Fusing Handcrafted and Contextual Features for Human Activity Recognition

Ioannis Vernikos*†, Eirini Mathe*‡, Evaggelos Spyrou*†, Alexandros Mitsou*, Theodore Giannakopoulos*
and Phivos Mylonas‡
*Institute of Informatics and Telecommunications, National Center for Scientific Research - "Demokritos," Athens, Greece
Email: {emathe,espyrou,tyianak}@iit.demokritos.gr
†Department of Computer Science and Telecommunications, University of Thessaly, Lamia, Greece
Email: {ivernikos,amitsou}@uth.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 activity that combines handcrafted features from 3D skeletal data and contextual features learnt by a trained deep Convolutional Neural Network (CNN). Our approach is based on the idea that contextual features, i.e., features learnt in a similar problem are able to provide a diverse representation, which, when combined with the handcrafted features is able to boost performance. To validate our idea, we train a CNN using a dataset for action recognition and use the output of the last fully-connected layer as a contextual feature representation. Then, a Support Vector Machine is trained upon an early fusion step of both representations. Experimental results prove that the proposed method significantly improves the recognition accuracy in an arm gesture recognition problem, compared to the use of handcrafted features only.**

***Index Terms*—Human Activity Recognition, Convolutional Neural Networks, Context-aware Deep Features**

I. Introduction

Human action recognition consists probably one of the most challenging problems in the field of computer vision. It is considered to be a sub-area of the human-centered activity recognition. According to Wang et al. [21], related recognition tasks fall into four categories: a) gesture; b) action; c) interaction; and d) group activity recognition. Gesture and action differ mainly in the amount of time required and also to the body parts involved. More specifically, a gesture is typically performed "instantly," i.e., it requires a small amount of time. Also, it usually involves a single body part. On the other hand, an action may involve more or even all body parts and requires a larger amount of time. However, they both involve one person, contrary to an interaction, which may involve two persons or a person and an object. Lastly, group activity involves more than one persons and is considered as a combination of some of the aforementioned categories.

At the early days of human action recognition, most approaches [1], [15] were based on the extraction of handcrafted features. The latter are typically algorithmic pipelines, manually designed to extract image properties using the information available within the image, in a deterministic way. Design takes place by keeping in mind that extracted features should be robust to variances such as illumination, viewpoint changes etc. Then, these features were used to train traditional machine learning algorithms such as support vector machines. Such approaches have several limitations including limiting robustness to viewpoint changes or significant drop of performance when trained for a large number of actions. Especially the first limitation is crucial, when dealing with real-life scenarios. Due to recent advances in the fields of hardware and more particularly in graphics processing units (GPUs), fast training of more complex network architectures has been enabled. Such architectures are typically referred to as "deep architectures" [5]. Their main advantage is that, contrary to traditional approaches, a prior handcrafted feature extraction step is omitted. Instead, features are "learnt" by the network, during the training process. Note, that efficient training of deep architectures requires a significantly large number of training examples.

Several human action recognition datasets have been proposed, comprising either of a small number of simple actions [15], such as *walking*, *running*, *hand clapping etc.*, or more realistic human actions e.g., *answer phone*, *get out of car*, *hand shake* [7]. More challenging datasets [6], [18] contain interactions with objects such as *playing cello*, *horse riding*, *swing baseball bat*, *fencing*. Finally, recent large scale datasets [9], [16] are comprised of large numbers of training video and depth sequences. An important problem when dealing with human action recognition tasks is the intense diversity between different datasets and benchmarks. More specifically, datasets show significant differences mainly in terms of size, visual data and classes.

Therefore, in this work, our goal is to experimentally demonstrate that the ability of deep architectures to learn patterns in large datasets may be used to complement handcrafted features and boost the performance of classification. More specifically, we use a Convolutional Neural Network (CNN) which has been pre-trained to classify contextual classes in a human action recognition problem as a visual feature extractor in the similar, yet different task of arm gesture recognition. We experimentally evaluate our approach, using the PKU-MMD dataset [9] which is a challenging large scale action

978-1-7281-3634-9/19/$31.00 © 2019 IEEE

recognition dataset to train the CNN and a dataset we had created in previous work [10] for arm gesture recognition.

