DYNAMIC ACTION CLASSIFICATION BASED ON ITERATIVE DATA SELECTION AND FEEDFORWARD NEURAL NETWORKS

Alexandros Iosifidis, Anastasios Tefas and Ioannis Pitas

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

ABSTRACT

In this paper we present a dynamic classification scheme involving Single-hidden Layer Feedforward Neural (SLFN) network-based non-linear data mapping and test sample-specific labeled data selection in multiple levels. The number of levels is dynamically determined by the test sample under consideration, while the use of Extreme Learning Machine (ELM) algorithm for SLFN network training leads to fast operation. The proposed dynamic classification scheme has been applied to human action recognition by employing the Bag of Visual Words (BoVW)-based action video representation providing enhanced classification performance compared to the static classification approach.

***Index Terms*—** Dynamic classification, Data selection, Feedforward Neural network, Extreme Learning Machine

1. INTRODUCTION

Classification methods can be categorized depending on the way they utilize the available labeled data in static and dynamic methods. Static classification methods employ all the available labeled data and the corresponding class labels in order to train a classifier that will be used in order to classify any (unknown) test sample. Dynamic classification methods involve a model adaptation process based either on the training set structure, or on the test sample to be classified.

By exploiting the information provided by the test sample under consideration, it has been shown that dynamic classification schemes can provide enhanced classification performance, compared to the static ones. A dynamic classification scheme exploiting sparsity constraints has been proposed in [1]. A given test sample is involved in a L1-minimization-based class-independent regression process by using an overcomplete dictionary formed by all the available labeled data. Multiple reconstruction samples are, subsequently, produced by exploiting the reconstruction weights corresponding to each class independently and the test sample under consideration is classified based on the minimum reconstruction error classification rule. The Dynamic Committee Machine (DCM) has been proposed in [2]. DCM employs five state-of-the-art classifiers in order to determine five classification results for a given test sample. The obtained classification results are, finally, fused by using test sample-specific combination weights. A dynamic classification scheme has been proposed in [3] for human action recognition. The classification process involved person identification and action classification based on a classifier trained by using labeled samples belonging the recognized person. A dynamic classification scheme involving training data clustering and Linear Discriminant Analysis (LDA)-based data projection in multiple levels is proposed in [4]. The procedure used in order to determine an appropriate training set for LDA-based data projection and classification is intuitive and effective. However, the LDA-based classification approach in this setting sets the assumption of linearly separable classes, which is not met in several classification problems where non-linear classification models are more appropriate. In order to overcome this assumption, a non-linear data mapping process has been employed in [5].

In this paper we present a dynamic classification scheme consisting of two iteratively repeated processing steps. In the first step, a non-linear mapping process for both the training data and the test sample under consideration is determined by training a Single-hidden Layer Feedforward Neural (SLFN) network. In the second step, test sample-specific training data selection is performed by exploiting the obtained network outputs corresponding to both the training data and test sample under consideration. SLFN-based data mapping and training data 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 data, the proposed dynamic classification scheme focuses the classification problem on the classes that should be able to discriminate. The adopted labeled data selection process is intuitive and effective, while the use of the Extreme Learning Machine (ELM) algorithm [6] for SLFN network training results to fast network training, leading to fast and effective dynamic classification.

The paper is structured as follows. Section 2 describes the proposed dynamic classification scheme. Section 3 illustrates experimental results conducted in order to evaluate its performance. Finally, conclusions are drawn in Section 4.

