Neural Networks for Digital Media Analysis and Description

Anastasios Tefas, Alexandros Iosifidis, and Ioannis Pitas

Department of Informatics
Aristotle University of Thessaloniki,
54124 Thessaloniki, Greece
{tefas,aiosif,pitas}@aiia.csd.auth.gr

Abstract. In this paper a short overview on recent research efforts for digital media analysis and description using neural networks is given. Neural networks are very powerful in analyzing, representing and classifying digital media content through various architectures and learning algorithms. Both unsupervised and supervised algorithms can be used for digital media feature extraction. Digital media representation can be done either in a synaptic level or at the output level. The specific problem that is used as a case study for digital media analysis is the human-centered video analysis for activity and identity recognition. Several neural network topologies, such as self organizing maps, independent subspace analysis, multi-layer perceptrons, extreme learning machines and deep learning architectures are presented and results on human activity recognition are reported.

Keywords: Neural Networks, Digital Media analysis, Activity recognition, Deep learning

1 Introduction

Recent advances in technological equipment, like digital cameras, smart-phones, etc., have led to an increase of the available digital media, e.g., videos, captured every day. Moreover, the amount of data captured for professional media production (e.g., movies, special effects, etc) has dramatically increased and diversified using multiple sensors (e.g., 3D scanners, multi-view cameras, very high quality images, motion capture, etc), justifying the digital media analysis as a big data analysis problem. As expected, most of these data are acquired in order to describe human presence and activity and are exploited either for monitoring (visual surveillance and security) or for personal use and entertainment. Basic problems in human centered media analysis are face recognition [1], facial expression recognition [2] and human activity recognition [3]. According to YouTube statistics[1], 100 hours of video are uploaded by the users every minute. Such a data growth, as well as the importance of visual information in many applications, has necessitated the creation of methods capable of automatic processing

[1] http://www.youtube.com/yt/press/statistics.html

and decision making when necessary. This is why a large amount of research has been devoted in the analysis and description of digital media in the last two decades.

Artificial Neural Networks (NN), played an important role towards the direction of developing techniques which can be used for digital media analysis, representation and classification. Beyond the methods that will described in more detail in the rest of the paper we should note recent developments in the area of deep learning neural networks [4]. Deep learning architectures have been successfully used for image retrieval [5], natural language processing [6], large scale media analysis problems [7], and feature learning [8]. Among the architectures used in deep learning are Deep and Restricted Boltzmann Machines, Auto-encoders, Convolutional neural networks, Recurrent neural networks, etc.

As it will be described in the following sections, NN-based techniques can be exploited in order to properly describe digital media, extract semantic information that is useful for analysis and make decisions, e.g., decide in which category (class) a video belongs to. We will discuss these steps in the context of two important applications, i.e., human action recognition and person identification from videos.

2 Problem Statement

Let us assume that a video database $\mathcal{U}$ contains N_T videos depicting human actions. Let us also assume that these videos have been manually annotated, i.e., they have been classified according to the performed action and/or the ID of the persons appearing in each of them. Thus, each video i depicting a human action, called action video hereafter, is accompanied by an action class and a person ID label, α_i and h_i, respectively. We would like to employ these videos, as well as the corresponding labels α_i, h_i, $i = 1, \ldots, N_T$, in order to train an algorithm that will be able to automatically perform action recognition and/or person identification, i.e., to classify a new, unknown, video to an action and/or a person ID class appearing in an action class set $\mathcal{A}$ and/or a person ID set $\mathcal{P}$, respectively.

The above described process is, usually, performed in two steps, as illustrated in Figure 1. The first one exploits an appropriate action/person description in order to determine a convenient video representation. The second one exploits the obtained video representation in order to determine action class and person ID models that will be used for the classification of a new (test) video.