The rest of this paper is organized as follows: Section II presents related work, focusing in deep learning approaches. Section III presents the proposed methodology. Experimental results are presented in Section IV, while conclusions are drawn in Section V, where plans for future work are also presented.

II. Related Work

During the last few years, several research works have experimented with fusion of handcrafted features with learnt features that are extracted using deep neural network architectures. Nanni et al. [11] proposed the use of a trained deep CNN as a feature extractor and mixed its output with handcrafted features. More specifically three substructures were proposed. Firstly they remapped the output of a CNN so as to solve a different problem than the one it was trained. Classification was performed with an SVM. Secondly, they fed an SVM with the output of the last dense layer of a CNN as a feature vector. Lastly, they fused the output of several layers and also fed an SVM. They also used several methods for extracting non-handcrafted features and a wide range of state-of-the-art algorithms for the handcrafted ones. They applied the method at several image classification problems. Wu et al. [22] proposed a model for person re-identification called "Feature Fusion Net." This model combines color histogram features and texture features with the last pooling layer of a CNN so as to adapt the weights of the CNN by taking into consideration the handcrafted features. Kashif et al. [4] proposed a technique that showed improved performance when handcrafted features were combined with raw data in order to be used as an input in a CNN model. They used the aforementioned features to detect tumor cells in histology images. Another approach, proposed by Egede et al. [2], combined handcrafted features with deep-learned features in a regression problem for automatic pain estimation. Nguyen et al. [12] used deep image features extracted by a CNN with local binary pattern handcrafted features in a presentation attack detection in face recognition. Finally, in previous work [3] we applied a methodology similar to the proposed one to solve a soundscape classification problem, combining statistical handcrafted short-term audio features with a CNN, trained on spectrogram representations of audio signals.

III. Proposed Methodology

A. Convolutional Neural Networks

The area of computer vision has significantly benefited from the use of deep learning architectures. The dominant deep architecture is undoubtedly the Convolutional Neural Network (CNN) [8]. The latter, resembles to traditional neural networks and has almost eliminated the need of extracted handcrafted features for the description of low-level visual properties. This is achieved by its key component, i.e., the *convolutional* layers. Their role is to learn a set of convolutional filters. The dimension of these filters is rather small (i.e., a few pixels). They are formed by grouping neurons in a rectangular grid and slide across the whole image during the forward pass of the algorithm. This way, the responses when a filter is centered at any pixel, produce a 2D activation map. Training takes place as with every other NN; a forward propagation of data and a backward propagation of error do take place to update weights. Ultimately, the filters learnt by the network will activate when they encounter certain types of visual features.

Pooling layers are usually placed between single or sets of serial or parallel convolutional layers. Their role is to progressively reduce the representation size. To achieve this, they take small rectangular blocks from the convolutional layer. By subsampling them, they end up to produce a single output from each block. This way, a step towards reducing the complexity of the network and controlling overfitting is performed. Finally, in *dense* layers (also known as "fully-connected" layers) each node is connected to all activations of their previous layer and their role is to perform classification based on the extracted features by the convolutional layers (which may have been subsampled by pooling layers). Within our approach, we also adopt the widely used *dropout* regularization technique [19]. Its goal is to reduce overfitting by preventing complex co-adaptations on training data, i.e., dependencies among neighboring neurons. To achieve this, several random neurons are ignored ("dropped-out") during training and also their weights are not updated.

B. Handcrafted Features

The set of handcrafted features used within this paper, has been proposed in our previous work [13] and is partially inspired by a set of features that have proposed by Sheng [17], who extended the features of Rubine [14] from the 2D to the 3D space. We use 3D skeletal data extracted by the Microsoft Kinect sensor, where the human skeleton is described by a structured graph that consists of a set of joints, which represent its main body parts (e.g., arms, legs, head, shoulders etc.). Note that joints are organized in a hierarchical structure, where a parent–child relationship is implied; e.g., the root is the *Hip Center*, its children are the *Spine*, the *Left Heap* and the *Right Heap* and so on. We extract features based on the spatial relation of joints to their parent and child joints, over time.