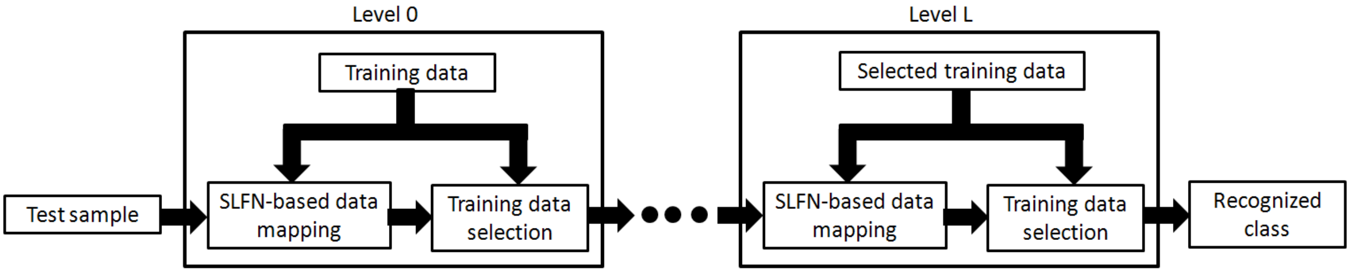

Fig. 1. *The proposed dynamic classification scheme.*

2. PROPOSED METHOD

The proposed dynamic classification scheme consists of two processing steps. The first one involves non-linear data mapping to a new feature space determined by the outputs of an SLFN network. The second step performs test sample-based training data selection. These two steps are performed multiple times (levels) in order to determine the class label of a given test sample. The procedure followed by the proposed dynamic classification method is illustrated in Figure 1. In the following, we describe the two above mentioned processing steps and the proposed dynamic classification method.

2.1. SLFN-based Data Mapping

Let $\mathcal{Z} = \{\mathbf{z}_i\}_{i=1}^{|\mathcal{Z}|}$ be a vector set formed by the $|\mathcal{Z}|$ training (labeled) vectors. We employ $\mathcal{Z}$ in order to determine a new feature space resulted by a non-linear mapping process determined by training an SLFN network. The SLFN network consists of N input (equal to the dimensionality of $\mathbf{z}_i$), Q hidden and C output (equal to the number of classes appearing in $\mathcal{Z}$) neurons. The network target vectors $\mathbf{t}_i$, $i = 1, \dots, |\mathcal{Z}|$ are set to $t_{ij} = 1$ for vectors belonging to class j and $t_{ij} = -1$ otherwise.

In order to achieve fast operation, we employ the ELM algorithm for SLFN network training [6]. In ELM, the network input weights $\mathbf{W}_{in} \in \mathbb{R}^{N \times Q}$ and bias values $\mathbf{b} \in \mathbb{R}^{Q}$ are randomly chosen, while the output weights $\mathbf{W}_{out} \in \mathbb{R}^{Q \times C}$ are analytically calculated. By storing the hidden layer neurons outputs g_{ij}, $i = 1, \dots, |\hat{\mathcal{Z}}|$, $j = 1, \dots, Q$ in a matrix $\mathbf{G}$, i.e.,:

$$\mathbf{G} = \begin{bmatrix} G(\mathbf{w}_1, b_1, \hat{\mathbf{z}}_1) & \cdots & G(\mathbf{w}_1, b_1, \hat{\mathbf{z}}_{|\hat{\mathcal{Z}}|}) \\ \dots & \ddots & \dots \\ G(\mathbf{w}_Q, b_Q, \hat{\mathbf{z}}_1) & \cdots & G(\mathbf{w}_Q, b_Q, \hat{\mathbf{z}}_{|\hat{\mathcal{Z}}|}) \end{bmatrix}, \quad (1)$$

and using linear activation function for the network output layer, the network's output vector corresponding to training vector $\mathbf{z}_i$ is given by $\mathbf{o}_i = \mathbf{W}_{out}^T \mathbf{g}_i$, where $\mathbf{g}_i$ is the i-th column of $\mathbf{G}$ and denotes the network hidden layer output for $\mathbf{z}_i$. In (1), $\mathbf{w}_j$, b_j denote the j-th column of $\mathbf{W}_{in}$ and the j-th element of $\mathbf{b}$. The network's outputs corresponding to all the labeled vectors forming $\mathcal{Z}$ can be written in a matrix form as $\mathbf{O} = \mathbf{W}_{out}^T \mathbf{G}$. Finally, by assuming that the network output vectors $\mathbf{o}_i$ are equal to the network target vectors $\mathbf{t}_i$, $\mathbf{W}_{out}$ can be calculated by:

$$\mathbf{W}_{out} = \left(\mathbf{G}\mathbf{G}^T\right)^{-1} \mathbf{G}\mathbf{T}^T, \quad (2)$$

where $\mathbf{T}[\mathbf{t}_1 \dots \mathbf{t}_{|\mathcal{Z}|}$ is a matrix containing the network target vectors.

