results

For the experimental evaluation of the proposed methodology, we considered eight cases of body part removal, so as to simulate occlusion. More specifically, we removed one arm/leg, both arms/legs, one arm and one leg from the same side. For comparison, we also performed experiments without any body part removal, for comparisons. In every case we evaluated classification upon removal and upon reconstruction. Experimental results are depicted in Table 4.3. The weighted accuracy (WA) was 0.92 without any body part removal. Moreover, it ranged between 0.21–0.90 in case of some body part removal, while it ranged between 0.70–0.91 upon reconstruction. In 7 out of 8 cases, significant improvement was observed, in terms of WA, while performance was almost equal in case of removal of Left Leg. Intuitively, one should observe that the majority of the activities we used to evaluate our approach mainly consists of upper body motion (i.e., left and/or right arm). Upon careful observation of the samples of the datasets, this assumption has been verified. This is also reflected to the results of Table 4.3, wherein it may observed that in cases of occluded arms the improvement is significantly large, with most notable example the case of both arms, wherein WA improves from 0.21 to 0.70.

Upon careful observation of the confusion matrices depicted in Fig. 4, for each occlusion case we should notice the following, when comparing with the case where all joints had been used: a) in case of any occluded arm, class *make a phone call/answer phone* is often confused with *playing with phone/tablet* and class *eat meal/snack* is often confused with *reading*; b) in case of occluded left leg, class *wear jacket* is often confused with *reading* or *standing up*; and c) finally, in case of both arms occluded, 7 classes show adequate performance.



(a) None (b) Left Arm (c) Right Arm

(d) Left Arm & Right Arm (e) Left Leg (f) Right Leg

(g) Left Leg & Right Leg (h) Left Arm & Left Leg (i) Right Arm & Right Leg

Fig. 4: Normalized confusion matrices for classification (a) without removing any body part, (b)–(i) upon removing the body part(s) denoted in the caption of the corresponding subfigure.

5 Conclusions and Future Work

In this paper we presented an approach for human activity recognition under occlusion, which was based on a convolutional recurrent neural network model and used as input 3D skeleton joint sequences. We simulated occlusion by removing one or two body parts (i.e., sets of joints corresponding to arms and/or legs). By using the aforementioned model, we managed to reconstruct missing joints using regression. We showed that this way, we could achieve a significant boost

Table 1: Experimental results of the proposed approach. "Rec." and "Ref." denote reconstructed and reference case (see section 3). Acc, P, R, F_1, WA denote Accuracy, Precision, Recall, F_1 score and Weighted Accuracy, respectively. By "None" we denote the case wherein all body parts are included. LA, RA, LL, RL denote the occlusion of left arm, right arm, left leg, right leg, respectively. Numbers in bold indicate cases where the performance of the reconstructed data is improved over the one of the reference case.