3 Video Representation

Video representations aiming at action recognition and person identification exploit either global human body information, in terms of binary silhouettes corresponding to the human body video frame locations, or shape and motion information appearing in local video locations. In the first case, action videos are

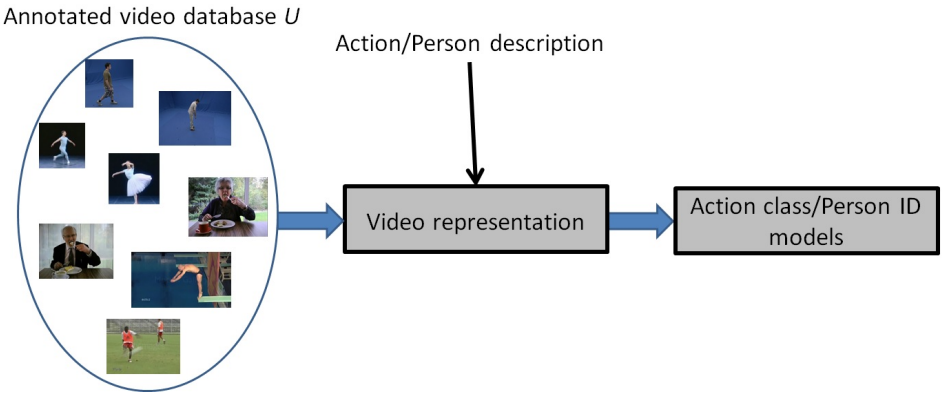

Fig. 1: *Processing steps frequently used by action recognition and person identification methods.*

usually described as sequences of successive human body poses, as illustrated in Figure 2.

Fig. 2: *Action 'walk' described by using binary human body silhouettes.*

By using such an action video representation, it has been shown that, in the case of everyday actions, both action recognition and person identification can be simultaneously performed [9, 10]. The adopted action video representation involves the determination of D human body pose prototypes $\mathbf{v}_d$, $d = 1, \ldots, D$. This is achieved by training a self-organizing NN (Self-Organizing Map) exploiting the human body poses $\mathbf{p}_{ij}$ of all the training action videos appearing in $\mathcal{U}$. Its training procedure involves two phases:

- **Competition:** For each of the training human body pose $\mathbf{p}_{ij}$, its Euclidean distance from every SOM neuron $\mathbf{v}_d$ is calculated. Wining neuron is the one providing the smallest distance, i.e.:

$$d^* = arg \min_d \| \mathbf{p}_{ij} - \mathbf{v}_d \|_2. \quad (1)$$

- **Co-operation:** Each SOM neuron is adapted with respect to its lateral distance from the winning neuron h_d, i.e.:

$$\mathbf{v}_d(n+1) = \mathbf{v}_d(n) + \eta(n) h_d(n) (\mathbf{p}_i - \mathbf{v}_d(n)), \quad (2)$$

where $h_d(n)$ is a function of the lateral distance $r_{d*,d}$ between the winning neuron d^* and neuron d, $\eta(n)$ is an adaptation rate parameter and n refers

to the algorithms training iteration. Typical choice of $h_d(n)$ is the Gaussian function $h_d(n) = \exp\left(-\frac{r^2_{d^*,d}}{2\sigma^2(n)}\right)$.

An example SOM obtained by using action videos depicting eight persons performing multiple instances of five actions is illustrated in Figure 3. As can be seen, the SOM neurons correspond to representative human body poses during action execution captured from different view angles. Furthermore, it can be observed that each SOM neuron captures human body shape properties of different persons in $\mathcal{U}$. For example, it can be seen that neuron $\{6, G\}$ depicts a man waving his hand and from a frontal view, while neuron $\{10, I\}$ depicts a woman jumping from a side view.

