Adaptive On-Line Neural Network Retraining for Real Life Multimodal Emotion Recognition

Spiros Ioannou[1], Loic Kessous[2], George Caridakis[1], Kostas Karpouzis[1],
Vered Aharonson[2], and Stefanos Kollias[1]

[1] School of Electrical and Computer Engineering, National Technical University of Athens,
Politechnioupoli, Zographou, Greece
{sivann, gcari, kkarpou,stefanos}@image.ece.ntua.gr
[2] Tel Aviv Academic College of Engineering
218 Bnei Efraim St. 69107, Tel Aviv, Israel
kessous@post.tau.ac.il, vered@nexsig.com

Abstract. Emotions play a major role in human-to-human communication enabling people to express themselves beyond the verbal domain. In recent years, important advances have been made in unimodal speech and video emotion analysis where facial expression information and prosodic audio features are treated independently. The need however to combine the two modalities in a naturalistic context, where adaptation to specific human characteristics and expressivity is required, and where single modalities alone cannot provide satisfactory evidence, is clear. Appropriate neural network classifiers are proposed for multimodal emotion analysis in this paper, in an adaptive framework, which is able to activate retraining of each modality, whenever deterioration of the respective performance is detected. Results are presented based on the IST HUMAINE NoE naturalistic database; both facial expression information and prosodic audio features are extracted from the same data and feature-based emotion analysis is performed through the proposed adaptive neural network methodology.

1 Introduction

Humans interact with each other in a multimodal manner to convey general messages; emphasis on certain parts of a message is given via speech and display of emotions by visual, vocal, and other physiological means, even instinctively. In the last decade much effort has been directed towards multimodal user interfaces that emulate human to human communication with the goal of enabling computer interfaces with means of natural, expressive and thus more intuitive ways of interaction.

Typical examples of human communication vehicles include auditory channels that carry speech or paralinguistic intonation and visual channels that convey facial expressions or body movements. The related senses of sight and hearing are examples of modalities. Everyday face-to-face communication utilizes many and diverse channels and modalities, increasing the flexibility of a communication scheme. In these situations, failure of one channel is usually recovered by another channel; this kind of behaviour should actually be considered as a model requirement for robust, natural and efficient multimodal HCI [12]. Therefore, the introduction of an emotion analysis

S. Kollias et al. (Eds.): ICANN 2006, Part I, LNCS 4131, pp. 81 – 92, 2006.

system that can analyse intonation and visual cues, to help infer the likely emotional state of a specific user in real life environments, can enhance the affective nature [13] of MMI applications. Adaptive artificial neural network classifiers are proposed in this paper, which can treat both sound and vision cues for emotion analysis, can evaluate their single or multi-modal performance and can adapt their knowledge, through on-line retraining, to real life changing environments.

Probably the most important issue when designing and training artificial neural networks in real life applications is network generalization. Many significant results have been derived during the last few years regarding generalization of neural networks when tested outside their training environment. Examples include algorithms for adaptive creation of the network architecture during training, such as pruning or constructive techniques, modular and hierarchical networks, or theoretical aspects of network generalization, such as the VC dimension. Specific results and mathematical formulations regarding error bounds and overtraining issues have been obtained when considering cases with known probability distributions of the data. Despite, however, the achievements obtained, most real life applications do not obey some specific probability distribution and may significantly differ from one case to another mainly due to changes of their environment. That is why straightforward application of trained networks, to data outside the training set, is not always adequate for solving image recognition, classification or detection problems, as is the case with (multimodal) emotion analysis. Instead, it would be desirable to have a mechanism, which would provide the network with the capability to automatically test its performance and be automatically retrained when its performance is not acceptable. The retraining algorithm should update the network weights taking into account both the former network knowledge and the knowledge extracted from the current input data.

This paper presents an approach for improving the performance of neural networks when handling real life multimodal emotion analysis, based on an automatic decision mechanism, which determines when network retraining should take place, and a retraining - nonlinear programming - algorithm.

Section 2 formulates the retraining problem under investigation. Section 3 presents the retraining technique, while section 4 presents the decision mechanism for activating retraining. Section 5 presents the multimodal emotion recognition problem and the application of the afore-mentioned technologies to the problem, while section 6 summarizes and provides conclusions on the capabilities of the proposed approach.

2 Formulation of the Problem

Let us assume that we seek to classify, to one of, say, p available emotion classes ω, each input vector $\underline{x}_i$ containing the features extracted by one or more input modalities. A neural network produces a p-dimensional output vector $\underline{y}(\underline{x}_i)$

$$\underline{y}(\underline{x}_i) = \left[p^i_{\omega_1} \, p^i_{\omega_2} \cdots p^i_{\omega_p} \right]^T \tag{1}$$

where $p^i_{\omega_j}$ denotes the probability that the ith input belongs to the jth class.