		None	**LA**		**RA**		**LA&RA**		**LL**		**RL**		**LL&RL**		**LA&LL**		**RA&RL**	
Class	Metric	Baseline	Rec.	Ref.	Rec.	Ref.	Rec.	Ref.	Rec.	Ref.	Rec.	Ref.	Rec.	Ref.	Rec.	Ref.	Rec.	Ref.
10	Acc.	0.90	**0.84**	0.11	**0.45**	0.03	**0.58**	0.00	0.79	0.92	**0.79**	0.66	**0.87**	0.74	**0.68**	0.21	**0.66**	0.24
	P	0.83	**0.68**	0.29	**0.77**	0.17	**0.52**	0.00	**0.94**	0.78	**0.83**	0.81	0.79	0.82	**0.72**	0.31	0.74	0.75
	R	0.89	**0.84**	0.11	**0.45**	0.03	**0.58**	0.00	0.79	0.92	**0.79**	0.66	**0.87**	0.74	**0.68**	0.21	**0.66**	0.24
	F_1	0.86	**0.75**	0.15	**0.57**	0.05	**0.55**	0.00	**0.86**	0.84	**0.81**	0.72	**0.82**	0.78	**0.70**	0.25	**0.69**	0.36
11	Acc.	0.97	0.97	0.97	0.97	0.97	**0.97**	0.00	0.97	0.97	0.95	0.97	0.97	0.97	0.97	1.00	0.97	0.97
	P	1.00	**1.00**	0.84	0.97	1.00	**0.92**	0.00	0.86	1.00	1.00	1.00	1.00	1.00	**0.97**	0.54	**1.00**	0.88
	R	0.97	0.97	0.97	0.97	0.97	**0.97**	0.00	0.97	0.97	0.95	0.97	0.97	0.97	0.97	1.00	0.97	0.97
	F_1	0.99	**0.99**	0.90	0.97	0.99	**0.95**	0.00	0.91	0.99	0.97	0.99	0.99	0.99	**0.97**	0.70	**0.99**	0.92
14	Acc.	1.00	0.88	1.00	1.00	1.00	**1.00**	0.81	**1.00**	0.94	1.00	1.00	1.00	1.00	**1.00**	0.94	1.00	1.00
	P	1.00	**1.00**	0.94	1.00	1.00	0.94	1.00	0.94	0.94	**1.00**	0.84	**1.00**	0.84	**1.00**	0.88	**0.94**	0.76
	R	1.00	0.88	1.00	1.00	1.00	**1.00**	0.81	**1.00**	0.94	1.00	1.00	1.00	1.00	**1.00**	0.94	1.00	1.00
	F_1	1.00	0.93	0.97	1.00	1.00	**0.97**	0.90	**0.97**	0.94	**1.00**	0.91	**1.00**	0.91	**1.00**	0.91	**0.97**	0.86
16	Acc.	0.88	**0.94**	0.88	**0.94**	0.88	**0.88**	0.69	**1.00**	0.88	**1.00**	0.88	**0.94**	0.88	**0.94**	0.81	**1.00**	0.88
	P	1.00	0.88	0.93	0.94	1.00	**0.93**	0.85	0.94	1.00	**0.94**	0.93	**1.00**	0.93	**1.00**	0.76	**1.00**	0.82
	R	0.88	**0.94**	0.88	**0.94**	0.88	**0.88**	0.69	**1.00**	0.88	**1.00**	0.88	**0.94**	0.88	**0.94**	0.81	**1.00**	0.88
	F_1	0.93	**0.91**	0.90	**0.94**	0.93	**0.90**	0.76	**0.97**	0.93	**0.97**	0.90	**0.97**	0.90	**0.97**	0.79	**1.00**	0.85
20	Acc.	0.82	**0.18**	0.03	**0.61**	0.39	0.00	0.00	**0.85**	0.79	0.82	0.88	0.79	0.88	**0.30**	0.00	0.39	0.97
	P	0.96	**1.00**	1.00	**0.87**	0.13	0.00	0.00	**0.97**	0.96	**0.90**	0.48	**0.96**	0.47	**0.91**	0.00	**1.00**	0.16
	R	0.82	**0.18**	0.03	**0.61**	0.39	0.00	0.00	**0.85**	0.79	0.82	0.88	0.79	0.88	**0.30**	0.00	0.39	0.97
	F_1	0.89	**0.31**	0.06	**0.71**	0.19	0.00	0.00	**0.90**	0.87	**0.86**	0.62	0.87	0.90	**0.45**	0.00	**0.57**	0.27
23	Acc.	0.95	0.93	0.98	**0.95**	0.83	**0.98**	0.05	**0.95**	0.93	**0.95**	0.02	**0.93**	0.05	0.91	1.00	**0.98**	0.05
	P	0.85	**0.61**	0.55	**0.74**	0.27	**0.39**	0.06	0.89	0.93	0.87	1.00	0.89	1.00	**0.64**	0.45	**0.67**	0.17
	R	0.95	0.93	0.98	**0.95**	0.83	**0.98**	0.05	**0.95**	0.93	**0.95**	0.02	**0.93**	0.05	0.90	1.00	**0.98**	0.05
	F_1	0.90	**0.74**	0.70	**0.83**	0.40	**0.56**	0.05	0.92	0.93	**0.91**	0.05	**0.91**	0.09	**0.75**	0.62	**0.80**	0.07
30	Acc.	0.84	0.60	0.70	**0.87**	0.24	**0.16**	0.00	**0.97**	0.76	0.87	0.89	**0.89**	0.84	**0.81**	0.30	**0.76**	0.70
	P	0.84	**0.79**	0.36	0.54	0.69	**0.33**	0.00	0.64	0.80	**0.71**	0.65	**0.87**	0.67	**0.67**	0.24	0.62	0.65
	R	0.84	0.59	0.70	**0.86**	0.24	**0.16**	0.00	**0.97**	0.76	0.86	0.89	**0.89**	0.84	**0.81**	0.30	**0.76**	0.70
	F_1	0.84	**0.68**	0.48	**0.67**	0.36	**0.22**	0.00	0.77	0.78	**0.78**	0.75	**0.88**	0.75	**0.73**	0.27	0.68	0.68
33	Acc.	0.98	**0.96**	0.74	**0.94**	0.06	**0.91**	0.00	0.85	0.98	0.94	0.98	0.98	0.98	**0.94**	0.11	**0.96**	0.00
	P	0.98	0.96	0.97	**0.98**	0.75	**0.98**	0.00	**0.98**	0.98	0.96	0.98	0.98	0.98	0.98	1.00	**0.98**	0.00
	R	0.98	**0.96**	0.74	**0.94**	0.06	**0.91**	0.00	0.85	0.98	0.94	0.98	0.98	0.98	**0.94**	0.11	**0.96**	0.00
	F_1	0.98	**0.96**	0.84	**0.96**	0.11	**0.94**	0.00	0.91	0.98	0.95	0.98	0.98	0.98	**0.96**	0.20	**0.97**	0.00
34	Acc.	0.96	0.89	0.96	**0.96**	0.19	**0.46**	0.00	**1.00**	0.94	**0.94**	0.90	**0.89**	0.85	**0.98**	0.54	**0.96**	0.00
	P	0.96	0.94	0.96	**0.93**	0.10	**0.94**	0.00	0.87	0.96	0.94	0.96	0.96	0.98	0.94	1.00	**0.94**	0.00
	R	0.96	0.88	0.96	**0.96**	0.19	**0.85**	0.00	**1.00**	0.94	**0.94**	0.90	**0.88**	0.85	**0.98**	0.54	**0.96**	0.00
	F_1	0.96	0.91	0.96	**0.94**	0.32	**0.89**	0.00	0.93	0.95	**0.94**	0.93	**0.92**	0.91	**0.96**	0.70	**0.95**	0.00
46	Acc.	0.87	0.84	0.89	**0.87**	0.49	**0.84**	0.65	**0.87**	0.84	0.87	0.87	0.87	0.87	0.84	0.87	**0.87**	0.65
	P	0.97	**0.94**	0.56	**0.89**	0.31	**0.97**	0.08	**0.97**	0.97	0.94	1.00	0.97	0.97	**0.91**	0.33	**0.91**	0.49
	R	0.86	0.84	0.89	**0.86**	0.49	**0.84**	0.65	**0.86**	0.84	0.86	0.86	0.86	0.86	0.84	0.86	**0.86**	0.65
	F_1	0.91	**0.89**	0.69	**0.88**	0.38	**0.90**	0.14	**0.91**	0.90	0.90	0.93	0.91	0.91	**0.87**	0.48	**0.89**	0.56
48	Acc.	0.92	**0.92**	0.28	**0.69**	0.13	**0.56**	0.00	0.59	0.85	0.85	0.85	0.90	0.90	**0.90**	0.05	**0.87**	0.13
	P	0.84	0.68	0.82	**0.84**	0.71	**0.59**	0.00	**1.00**	0.89	**0.89**	0.62	**0.73**	0.64	**0.83**	0.67	0.89	1.00
	R	0.92	**0.92**	0.28	**0.69**	0.13	**0.56**	0.00	0.59	0.85	0.85	0.85	0.90	0.90	**0.90**	0.05	**0.87**	0.13
	F_1	0.88	**0.78**	0.43	**0.76**	0.22	**0.58**	0.00	0.74	0.87	**0.87**	0.72	**0.80**	0.74	**0.86**	0.10	**0.88**	0.23
all	WA	0.92	**0.82**	0.68	**0.84**	0.40	**0.70**	0.21	0.89	0.90	**0.90**	0.80	**0.91**	0.80	**0.84**	0.48	**0.86**	0.41