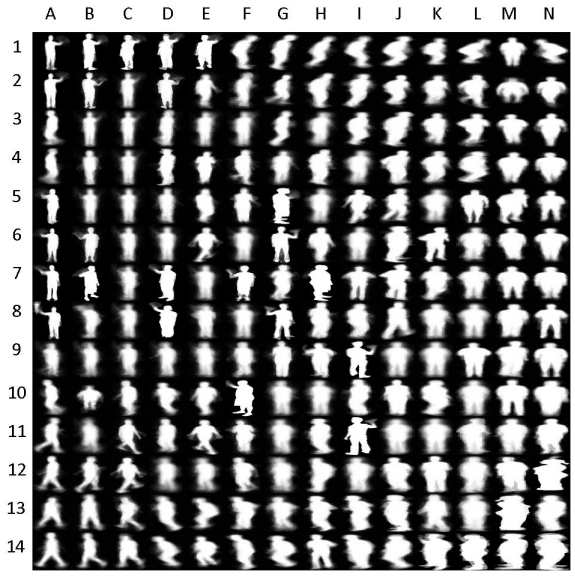

Fig. 3: *A* 14×14 *SOM obtained by using action videos depicting eight persons performing multiple instances of actions walk, run, jump in place, jump forward and wave one hand.*

After SOM determination, each human body pose $\mathbf{p}_{ij}$ is mapped to the so-called membership vector $\mathbf{u}_{ij} = [u_{ij1}, \ldots, u_{ijD}]^T$ encoding the fuzzy similarity between $\mathbf{p}_{ij}$ with all the human body prototypes $\mathbf{v}_d$, according to a fuzzification parameter $m > 1$:

$$u_{ijd} = (\|\mathbf{p}_{ij} - \mathbf{v}_d\|_2)^{\frac{2}{m-1}} . \tag{3}$$

Finally, each action video i, consisting of N_i video frames, is represented by the so-called action vectors $\mathbf{s}_i \in \mathbb{R}^D$:

$$\mathbf{s}_i = \frac{1}{N_i} \sum_{j=1}^{N_i} \frac{\mathbf{u}_{ij}}{\|\mathbf{u}_{ij}\|_1}. \tag{4}$$

Regarding video representations exploiting local video information, a popular choice is to use overlapping 3D video blocks, where the third dimension refers to time, in order to learn representative 3D blocks describing local shape and motion information. Independent Subspace Analysis (ISA) has been proposed to this end in [11]. An ISA network can be considered to be a neural network consisting of two layers, with square and square-root nonlinearities in the first and second layer respectively. Let us denote by $\mathbf{x}_t$ a given training input pattern. The activation function of each second layer unit is given by:

$$p_i(\mathbf{x}_t; \mathbf{U}, \mathbf{V}) = \left(\sum_{k=1}^{m} V_{ik} \left(\sum_{j=1}^{m} U_{kj} x_{tj} \right)^2 \right)^{\frac{1}{2}}. \tag{5}$$

Parameters $\mathbf{U}$ are learned through finding sparse representations in the second layer by solving:

$$\begin{aligned} &\underset{\mathbf{W}}{\text{maximize}} \quad \sum_{t=1}^{T} \sum_{i=1}^{m} p_i(\mathbf{x}_t; \mathbf{U}, \mathbf{V}) \\ &subject\ to: \mathbf{U}\mathbf{U}^T = \mathbf{I} \end{aligned} \tag{6}$$

$\mathbf{U}$, $\mathbf{V}$ in (5), (6) are matrices containing the weights connecting the input data to the first layer units and the units of the first layer to the second layer units, respectively. In order to reduce the computational cost of the training process, PCA is performed in order to reduce the dimensionality of the input data. In order to learn high-level concepts, a convolution and stacking technique is employed. According to this, small input patches are employed in order to train an ISA network. Subsequently, the learned network is convolved with a larger region of the input video. The convolved responses are fed to another ISA network. Example filters learned from video frames depicting traditional Greek dances are illustrated in Figure 4 [12].