More specifically, for a given joint J we use its child and parent joints J_c and J_p, accordingly. Features are extracted from video sequences. Let $F_i,\ i = 1, 2, \dots N$ denote a given video frame and $\mathbf{v}_i^{(J)} = \left(v_{x,i}^{(J)}, v_{y,i}^{(J)}, v_{z,i}^{(J)}\right)$ a vector corresponding to the 3D coordinates of J at frame F_i. Also, let $\mathcal{V}^{(\mathcal{J})}$ denote the set of all $\mathbf{v}_i^{(J)}$, $B(\mathcal{V}^{(\mathcal{J})})$ the 3D bounding box of $\mathcal{V}^{(\mathcal{J})}$, by $a_{B(\mathcal{V}^{(\mathcal{J})})}$ and $b_{B(\mathcal{V}^{(\mathcal{J})})}$ the two different lengths of its sides. Extracted features are depicted in Table I.

C. Context-aware deep Feature Extraction using a CNN

In this work we use a visual representation of skeletal information, which has been proposed in our previous work [20], in order to create images so that they will be used as input to a CNN. We create pseudo-colored images which aim at capturing inter-joint distances during an action. For a

TABLE I

PROPOSED FEATURES, EXTRACTED FROM THE SKELETAL JOINTS. NOTE THAT EACH FEATURE IS CALCULATED FOR A GIVEN JOINT J AND MAY INVOLVE A DIFFERENT SUBSET OF FRAMES. FEATURES MARKED WITH $*$, ARE CALCULATED USING ONLY *HandLeft* AND/OR *HandRight*. ALSO, $B(\bullet)$ DENOTES A 3D BOUNDING BOX OF A SET OF VECTORS, WHILE $a_{B(\bullet)}, b_{B(\bullet)}$ THE LENGTHS OF THE SIDES OF $B(\bullet)$. FOR A GIVEN JOINT J, J_c, J_p DENOTE ITS CHILD AND PARENT NODE, RESPECTIVELY.

Feature name	Frames involved	Equation
Spatial angle	F_2, F_1	$\arccos \frac{\mathbf{v}_2^{(J)} \cdot \mathbf{v}_1^{(J)}}{\left\|\mathbf{v}_2^{(J)}\right\| \cdot \left\|\mathbf{v}_1^{(J)}\right\|}$
Spatial angle	F_N, F_{N-1}	$\arccos \frac{\mathbf{v}_N^{(J)} \cdot \mathbf{v}_{N-1}^{(J)}}{\left\|\mathbf{v}_N^{(J)}\right\| \cdot \left\|\mathbf{v}_{N-1}^{(J)}\right\|}$
Spatial angle	F_N, F_1	$\arccos \frac{\mathbf{v}_N^{(J)} \cdot \mathbf{v}_1^{(J)}}{\left\|\mathbf{v}_N^{(J)}\right\| \cdot \left\|\mathbf{v}_1^{(J)}\right\|}$
Total vector angle	$F_1, \ldots, F_N$	$\sum_{i=1}^{N} \arccos \left(\frac{\mathbf{v}_i^{(J)} \cdot \mathbf{v}_{i-1}^{(J)}}{\left\|\mathbf{v}_i^{(J)}\right\| \left\|\mathbf{v}_{i-1}^{(J)}\right\|} \right)$
Squared total vector angle	$F_1, \ldots, F_N$	$\sum_{i=1}^{n} \arccos \left(\frac{\mathbf{v}_i^{(J)} \cdot \mathbf{v}_{i-1}^{(J)}}{\left\|\mathbf{v}_i^{(J)}\right\| \left\|\mathbf{v}_{i-1}^{(J)}\right\|} \right)^2$
Total vector displacement	F_N, F_1	$\left\|\mathbf{v}_N^{(J)} - \mathbf{v}_1^{(J)}\right\|$
Total displacement	$F_1, \ldots, F_N$	$\sum_{i=1}^{n} \left\|\mathbf{v}_i^{(J)} - \mathbf{v}_{i-1}^{(J)}\right\|$
Maximum displacement	$F_1, \ldots, F_N$	$\max_{i=2,\ldots,N} \left(\left\|\mathbf{v}_i^{(J)} - \mathbf{v}_{i-1}^{(J)}\right\| \right)$
Bounding box diagonal length*	$F_1, \ldots, F_N$	$\sqrt{a^2_{B(\mathcal{V}^{(\mathcal{J})})} + b^2_{B(\mathcal{V}^{(\mathcal{J})})}}$
Bounding box angle*	$F_1, \ldots, F_N$	$\arctan \frac{b_{B(\mathcal{V}^{(\mathcal{J})})}}{a_{B(\mathcal{V}^{(\mathcal{J})})}}$
Initial angle	F_1	$\angle \mathbf{v}_1^{(J)} \mathbf{O} \mathbf{v}_1^{(J_p)}$ or $\angle \mathbf{v}_1^{(J)} \mathbf{O} \mathbf{v}_1^{(J_c)}$
Final angle	F_N	$\angle \mathbf{v}_N^{(J)} \mathbf{O} \mathbf{v}_N^{(J_p)}$ or $\angle \mathbf{v}_N^{(J)} \mathbf{O} \mathbf{v}_N^{(J_c)}$
Mean angle	$F_1, \ldots, F_N$	$\frac{1}{N} \sum_{i=1}^{N} \angle \mathbf{v}_i^{(J)} \mathbf{O} \mathbf{v}_i^{(J_p)}$ or $\frac{1}{N} \sum_{i=1}^{N} \angle \mathbf{v}_i^{(J)} \mathbf{O} \mathbf{v}_i^{(J_c)}$
Max angle	$F_1, \ldots, F_N$	$\max_{i=1}^{N} \angle \mathbf{v}_i^{(J)} \mathbf{O} \mathbf{v}_i^{(J_p)}$ or $\max_{i=1}^{N} \angle \mathbf{v}_i^{(J)} \mathbf{O} \mathbf{v}_i^{(J_c)}$