Let us first consider that a neural network has been initially trained to perform the previously described classification task using a specific training set, say, $S_b = \{(\underline{x}'_1, \underline{d}'_1), \cdots, (\underline{x}'_{m_b}, \underline{d}'_{m_b})\}$, where vectors $\underline{x}'_i$ and $\underline{d}'_i$ with $i = 1,2,\cdots,m_b$ denote the ith input training vector and the corresponding desired output vector consisting of p elements. Let $\underline{y}(\underline{x}_i)$ denote the network output when applied to the ith input outside the training set, corresponding to a new user, or to a change of the environmental conditions; new network weights should be estimated in such cases.

Let $\underline{w}_b$ include all weights of the network before retraining, and $\underline{w}_a$ the new weight vector which is obtained after retraining. A training set S_c is assumed to be extracted from the current operational situation composed of, (one or more), say, m_c inputs; $S_c = \{(\underline{x}_1, \underline{d}_1), \cdots, (\underline{x}_{m_c}, \underline{d}_{m_c})\}$ where $\underline{x}_i$ and $\underline{d}_i$ with $i = 1,2,\cdots,m_c$ similarly correspond to the ith input and desired output retraining data. The retraining algorithm that is activated, whenever such a need is detected, computes the new network weights $\underline{w}_a$, minimizing the following error criterion with respect to weights,

$$E_a = E_{c,a} + \eta E_{f,a}$$

$$E_{c,a} = \frac{1}{2}\sum_{i=1}^{m_c} \left\| \underline{z}_a(\underline{x}_i) - \underline{d}_i \right\|_2, \qquad E_{f,a} = \frac{1}{2}\sum_{i=1}^{m_b} \left\| \underline{z}_a(\underline{x}'_i) - \underline{d}'_i \right\|_2 \tag{2}$$

where $E_{c,a}$ is the error performed over training set S_c ("current" knowledge), $E_{f,a}$ the corresponding error over training set S_b ("former" knowledge); $\underline{z}_a(\underline{x}_i)$ and $\underline{z}_a(\underline{x}'_i)$ are the outputs of the retrained network, corresponding to input vectors $\underline{x}_i$ and $\underline{x}'_i$ respectively, of the network consisting of weights $\underline{w}_a$. Similarly $\underline{z}_b(\underline{x}_i)$ would represent the output of the network, consisting of weights $\underline{w}_b$, when accepting vector $\underline{x}_i$ at its input; when retraining the network for the first time $\underline{z}_b(\underline{x}_i)$ is identical to $\underline{y}(\underline{x}_i)$. Parameter η is a weighting factor accounting for the significance of the current training set compared to the former one and $\|\cdot\|_2$ denotes the L_2 -norm.

3 The Retraining Approach

The goal of the training procedure is to minimize (2) and estimate the new network weights $\underline{w}_a$, i.e., $\mathbf{W}_a^0$ and $\underline{w}_a^1$ respectively. The adopted algorithm has been proposed by the authors in [2]. Let us first assume that a small perturbation of the network weights (before retraining) $\underline{w}_b$ is enough to achieve good classification performance. Then,

$$\mathbf{W}_a^0 = \mathbf{W}_b^0 + \Delta\mathbf{W}^0, \quad \underline{w}_a^1 = \underline{w}_b^1 + \Delta\underline{w}^1 \tag{3}$$

where $\Delta\mathbf{W}^0$ and $\Delta\underline{w}^1$ are small increments. This assumption leads to an analytical and tractable solution for estimating $\underline{w}_a$, since it permits linearization of the non-linear activation function of the neuron, using a first order Taylor series expansion.

Equation (2) indicates that the new network weights are estimated taking into account both the current and the previous network knowledge. To stress, however, the importance of current training data in (2), one can replace the first term by the constraint that the actual network outputs are equal to the desired ones, that is

$$z_a(\underline{x}_i) = d_i \quad i = 1,\ldots,m_c, \quad \text{for all data in } S_c \tag{4}$$

Equation (4) indicates that the first term of (2), corresponding to error $E_{c,a}$, takes values close to zero, after estimating the new network weights.

Through linearization, solution of (4) with respect to the weight increments is equivalent to a set of linear equations

$$\underline{c} = \mathbf{A} \cdot \Delta\underline{w} \tag{5}$$

where $\Delta\underline{w} = \left[\, (\Delta\underline{w}^0)^T (\Delta\underline{w}^1)^T \,\right]^T$, $\Delta\underline{w}^0 = \mathrm{vec}\left\{\Delta\mathbf{W}^0\right\}$, with $\mathrm{vec}\left\{\Delta\mathbf{W}^0\right\}$ denoting a vector formed by stacking up all columns of $\Delta\mathbf{W}^0$; vector $\underline{c}$ and matrix $\mathbf{A}$ are appropriately expressed in terms of the previous network weights. In particular,

$$\underline{c} = \left[z_a(\underline{x}_1) \cdots z_a(\underline{x}_{m_c})\right]^T - \left[z_b(\underline{x}_1) \cdots z_b(\underline{x}_{m_c})\right]^T,$$

expressing the difference between network outputs after and before retraining for all input vectors in S_c. $\underline{c}$ can be written as

$$\underline{c} = \left[d_1 \cdots d_{m_c}\right]^T - \left[z_b(\underline{x}_1) \cdots z_b(\underline{x}_{m_c})\right]^T \tag{6}$$