After training the two-layered ISA network by following the above described procedure, action videos are represented by using the Bag of Features (BoFs) model. That is, the responses of the ISA network corresponding to all the training action videos are clustered in order to determine a set of K representative ISA features, which form the so-called codebook. Finally, each action video i is represented by the corresponding histogram $\mathbf{s}_i \in \mathbb{R}^K$ calculated by employing the obtained codebook.

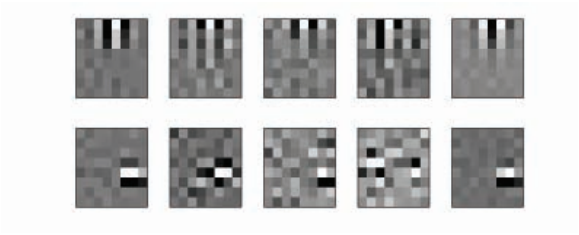

Fig. 4: *Filters learned by the ISA algorithm when trained on video frames depicting traditional Greek dances.*

4 Video Classification

By following either of the above described procedures, each training action video is represented by the corresponding action vector $\mathbf{s}_i$. That is, action video classification has been transformed to the corresponding action vector classification task. Feedforward Neural Networks have been widely adopted to this end, due to their ability to universally approximate any continuous target functions in any compact subset $\mathcal{X}$ of the Euclidean space $\mathbb{R}^D$ [13, 14].

A Multi-layer Perceptron (MLP), i.e., a Feedforward NN consisting of D input (equal to action vectors dimensionality) and N_A output neurons (equal to the number of classes forming the classification problem), has been employed to this end in [9] for human action recognition. In the training phase, training action vectors $\mathbf{s}_i$ accompanied by the corresponding action class labels α_i are used in order to define MLP weights $\mathbf{W}$ by using the Backpropagation algorithm [13]. Action class labels α_i are employed in order to set the corresponding network target vectors $\mathbf{t}_i$. For each of the action vectors, MLP response $\mathbf{o}_i = [o_{i1}, \ldots, o_{iN_P}]^T$ is calculated by:

$$o_{ik} = f_{sigmoid}\left(\mathbf{w}_k^T \mathbf{s}_i\right), \tag{7}$$

where $\mathbf{w}_k$ is a vector containing the MLP weights corresponding to output k. The training procedure is performed in an on-line form, i.e., adjustments of the MLP weights are performed for each training action vector. After the feed of a training action vector $\mathbf{s}_i$ and the calculation of the MLP response $\mathbf{o}_i$ the weight connecting neurons i and j follows the update rule:

$$\Delta\mathbf{W}_{ji}(n+1) = c\Delta\mathbf{W}_{ji}(n) + \eta\delta_j(n)\psi_i(n), \tag{8}$$

where $\delta_j(n)$ is the local gradient for neuron j, ψ_i is the output of neuron i, η is the learning rate parameter and c is a positive number, called momentum constant. This procedure is applied until the Mean Square Error (MSE) between the network output vectors $\mathbf{o}_i$ and the network target vectors $\mathbf{t}_i$ falls under an acceptable error rate ϵ.

Single-hidden Layer Feedforward Neural (SLFN) networks have been adopted for action recognition and person identification in [10, 16–18]. A SLFN network consists of D input, L hidden and N_A output neurons, as illustrated in Figure

5. In order to perform fast and efficient network training, the Extreme Learning Machine (ELM) algorithm [15] has been employed in [16].