By observing (2) it can be seen that this equation can be used for $\mathbf{W}_{out}$ calculation only in the cases where the matrix $\hat{\mathbf{G}} = \mathbf{G}\mathbf{G}^T$ is non-singular, i.e., in the cases where $|\hat{\mathcal{Z}}| > Q$. However, considering the fact that after performing multiple data selections for a level $l > 1$ the cardinality of $\hat{\mathcal{Z}}$ will be very small compared to the dimensionality of the network hidden layer output vectors, we employ a regularized version of (2) proposed in [7], i.e.:

$$\mathbf{W}_{out} = \mathbf{G}\left(\mathbf{G}^T\mathbf{G} + \frac{1}{c}\mathbf{I}\right)^{-1} \mathbf{T}^T. \quad (3)$$

The value of regularization parameter c is determined by following a grid search strategy, as it will be described in the experimental section. By using (3), the network output vector corresponding to $\mathbf{z}_i$ is obtained by:

$$\mathbf{o}_i = \mathbf{W}_{out}^T \mathbf{g}_i = \mathbf{T}\left(\mathbf{\Omega} + \frac{1}{c}\mathbf{I}\right)^{-1} \mathbf{K}_i, \quad (4)$$

where $\mathbf{K}_i = \mathbf{G}^T \mathbf{g}_i$, $\mathbf{\Omega} = \mathbf{G}^T \mathbf{G}$ are the kernel matrices corresponding to $\mathbf{z}_i$ and the entire training set, respectively [7]. We employ (4) in all our experiments, since in this case the dimensionality of the network hidden layer is inherently determined by exploiting the kernel trick [8] and needs not to be provided by the user.

After training the SLFN network, the training vectors $\mathbf{z}_i$ are introduced to the network in order to determine the vector set $\mathcal{O} = \{\mathbf{o}_i\}_{i=1}^{|\mathcal{Z}|}$, where $\mathbf{o}_i$ is the network output for $\mathbf{z}_i$. The test sample $\mathbf{z}_{test}$ is, also, introduced to the trained SLFN network in order to obtain its response $\mathbf{o}_{test}$.

2.2. Dynamic Data Selection

Let $\mathcal{O} = \{\mathbf{o}_i\}_{i=1}^{|\mathcal{Z}|}$ and $\mathbf{o}_{test}$ denote the network outputs for the training vectors $\mathbf{z}_i$, $i = 1, \dots, |\mathcal{Z}|$ and the test sample under consideration $\mathbf{z}_{test}$, respectively. In order to determine the training vectors that provide the M most similar to the test sample $\mathbf{z}_{test}$ network outputs, we calculate the Euclidean distances between $\mathbf{o}_{test}$ and $\mathbf{o}_i$:

$$d_i = \|\mathbf{o}_i - \mathbf{o}_{test}\|. \tag{5}$$

The obtained distances $d_i,\ k = 1,...,|\mathcal{Z}|$ are sorted in an ascending order and the training vectors that provide the M most similar to the test sample $\mathbf{z}_{test}$ network outputs are those providing the M smallest distance values. In our experiments M is automatically determined by using $M = m|\mathcal{Z}|$, where $m < 1$.

Alternatively, one may cluster the network output vectors $\mathbf{o}_i$ in K groups, e.g., by applying K-Means algorithm, and select the training vectors belonging to the group where $\mathbf{o}_{test}$ belongs to, similar to [4, 5]. This approach has the advantage that the number of selected training data M is dynamically determined by the test vector under consideration. However, clustering $\mathcal{O}$ is computationally demanding compared to the adopted approach. Furthermore, in the cases where the test sample network output vector $\mathbf{o}_{test}$ is far from the corresponding group center, clustering would not result to optimal training data selection for classification.