given joint J_i, $\mathbf{v}_i^{(J)} = \left(v_{x,i}^{(J)}, v_{y,i}^{(J)}, v_{z,i}^{(J)}\right)$ is the aforementioned vector capturing its 3D location at frame i, i.e., its trajectory. The 3D trajectories of skeletal joints are used as input for the construction of the representation. Each joint corresponds to a row of the constructed images, while, its x, y, z coordinates correspond to R, G, B color channels, respectively.

We should emphasize our assumption that each video segment contains *exactly* one action to be recognized. To address the problem of temporal variability between actions and between users which results to video segments of different lengths, a linear interpolation step is imposed, setting the duration of all videos equal to N frames. By using skeletal information collected by the Microsoft Kinect v2, i.e., consisting of 25 joints, we ended up with pseudocolored images having dimension $25 \times N \times 3$. Each is created as follows: we calculate coordinate differences between consecutive frames, thus if $v_{x,i}^{(j)}$ denotes the x-position of the j-th joint in the i-th frame. Let R denote the red channel of the color image. The value of $R(j,i)$ is calculated as: $R(j,i) = v_{x,i+1}^{(j)} - v_{x,i}^{(j)}$, $i = 1, \ldots, N$. Similarly, blue and green channels are constructed. As it is exhibited, the way these pseudo-colored images are formed, leads to preserving both the temporal and the spatial properties of the skeleton trajectories. Examples of pseudo-colored images are illustrated in Fig. 1.

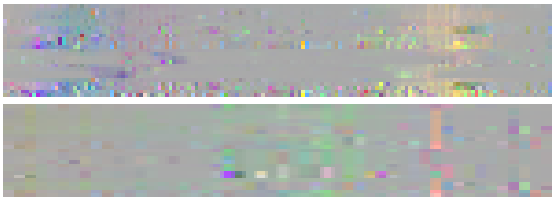

Fig. 1. Representative pseudo-colored images from actions of the PKU-MMD dataset [9] and by using all 25 skeletal joints. Top: *eat_meal_snack*, bottom: *falling*.

The architecture of our CNN is presented in detail in Fig. 2. First, we used a convolutional layer to filter the 25×149 input image with 16 kernels of size 3×3. Then, we continued with the first pooling layer, which uses "max-pooling" to perform 2×2 subsampling. A second and a third convolutional layer filters the 11×73 and 9×71 , resulting images with 32 kernels of size 3×3, respectively. A second pooling layer follows and uses "max-pooling" to perform 2×2 subsampling. Next, the fourth convolutional layer filters the 2×33 resulting image with 64 kernels of size 3×3. A third pooling layer uses "max-pooling" to perform 2×2 subsampling. Then, a flatten layer transforms the output image of size 1×16 of the third 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, however this layer is omitted when the CNN is used for feature extraction.