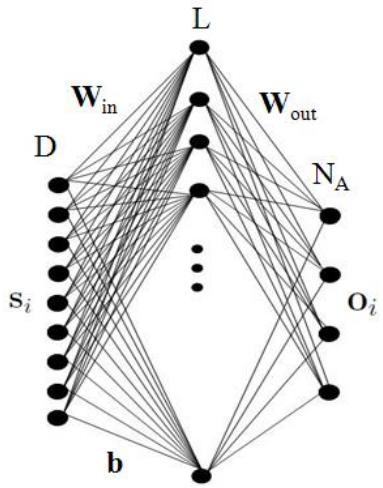

Fig. 5: *SLFN network topology.*

In ELM, the network's input weights $\mathbf{W}_{in}$ and the hidden layer bias values $\mathbf{b}$ are randomly assigned, while the output weights $\mathbf{W}_{out}$ are analytically calculated. Let $\mathbf{v}_j$ denote the j-th column of $\mathbf{W}_{in}$ and $\mathbf{w}_k$ the k-th column of $\mathbf{W}_{out}$. For a given activation function $\Phi()$, the output $\mathbf{o}_i = [o_1, \ldots, o_{N_A}]^T$ of the ELM network corresponding to training action vector $\mathbf{s}_i$ is calculated by:

$$o_{ik} = \sum_{j=1}^{L} \mathbf{w}_k^T \, \Phi(\mathbf{v}_j, b_j, \mathbf{s}_i), \; k = 1, ..., N_A. \tag{9}$$

By storing the hidden layer neurons outputs in a matrix $\mathbf{\Phi}$, i.e.:

$$\mathbf{\Phi} = \begin{bmatrix} \Phi(\mathbf{v}_1, b_1, \mathbf{s}_1) & \cdots & \Phi(\mathbf{v}_1, b_1, \mathbf{s}_{N_T}) \\ \cdots & \ddots & \cdots \\ \Phi(\mathbf{v}_L, b_L, \mathbf{s}_1) & \cdots & \Phi(\mathbf{v}_L, b_L, \mathbf{s}_{N_T}) \end{bmatrix}, \tag{10}$$

Equation (9) can be written in a matrix form as $\mathbf{O} = \mathbf{W}_{out}^T \mathbf{\Phi}$. Finally, by assuming that the network's predicted outputs $\mathbf{O}$ are equal to the network's desired outputs, i.e., $\mathbf{o}_i = \mathbf{t}_i$, and using linear activation function for the output neurons, $\mathbf{W}_{out}$ can be analytically calculated by $\mathbf{W}_{out} = \mathbf{\Phi}^\dagger \, \mathbf{T}^T$, where $\mathbf{\Phi}^\dagger = \left(\mathbf{\Phi}\mathbf{\Phi}^T\right)^{-1} \mathbf{\Phi}$ is the Moore-Penrose generalized pseudo-inverse of $\mathbf{\Phi}^T$ and $\mathbf{T} = [\mathbf{t}_1, \ldots, \mathbf{t}_{N_T}]$ is a matrix containing the network's target vectors.

A regularized version of the ELM algorithm has, also, been used in [10, 17]. According to this, the network output weights $\mathbf{W}_{out}$ are calculated by solving

the following optimization problem:

$$\textbf{Minimize:} \quad L_P = \frac{1}{2}\|\mathbf{W}_{out}^T\|_F + \frac{c}{2}\sum_{i=1}^{N_V}\|\boldsymbol{\xi}_i\|_2^2 \tag{11}$$

$$\textbf{Subject to:} \ \boldsymbol{\phi}_i^T\mathbf{W}_{out} = \mathbf{t}_i^T - \boldsymbol{\xi}_i^T, \ i = 1, ..., N_T, \tag{12}$$

where $\boldsymbol{\xi}_i$ is the training error vector corresponding to action vector $\mathbf{s}_i$, $\boldsymbol{\phi}_i$ denotes the i-th column of $\boldsymbol{\Phi}$, i.e., the $\mathbf{s}_i$ representation in the ELM space, and c is a parameter denoting the importance of the training error in the optimization problem. By substituting the condition (12) in (11) and solving for $\frac{\vartheta L_P}{\vartheta \mathbf{W}_{out}} = 0$, $\mathbf{W}_{out}$ can be obtained by:

$$\mathbf{W}_{out} = \left(\boldsymbol{\Phi}\boldsymbol{\Phi}^T + \frac{1}{c}\mathbf{I}\right)^{-1}\boldsymbol{\Phi}\mathbf{T}^T, \tag{13}$$

or

$$\mathbf{W}_{out} = \boldsymbol{\Phi}\left(\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \frac{1}{c}\mathbf{I}\right)^{-1}\mathbf{T}^T. \tag{14}$$

where $\mathbf{I}$ is the identity matrix.