2.3. Dynamic Classification Scheme

Let $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^{|\mathcal{X}|}$ be a vector set containing training vectors $\mathbf{x}_i \in \mathbb{R}^N$ which are followed by class labels c_i appearing in a class label set $\mathcal{C}$. Let $\mathbf{x}_{test} \in \mathbb{R}^N$ be a vector representing the test sample under consideration. $\mathcal{X}$ is used in order to train a SLFN network by following the procedure described in subsection 2.1. After training the SLFN network, both $\tilde{\mathcal{X}}_1$ and $\mathbf{x}_{test}$ are introduced to the trained network in order to obtain $\mathcal{O}_1$ and $\mathbf{o}_{test,1}$, respectively. Here, we have introduced an index denoting the level of the proposed dynamic classification scheme. After obtaining $\mathcal{O}_1$ and $\mathbf{o}_{test,1}$, the training vectors that provide the M most similar to the test sample network outputs are determined by following the procedure described in subsection 2.2. These vectors are selected in order to form the algorithm's second level training data set $\mathcal{X}_2$.

In the general case, after obtaining the l-th SLFN network outputs $\mathcal{O}_l$ and $\mathbf{o}_{test,l}$, the l-th level training vectors providing the $M = m|\mathcal{X}_l|$ most similar to the test sample network outputs are determined by following the procedure described in subsection 2.2. The obtained vectors are used to form the $l+1$-th level training set $\tilde{\mathcal{X}}_{l+1}$, which is used in order to train an SLFN network by following the procedure described in subsection 2.1.

The above described process is performed for multiple levels L until the labeled vectors forming the SLFN network training set belong to one class only. That is, the number of mapping levels L depends on the test sample under consideration. In the cases where the classification problem involves well distinguished classes we expect the number of mapping levels L to be low. In the cases of overlapping classes multiple mapping levels will be performed in order to obtain the final classification result.

3. EXPERIMENTS

We have applied the proposed dynamic classification scheme on three publicly available human action recognition databases. A brief description of the adopted databases and the experimental protocols adopted in each case are given in subsection 3.1. We have employed Harris3D Space Time Interest Point (STIP) detector [9] in order to determine STIP locations on action videos. Histogram of Oriented Gradient (HOG) and Histogram of Optical Flow (HOF) descriptors [10] have been calculated on STIP action video locations and have been concatenated in order to provide the obtained descriptor. HOG/HOF descriptors have been normalized to have unit L2 norm. The obtained normalized HOG/HOF descriptors have been employed in order to represent action videos by following the Bag of Visual Words (BoVW)-based approach. In our experiments, codebooks are constructed by applying K-Means clustering. We set the number of codebook vectors equal to $N = 4000$, since this value has been shown to empirically give good results for a wide range of datasets. To limit complexity, we cluster a subset of 10^5 randomly selected HOG/HOF vectors. To increase precision, we initialize K-Means 8 times and keep the codebook providing the lowest intra-cluster variance. HOG/HOF vectors are assigned to the closest codebook vector using Euclidean distance. The resulting histograms of HOG/HOF occurrences are used in order to represent action videos.

For SLFN network training, we use (4) and χ^2 kernel:

$$K(\mathbf{z}_i, \mathbf{z}_j) = exp\left(-\frac{1}{D}\sum_{n=1}^{N}\frac{(z_{in} - z_{jn})^2}{2(z_{in} + z_{jn})}\right), \tag{6}$$

where D is the mean value of distances between all training data $\mathbf{z}_i$. The training data selection parameter m and the optimal ELM regularization parameter value have been determined by following a grid search strategy using the values $m = 0.1\mu,\ \mu = 1,\dots,5$ and $c = 2^r,\ r = -20,\dots,20$.