D. Human-Activity Recognition

A visual overview of the proposed approach is illustrated in Fig. 3. As it may be seen, we use two distinct feature extraction steps. The first consists of the handcrafted features that are extracted by 3D trajectories of skeletal joints as described in subsection III-B. Each activity is represented by a feature vector. We should emphasize that this description aims to represent the activity in a space that is discriminative with regards to the involved classes of the problem at hand. The second consists of the context-aware deep features that are extracted using a CNN, trained by using visual representations of 3D skeletal joint trajectories, as described in subsection III-C. The CNN has been trained in a different (yet semantically similar) problem and the output of its last fully connected layer is used as the extracted feature. Both feature representations are combined in an early fusion step and an SVM is used for classification upon a PCA step.

The motivation for our approach is that since, the CNN is trained using different context classes, it learns a complementary representation to the one provided by the handcrafted features. As it has been shown to a plethora of works [3], the combination of such diverse representations leads to improved recognition accuracy. In our approach, the CNN is trained at an action recognition problem, while the problem at hand is arm gesture recognition, i.e., both are problems that lie in the area of human activity recognition.

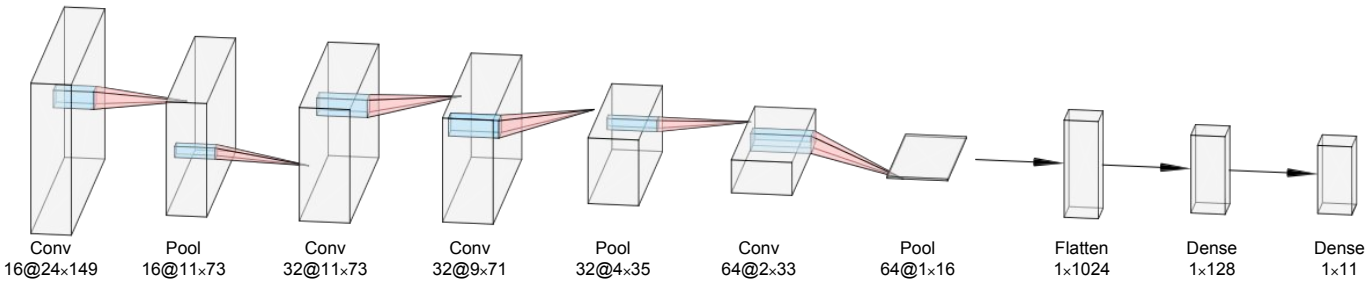

Fig. 2. The deep CNN that has been used for contextual feature extraction.

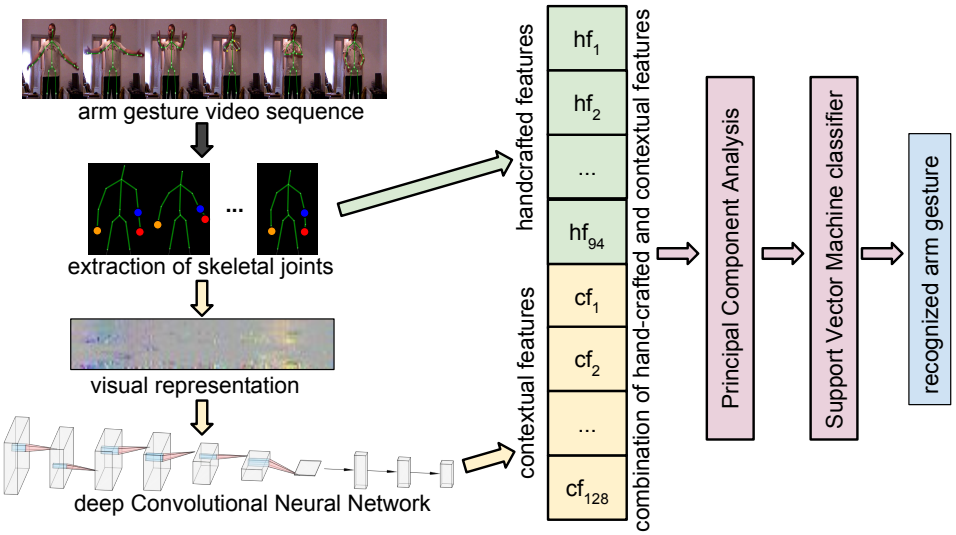

Fig. 3. A visual overview of the proposed approach.