Exploiting the fact that the ELM algorithm can be considered to be a non-linear data mapping process to a high dimensional feature space followed by linear projection and classification, the Minimum Class Variance ELM (MCVELM) algorithm has been proposed in [18] for action recognition. MCVELM tries to simultaneously minimize the network output weights norm and within-class variance of the network outputs. The network output weights $\mathbf{W}_{out}$ are calculated by solving the following optimization problem:

$$\textbf{Minimize:} \quad L_P = \frac{1}{2}\|\mathbf{S}_w^{1/2}\mathbf{W}_{out}^T\|_F + \frac{c}{2}\sum_{i=1}^{N_V}\|\boldsymbol{\xi}_i\|_2^2 \tag{15}$$

$$\textbf{Subject to:} \ \boldsymbol{\phi}_i^T\mathbf{W}_{out} = \mathbf{t}_i^T - \boldsymbol{\xi}_i^T, \ i = 1, ..., N_T, \tag{16}$$

and the network output weights are given by:

$$\mathbf{W}_{out} = \left(\boldsymbol{\Phi}\boldsymbol{\Phi}^T + \frac{1}{c}\mathbf{S}_w\right)^{-1}\boldsymbol{\Phi}\mathbf{T}^T. \tag{17}$$

$\mathbf{S}_w$ in (15), (17) is the within-class scatter matrix of the network hidden layer outputs, i.e., the representation of $\mathbf{s}_i$ in the so-called ELM space. Two cases have been exploited. In the case of unimodal action classes, the within-class scatter matrix is of the form:

$$\mathbf{S}_w = \sum_{j=1}^{N_A}\sum_{i=1}^{N_V}\frac{\beta_{ij}}{N_j}(\boldsymbol{\phi}_i - \boldsymbol{\mu}_j)(\boldsymbol{\phi}_i - \boldsymbol{\mu}_j)^T. \tag{18}$$

In (18), β_{ij} is an index denoting if training action vector $\mathbf{s}_i$ belongs to action class j, i.e., $\beta_{ij} = 1$, if $c_i = j$ and $\beta_{ij} = 0$ otherwise, and $N_j = \sum_{i=1}^{N_V}\beta_{ij}$ is the number

of training action vectors belonging to action class j. $\boldsymbol{\mu}_j = \frac{1}{N_j}\sum_{i=1}^{N_V}\beta_{ij}\boldsymbol{\phi}_i$ is the mean vector of class j in the ELM space.

In the case of multi-modal action classes, the within-class scatter matrix is of the form:

$$\mathbf{S}_{w,CDA} = \sum_{j=1}^{N_A}\sum_{k=1}^{b_j}\sum_{i=1}^{N_V}\frac{\beta_{ijk}(\boldsymbol{\phi}_i - \boldsymbol{\mu}_{jk})(\boldsymbol{\phi}_i - \boldsymbol{\mu}_{jk})^T}{N_{jk}}. \tag{19}$$

Here, it is assumed that class j consists of b_j clusters, containing N_{jk}, $j = 1, ..., N_A$, $k = 1, ..., b_j$ action vectors each. β_{ijk} is an index denoting if action vector $\mathbf{s}_i$ belongs to the k-th cluster of action class j and $\boldsymbol{\mu}_{jk} = \frac{1}{N_{jk}}\sum_{i=1}^{N_V}\beta_{ijk}\boldsymbol{\phi}_i$ denotes the mean vector of the k-th cluster of class j in the ELM space.