3.1. Adopted Action Databases

3.1.1. KTH action database

The KTH action database consists of 600 action videos depicting 25 persons, each performing six actions [11]. The actions appearing in the database are: 'walking', 'jogging', 'running', 'boxing', 'hand waving' and 'hand clapping'. Four different scenarios have been recorded: outdoors (s1), outdoors with scale variation (s2), outdoors with different clothes (s3) and indoors (s4), as illustrated Figure 2. The persons are free to change motion speed and direction between different action realizations. The most widely adopted experimental

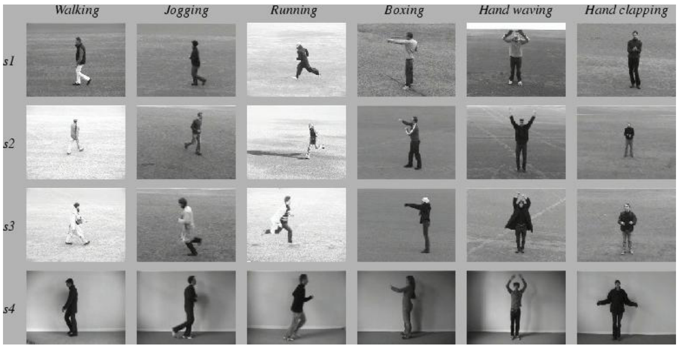

Fig. 2. *Video frames of the KTH action database for the four different scenarios.*

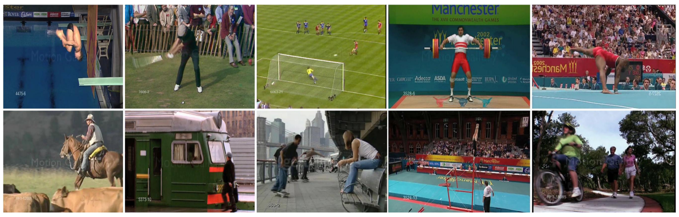

Fig. 3. *Video frames of the UCF sports action database.*

setting on this data set is based on a split (16 training and 9 test persons) that has been used in [11].

3.1.2. UCF sports action database

The UCF sports action database consists of 150 action videos depicting actions collected from ten sports which are typically featured on broadcast television channels, such as the BBC and ESPN [12]. The actions appearing in the database are: 'diving', 'golf swinging', 'kicking', 'lifting', 'horse riding', 'running', 'skating', 'bench swinging', 'swinging' and 'walking'. The videos were obtained from a wide range of stock footage websites including BBC Motion gallery and GettyImages. The collection represents a natural pool of actions featured in a wide range of scenes and viewpoints. The Leave-One-Video-Out cross-validation procedure is used by most action recognition methods evaluating their performance on this data set. Example video frames are illustrated in Figure 3.

3.1.3. Hollywood2 action database

The Hollywood2 action database consists of 1707 action videos depicting actions collected from 69 different Hollywood movies [10]. The actions appearing in the database are: 'answering the phone', 'driving car', 'eating', 'fighting', 'getting out of the car', 'hand shaking', 'hugging', 'kissing', 'running', 'sitting down', 'sitting up' and 'standing up'. The most widely adopted experimental setting on this data set is based on a split (823 training and 884 test action videos) that is provided by the database. Example video frames are illustrated in Figure 4.

Fig. 4. *Video frames of the Hollywood2 action database.*

3.2. Experimental Results

Table 1 illustrates the classification rates obtained by applying the proposed dynamic classification scheme on all the three databases. In this table we, also, provide the action classification rates obtained by applying action classification following the static classification approach, i.e., by applying only the first level of the proposed dynamic classification scheme, referred to as Static ELM. As can be seen, the adoption of a dynamic classification approach enhances the action classification performance in all the three cases, providing up to 3.7% improvement on the obtained action classification rate.

Table 1. Comparison results on the KTH, UCF sports and Hollywood2 databases for the static and dynamic classification schemes.