IV. Experiments

A. Data sets

Over the last decade, a large number of benchmark datasets for human action recognition have been created and become publicly available. In order to train the CNN used throughout our experiments, we used a recent large-scale dataset, namely the PKU-MMD [9]. It contains approx. 20K action instances from 51 action categories, spanning into 5.4M video frames. The dataset is captured via the Kinect v2 sensor and provides multi-modality data sources, including skeleton, infrared radiation, depth and RGB sequences. Note that all data had been captured by the performance of 66 subjects under three camera views. The CNN was trained using all the available data. Apart from the aforementioned data set we also used a real-life dataset [10], constructed by us. More specifically this dataset consists of 200 arm gestures performed by 10 users and captured by one camera. Gestures are divided into the following classes: *clapping*, *hands raise*, *hands circle*, *swipe up*, *swipe down*,*swipe upright*, *swipe downright* In both cases, we have used the 3D skeletal data derived from the joints' motion.

B. Model Training

Since our focus was on solving an arm gesture classification problem, we extracted handcrafted features from hands, arms and wrists, i.e., $J \in \{$*ElbowLeft*, *ElbowRight*, *HandLeft*, *HandRight*, *WristLeft*, *WristRight*$\}$. From each of the 6 aforementioned joints we extracted the features of Table I, resulting to a feature vector representation of size 94. The CNN was trained for 11 classes of the PKU-MMD dataset, as in [20] which were: *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*. For classification, we have used a linear SVM. The dimension of the last fully connected layer was equal to 128, resulting to a combined representation of size 212. Upon PCA, we kept only the components that correspond to 95% of total variance, resulting to a feature vector of size 57, which was fed to the SVM.

Note that data from the PKU-MMD dataset have been collected using the Microsoft Kinect v2 camera, while data from our own dataset have been collected using the Microsoft v1 camera, where only a subset of the available joints have been used. Thus, in order to create the visual representation we have set to 0 all rows corresponding to non-available joints.

TABLE II
CLASSIFICATION ACCURACY FOR ALL FEATURE EXTRACTION METHODS.

Method	Accuracy
HF	0.87
CF	0.43
HF+CF	**0.93**

Also, we imposed a linear interpolation step, so as to make their dimension equal to the input of the trained CNN.

C. Results

For the experimental evaluation of the proposed approach, we conducted three series of experiments. First, we used only the handcrafted features (HF). Then, we used only the contextual features (CF) extracted by the CNN. Last, we used the combined features (HF+CF). In all cases, a linear SVM was trained upon PCA pre-processing of the features. Results are summarized in Table II. It can be seen that the HF+CF feature extraction approach clearly outperforms both other approaches. The combination of handcrafted and contextual features leads to a performance boosting of approx. 7%. Note, that the use of CF only showed by far the poorest performance.

V. CONCLUSIONS AND FUTURE WORK

In this paper we presented an approach for utilizing a Convolutional Neural Network that has been trained to classify human actions in an arm gesture classification task. To this goal, we have used a combination of handcrafted features with features learnt from the CNN. We demonstrated that this early fusion approach is able to provide a performance boosting, even though learnt features showed the poorest performance when used alone. This is due to the fact that the CNN is able to introduce a highly diverse representation, which is not captured by the handcrafted features. In our opinion, the main contribution of this work is the experimental proof that the transfer of contextual knowledge using a CNN is able to improve classification accuracy of handcrafted features. Future work will focus on experimenting with more powerful classifiers and also to the extensive evaluation of the proposed approach on several publicly available datasets. Among our goals is also the application into a real-like or even real-life ambient assistive living environment.