of performance in a classification task of 11 activities. This could be of great utilization e.g., in AR environments and applications, where such performance plays a significant and important role for the overall user experience and also may act as a means of assessing user engagement, e.g., when a visitor of a museum makes a phone call while interacting with an AR application, this should be an indicator of low engagement, while when she/he is reading in front of an AR screen, this should be an indicator of high engagement. Moreover, another important field of application would be an ambient assisted environment, where the goal is to detect activities of daily living (ADLs) [13].

Future research work may focus on several aspects of the problem of occlusion. Firstly, we would like to investigate cases such as temporally partial occlusion. Then we would like to investigate the use of other deep neural network architectures, such as generative adversarial networks (GANs). Moreover, we would like to perform experiments using full PKU-MMD and possibly other datasets. We would like to perform comparisons of the given approach to one that uses occluded samples for training the neural network that we have herein used for classification, without a regression step. Finally, we plan to perform real-life experiments within the AR environment of the Mon Repo project[5].

Acknowledgements This research was co-financed by the European Union and Greek national funds through the Competitiveness, Entrepreneurship and Innovation Operational Programme, under the Call "Special Actions "Aquaculture" - "Industrial materials" - "Open innovation in culture""; project title: "Strengthening User Experience & Cultural Innovation through Experiential Knowledge Enhancement with Enhanced Reality Technologies — MON REPO"; project code: $T6YB\Pi$ - 00303; MIS code: 5066856