Method	KTH	UCF sports	Hollywood2
Static ELM	88.89%	78%	47.38%
Proposed Scheme	**92.59%**	**80.66%**	**50.11%**

For comparison reasons, we have implemented the dynamic classification schemes proposed in [4, 5, 1] and applied them to the three action databases employing the adopted action video representation. Comparison results between the four dynamic classification schemes are provided in Table 2. Finally, we provide comparison results between the proposed dynamic classification scheme for human action recognition and other methods proposed in the literature employing Harris3D STIP detector and HOG/HOF-based action video representation in Table 3. As can be seen in these Tables, the proposed dynamic classification scheme clearly outperforms all the competing dynamic and static classification schemes in all the three databases.

Table 2. Comparison results on the KTH, UCF sports and Hollywood2 databases for different dynamic classification schemes.

Method	KTH	UCF sports	Hollywood2
Method [4]	90.74.%	78.66%	47.51%
Method [5]	92.13%	79.33%	47.62%
Method [1]	91.66%	79.33%	48.75%
Proposed Scheme	**92.59%**	**80.66%**	**50.11%**

Table 3. Comparison results on the KTH, UCF sports and Hollywood2 databases for methods employing Harris3D STIP detector and HOG/HOF-based action video representation.

Method	KTH	UCF sports	Hollywood2
Method [10]	–	–	32.4%
Method [13]	91.8%	78.1%	47.6%
Proposed Scheme	**92.59%**	**80.66%**	**50.11%**

4. CONCLUSION

In this paper we presented a dynamic classification scheme involving two iteratively repeated processing steps. The first one determines a non-linear data mapping to a feature space determined by the outputs of a SLFN network trained by using training (labeled) data. The second one, performs test sample-specific training data selection for optimal SLFN network training. The method has been tested on three publicly available action recognition databases providing enhanced classification performance compared to the static classification approach.

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.

5. REFERENCES

[1] J. Wright, A.Y. Yang, A. Ganesh, S.S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, 2009.

[2] H.M. Tang, M.R. Lyu, and I. King, "Face recognition committee machines: dynamic vs. static structures," in *International Conference on Image Analysis and Processing*, 2003, pp. 121–126.

[3] A. Iosifidis, A. Tefas, and I. Pitas, "Person specific activity recognition using fuzzy learning and discriminant analysis," *European Signal Processing Conference*, pp. 1974–1978, 2011.

[4] M. Kyperountas, A. Tefas, and I. Pitas, "Dynamic training using multistage clustering for face recognition," *Pattern Recognition*, vol. 41, no. 3, pp. 894–905, 2008.

[5] A. Iosifidis, A. Tefas, and I. Pitas, "Dynamic action recognition based on dynemes and extreme learning machine," *Pattern Recognition Letters*, p. in press, 2013.

[6] G.B. Huang, Q.Y. Zhu, and C.K. Siew, "Extreme learning machine: a new learning scheme of feedforward neural networks," in *International Joint Conference on Neural Networks*. IEEE, 2004, vol. 2, pp. 985–990.

[7] G.B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, no. 2, pp. 513–529, 2012.

[8] B. Scholkopf and A.J. Smola, "Learning with kernels: Support vector machines, regularization, optimization, and beyond"," *MIT Press*, 2001.

[9] I. Laptev and T. Lindeberg, "Space-time interest points," *IEEE International Conference on Computer Vision*, pp. 432–439, 2008.

[10] M. Marszalek, I. Laptev, and C. Schmid, "Actions in context," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2929–2936, 2009.

[11] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local svm approach," *International Conference on Pattern Recognition*, vol. 3, pp. 32–36, 2004.

[12] M.D. Rodriguez and M. Shah, "Action mach a spatio-temporal maximum average correlation height filter for action recognition," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2008.

[13] H. Wang, M.M. Ullah, A. Klaser, I. Laptev, and C. Schmid, "Evaluation of local spatio-temporal features for action recognition," *British Machine Vision Conference*, vol. 42, no. 1, pp. 1–11, 2009.