By exploiting the fast and efficient ELM algorithm for SLFN network training, a dynamic classification schemes have been proposed for human action recognition in [19]. It consists of two iteratively repeated steps. In the first step, a non-linear mapping process for both the training action vectors and the test sample under consideration is determined by training a SLFN network. In the second step, test sample-specific training action vectors selection is performed by exploiting the obtained network outputs corresponding to both the training action vectors and the test sample under consideration. SLFN-based data mapping and training action vectors selection are performed in multiple levels, which are determined by the test-sample under consideration. At each level, by exploiting only the more similar to the test sample training action vectors, the dynamic classification scheme focuses the classification problem on the classes that should be able to discriminate. A block diagram of the above described dynamic classification scheme is illustrated in Figure 6.

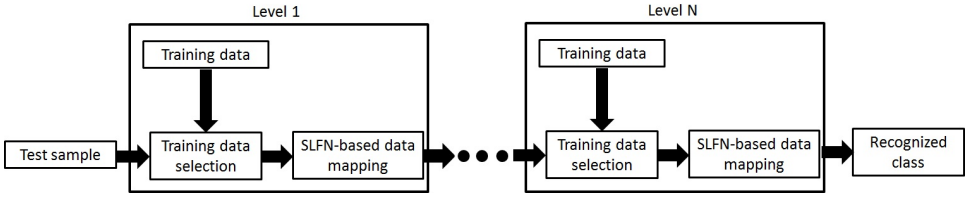

Fig. 6: *SLFN-based dynamic classification scheme.*

Considering the fact that after performing multiple data selections for a level $l > 1$ the cardinality of the training action vectors set that will be used for SLFN network training will be very small compared to the dimensionality of the ELM space, the regularized version of ELM algorithm (13) has been employed in [19]. In order to avoid the determination of the number of hidden layer neurons at each level l, the regularized version of ELM algorithm (14) has been employed.

In this case, the network output vector corresponding to $\mathbf{s}_i$ is obtained by:

$$\mathbf{o}_i = \mathbf{W}_{out}^T \phi_i = \mathbf{T} \left(\mathbf{\Omega} + \frac{1}{c} \mathbf{I} \right)^{-1} \mathbf{K}_i, \tag{20}$$

where $\mathbf{K}_i = \mathbf{\Phi}^T \phi_i$, $\mathbf{\Omega} = \mathbf{\Phi}^T \mathbf{\Phi}$ are the kernel matrices corresponding to $\mathbf{s}_i$ and the entire SLFN training set, respectively. Thus, in this case the ELM space dimensionality is inherently determined by exploiting the kernel trick [21] and needs not be defined in advance.

Experimental results in real video data using all the previously presented methods can be found in the corresponding references. The results indicate that various neural network topologies can be used for solving difficult tasks, such as video analysis and semantic information extraction in digital media. The results obtained indicate that neural networks are among the state-of-the-art solutions for digital media analysis, representation and classification.

5 Conclusion

In this paper a survey on neural networks based methods for digital media analysis and description is presented. Neural Networks are very powerful both on analysing/representing and on classifying digital media content. The semantic information of focus is the human activity and identity and the problem used as case-study is activity recognition and person identification from video data. The presented approaches are generic and can be easily used for other semantic concepts, especially those that involve human presence in digital media content.

Acknowledgment

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement number 316564 (IMPART). This publication reflects only the authors views. The European Union is not liable for any use that may be made of the information contained therein.