References

1. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Zheng, X. (2016). TensorFlow: A System for Large-Scale Machine Learning. In 12th USENIX symposium on operating systems design and implementation (OSDI 16).
2. Angelini, F., Fu, Z., Long, Y., Shao, L., & Naqvi, S. M. (2019). 2d pose-based real-time human action recognition with occlusion-handling. IEEE Transactions on Multimedia, 22(6), 1433-1446.
3. Antoshchuk, S., Kovalenko, M., & Sieck, J. (2018). Gesture recognition-based human–computer interaction interface for multimedia applications. In Digitisation of Culture: Namibian and International Perspectives. Springer, Singapore.
4. F. Chollet, et al., Keras, `https://github.com/fchollet/keras` (2015).
5. Debes, C., Merentitis, A., Sukhanov, S., Niessen, M., Frangiadakis, N., & Bauer, A. (2016). Monitoring activities of daily living in smart homes: Understanding human behavior. IEEE Signal Processing Magazine, 33(2), 81-94.
6. Du, Y., Fu, Y., & Wang, L. (2015). Skeleton based action recognition with convolutional neural network. In 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR) (pp. 579-583). IEEE.

[5] `https://monrepo.online/`

7. Giannakos, I., Mathe, E., Spyrou, E., & Mylonas, P. (2021). A study on the Effect of Occlusion in Human Activity Recognition. In The 14th PErvasive Technologies Related to Assistive Environments Conference (pp. 473-482).
8. Gu, R., Wang, G., & Hwang, J. N. (2021). Exploring severe occlusion: Multi-person 3d pose estimation with gated convolution. In 2020 25th International Conference on Pattern Recognition (ICPR) (pp. 8243-8250). IEEE.
9. Hou, Y., Li, Z., Wang, P. & Li, W. (2016). Skeleton optical spectra-based action recognition using convolutional neural networks. IEEE Transactions on Circuits and Systems for Video Technology, 28(3), 807-811.
10. Iosifidis, A., Tefas, A., & Pitas, I. (2012). Multi-view human action recognition under occlusion based on fuzzy distances and neural networks. In Proceedings of the 20th European Signal Processing Conference (EUSIPCO). IEEE.
11. Ke, Q., An, S., Bennamoun, M., Sohel, F., & Boussaid, F. (2017). Skeletonnet: Mining deep part features for 3-d action recognition. IEEE signal processing letters, 24(6), 731-735.
12. Keogh, A., Dorn, J. F., Walsh, L., Calvo, F., & Caulfield, B. (2020). Comparing the usability and acceptability of wearable sensors among older irish adults in a real-world context: observational study. JMIR mHealth and uHealth, 8(4), e15704.
13. Lawton, M. P., & Brody, E. M. (1969). Assessment of older people: self-maintaining and instrumental activities of daily living. The gerontologist, 9(3 Part 1), 179-186.
14. Li, C., Hou, Y., Wang, P., & Li, W. (2017). Joint distance maps based action recognition with convolutional neural networks. IEEE Signal Processing Letters, 24(5), 624-628.
15. Liu, M., Liu, H., & Chen, C. (2017). Enhanced skeleton visualization for view invariant human action recognition. Pattern Recognition, 68, 346-362.
16. Liu, C., Hu, Y., Li, Y., Song, S., & Liu, J. (2017). PKU-MMD: A large scale benchmark for continuous multi-modal human action understanding. arXiv preprint arXiv:1703.07475.
17. Liu, T., Sun, J. J., Zhao, L., Zhao, J., Yuan, L., Wang, Y., ... & Adam, H. (2022). View-invariant, occlusion-robust probabilistic embedding for human pose. International Journal of Computer Vision, 130(1), 111-135.
18. Majumder, S., Mondal, T., & Deen, M. J. (2017). Wearable sensors for remote health monitoring. Sensors, 17(1), 130.
19. Papadakis, A., Mathe, E., Spyrou, E., & Mylonas, P. (2019). A geometric approach for cross-view human action recognition using deep learning. In 11th International Symposium on Image and Signal Processing and Analysis (ISPA). IEEE.
20. Ranasinghe, S., Al Machot, F., & Mayr, H. C. (2016). A review on applications of activity recognition systems with regard to performance and evaluation. International Journal of Distributed Sensor Networks, 12(8), 1550147716665520.
21. Vernikos, I., Mathe, E., Papadakis, A., Spyrou, E., & Mylonas, P. (2019, June). An image representation of skeletal data for action recognition using convolutional neural networks. In Proceedings of the 12th ACM International Conference on PErvasive Technologies Related to Assistive Environments (pp. 325-326).
22. Wang, P., Li, W., Li, C., & Hou, Y. (2018). Action recognition based on joint trajectory maps with convolutional neural networks. Knowledge-Based Systems, 158, 43-53.
23. Wang, P., Li, W., Ogunbona, P., Wan, J., & Escalera, S. (2018). RGB-D-based human motion recognition with deep learning: A survey. Computer Vision and Image Understanding, 171, 118-139.