ACKNOWLEDGMENT

We acknowledge support of this work by the project SYNTELESIS "Innovative Technologies and Applications based on the Internet of Things (IoT) and the Cloud Computing" (MIS 5002521) which is implemented under the "Action for the Strategic Development on the Research and Technological Sector", funded by the Operational Programme "Competitiveness, Entrepreneurship and Innovation" (NSRF 2014-2020) and co-financed by Greece and the European Union (European Regional Development Fund). This work has also been supported by the project VISOR "Virtual coachIng Services for OldeR adults" which is implemented under the "1st HFRI Call for Scholarships for PhD Candidates," funded by the General Secretariat for Research and Technology and the Hellenic Foundation for Research and Innovation (HFRI).

REFERENCES

[1] Dollr, P., Rabaud, V., Cottrell, G., & Belongie, S. (2005, October). Behavior recognition via sparse spatio-temporal features. In 2005 IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (pp. 65-72). IEEE.

[2] Egede, J., Valstar, M., & Martinez, B. (2017, May). Fusing deep learned and hand-crafted features of appearance, shape, and dynamics for automatic pain estimation. In 2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017) (pp. 689-696). IEEE.

[3] Giannakopoulos, Th., Spyrou, E., & Perantonis, S. (2019). Recognition of urban sound events using deep context-aware feature extractors and handcrafted features. In Proc. of Mining Humanistic Data Workshop (MHDW), located at the Int'l Conference on Artificial Intelligence Applications and Innovations (AIAI).

[4] Kashif, M. N., Raza, S. E. A., Sirinukunwattana, K., Arif, M., & Rajpoot, N. (2016, April). Handcrafted features with convolutional neural networks for detection of tumor cells in histology images. In 2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI) (pp. 1029-1032). IEEE.

[5] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105).

[6] Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., & Serre, T. (2011, November). HMDB: a large video database for human motion recognition. In 2011 International Conference on Computer Vision (pp. 2556-2563). IEEE.

[7] Laptev, I., Marszalek, M., Schmid, C., & Rozenfeld, B. (2008, June). Learning realistic human actions from movies. In 2008 IEEE Conference on Computer Vision and Pattern Recognition (pp. 1-8). IEEE.

[8] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.

[9] 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.

[10] Mathe, E., Mitsou, A., Spyrou, E., & Mylonas, Ph. (2018). Arm Gesture Recognition using a Convolutional Neural Network. In Proc. of Intl Workshop on Semantic and Social Media Adaptation and Personalization (SMAP).

[11] Nanni, L., Ghidoni, S., & Brahnam, S. (2017). Handcrafted vs. non-handcrafted features for computer vision classification. Pattern Recognition, 71, 158-172.

[12] Nguyen, D. T., Pham, T. D., Baek, N. R., & Park, K. R. (2018). Combining deep and handcrafted image features for presentation attack detection in face recognition systems using visible-light camera sensors. Sensors, 18(3), 699.

[13] Paraskevopoulos, G., Spyrou, E., & Sgouropoulos, D. (2016, May). A real-time approach for gesture recognition using the Kinect sensor. In Proceedings of the 9th Hellenic Conference on Artificial Intelligence (p. 31). ACM.

[14] Rubine, D. (1991). Specifying gestures by example (Vol. 25, No. 4, pp. 329-337). ACM.

[15] Schuldt, C., Laptev, I., & Caputo, B. (2004, August). Recognizing Human Actions: A Local SVM Approach. In Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 3-Volume 03 (pp. 32-36). IEEE Computer Society.

[16] Shahroudy, A., Liu, J., Ng, T. T., & Wang, G. (2016). NTU RGB+D: A large scale dataset for 3D human activity analysis. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1010-1019).

[17] Sheng, J. (2003). A study of adaboost in 3d gesture recognition. Department of Computer Science, University of Toronto.

[18] Soomro, K., Zamir, A. R., & Shah, M. (2012). UCF101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402.

[19] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research, 15(1), 1929-1958.

[20] Vernikos, I., Mathe, E., Papadakis, A., Spyrou, E., & Mylonas, Ph. (2019). An Image Representation of Skeletal Data for Action Recognition using Convolutional Neural Networks. In Proc. of PErvasive Technologies Related to Assistive Environments (PETRA) Conference.

[21] 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.

[22] Wu, S., Chen, Y. C., Li, X., Wu, A. C., You, J. J., Zheng, W. S. (2016, March). An enhanced deep feature representation for person re-identification. In 2016 IEEE winter conference on applications of computer vision (WACV) (pp. 1-8). IEEE.