References

1. Kyperountas, M. and Tefas, A. and Pitas, I.: Dynamic training using multistage clustering for face recognition. Pattern Recognition, 894–905 (2008)
2. Kyperountas, M. and Tefas, A. and Pitas, I.: Salient feature and reliable classifier selection for facial expression classification. Pattern Recognition, 972–986 (2010)
3. Gkalelis, N. and Tefas, A. and Pitas, I.: Combining fuzzy vector quantization with linear discriminant analysis for continuous human movement recognition. IEEE Transactions on Circuits and Systems for Video Technology, 1511–1521 (2008)
4. Bengio, Y. and Courville, A.C. and Vincent, P.: Unsupervised Feature Learning and Deep Learning: A Review and New Perspectives. Arxiv, (2012)

5. Krizhevsky, A. and Sutskever, I. and Hinton, G.E.: ImageNet Classification with Deep Convolutional Neural Networks. Neural Information Processing Systems, (2012)
6. Bordes, A. and Glorot, X. and Weston, J. and Bengio, Y.: Joint Learning of Words and Meaning Representations for Open-Text Semantic Parsing. In: Proceedings of the 15th International Conference on Artificial Intelligence and Statistics (AISTATS), (2012)
7. Le, Q.V. and Ranzato, M. and Monga, R. and Devin, M. and Chen, K. and Corrado, G.S. anf Dean, J. and Ng, A.Y.: Building High-level Features Using Large Scale Unsupervised Learning. ICML (2012)
8. Goodfellow, I. and Courville, A. and Bengio, Y.: Large-Scale Feature Learning With Spike-and-Slab Sparse Coding. ICML (2012)
9. Iosifidis, A. and Tefas, A. and Pitas, I.: View-invariant action recognition based on Artificial Neural Networks. IEEE Transactions on Neural Networks and Learning Systems. vol. 23, no. 3, pp. 412–424 (2012)
10. Iosifidis, A. and Tefas, A. and Pitas, I.: Person Identification from Actions based on Artificial Neural Networks. Symposium Series on Computational Intelligence (SSCI): Computational Intelligence in Biometrics and Identity Management, Singapore, (2013)
11. Le, Q.V. and Zou, W.Y. and Yeung, S.Y. and Ng, A.Y.: Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, pp. 3361–3368, IEEE Press, Colorado (2011)
12. Kapsouras, I. and Karanikolos, S. and Nikolaidis, N. and Tefas, A.: Feature Comparison and Feature Fusion for Traditional Dances Recognition. In: 14th Engineering Applications of Neural Networks Conference, Halkidiki, (2013)
13. Haykin, S. : Neural Networks and Learning Machines. Upper Saddle River, New Jersey, (2008)
14. Huang, G.B. and Chen, L. and Siew, C.: Universal approximation using incremental constructive feedforward networks with random hidden nodes. IEEE Transactions on Neural Networks, vol. 17, no. 4, pp. 879–892, (2006)
15. Huang, G.B. and Zhou, H. and Ding, X. and Zhang, R.: Extreme Learning Machine for Regressiona and Multiclass Classification. IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics, vol. 42, no. 2, pp. 513–529, (2012)
16. Minhas, R. and Baradarani, S. and Seifzadeh, S. and Wu, Q.J..: Human action recognition using extreme learning machine based on visual vocabularies,. Neurocomputing, vol. 73, no. 10–12, pp. 1906–1917, (2010)
17. Iosifidis, A. and Tefas, A. and Pitas, I.: Multi-view Human Action Recognition under Occlusion based on Fuzzy Distances and Neural Networks. European Signal Processing Conference, pp. 1129–1133, (2012)
18. Iosifidis, A. and Tefas, A. and Pitas, I.: Minimum Class Variance Extreme Learning Machine for Human Action Recognition. IEEE Transactions on Circuits and Systems for Video Technology, accepted, (2013)
19. Iosifidis, A. and Tefas, A. and Pitas, I.: Dynamic action recognition based on dynemes and Extreme Learning Machine. Pattern Recognition Letters, accepted, (2013)
20. Iosifidis, A. and Tefas, A. and Pitas, I.: Dynamic Action Classification Based on Iterative Data Selection and Feedfforward Neural Networks. European Signal Processing Conference, accepted, (2013)
21. Scholkopf, B. and Smola, A.J..: Learning with kernels: Support vector machines, regularization, optimization, and beyond. MIT Press, (2001)