Equation (6) is valid only when weight increments $\Delta\underline{w}$ are small quantities. It can be shown [2] that, given a tolerated error value, proper bounds ϑ and ϕ can be computed for the weight increments and input vector $\underline{x}_i$ in S_c

Let us assume that the network weights before retraining, i.e., $\underline{w}_b$, have been estimated as an optimal solution over data of set S_b. Furthermore, the weights after retraining are considered to provide a minimal error over all data of the current set S_c. Thus, minimization of the second term of (2), which expresses the effect of the new network weights over data set S_b, can be considered as minimization of the absolute difference of the error over data in S_b with respect to the previous and the current network weights. This means that the weight increments are minimally modified, resulting in the following error criterion

$$E_S = \left\| E_{f,a} - E_{f,b} \right\|_2 \tag{7}$$

with $E_{f,b}$ defined similarly to $E_{f,a}$, with $\underline{z}_a$ replaced by $\underline{z}_b$ in (2).

It can be shown [2] that (7) takes the form of

$$E_S = \frac{1}{2}(\Delta \underline{w})^T \cdot \mathbf{K}^T \cdot \mathbf{K} \cdot \Delta \underline{w} \tag{8}$$

where the elements of matrix **K** are expressed in terms of the previous network weights $\underline{w}_b$ and the training data in S_b. The error function defined by (8) is convex since it is of squared form. The constraints include linear equalities and inequalities. Thus, the solution should satisfy the constraints and minimize the error function in (8). The gradient projection method is adopted to estimate the weight increments.

Each time the decision mechanism ascertains that retraining is required, a new training set S_c is created, which represents the current condition. Then, new network weights are estimated taking into account both the current information (data in S_c) and the former knowledge (data in S_b). Since the set S_c has been optimized over the current condition, it cannot be considered suitable for following or future states of the environment. This is due to the fact that data obtained from future states of the environment may be in conflict with data obtained from the current one. On the contrary, it is assumed that the training set S_b, which is in general provided by a vendor, is able to roughly approximate the desired network performance at any state of the environment. Consequently, in every network retraining phase, a new training set S_c is created and the previous one is discarded, while new weights are estimated based on the current set S_c and the old one S_b, which remains constant throughout network operation.

4 Decision Mechanism for Network Retraining

The purpose of this mechanism is to detect when the output of the neural network classifier is not appropriate and consequently to activate the retraining algorithm at those time instances when a change of the environment occurs.

Let us index images or video frames (similar definitions are used for speech signals) in time, denoting by $\underline{x}(k, N)$ the feature vector of the kth image or image frame, following the image at which the Nth network retraining occurred. Index k is therefore reset each time retraining takes place, with $\underline{x}(0, N)$ corresponding to the feature vector of the image where the Nth retraining of the network was accomplished. Retraining of the network classifier is accomplished at time instances where its performance deteriorates, i.e., the current network output deviates from the desired one. Let us recall that vector $\underline{c}$ expresses the difference between the desired and the actual network outputs based on weights $\underline{w}_b$ and applied to the current data set S_c. As a result, if the norm of vector $\underline{c}$ increases, network performance deviates from the desired one and retraining should be applied. On the contrary, if vector $\underline{c}$ takes small values, then no retraining is required. In the following we denote this vector as $\underline{c}(k,N)$ depending upon feature vector $\underline{x}(k,N)$.

Let us assume that the Nth retraining phase of the network classifier has been completed. If the classifier is then applied to all instances $\underline{x}(0,N)$, including the ones used for retraining, it is expected to provide classification results of good quality. The difference between the output of the retrained network and of that produced by the initially trained classifier at feature vector $\underline{x}(0,N)$ constitutes an estimate of the level of improvement that can be achieved by the retraining procedure. Let us denote by $e(0,N)$ this difference and let $e(k,N)$ denote the difference between the corresponding classification outputs, when the two networks are applied to the feature set of the kth image or image frame (or speech segment) following the Nth network retraining phase. It is anticipated that the level of improvement expressed by $e(k,N)$ will be close to that of $e(0,N)$ as long as the classification results are good. This will occur when input images are similar to the ones used during the retraining phase. An error $e(k,N)$, which is quite different from $e(0,N)$, is generally due to a change of the environment. Thus, the quantity $a(k,N)=|e(k,N)-e(0,N)|$ can be used for detecting the change of the environment or equivalently the time instances where retraining should occur. Thus, no retraining is needed if:

$$a(k,N)<T \tag{9}$$

where T is a threshold which expresses the max tolerance, beyond which retraining is required for improving the network performance. In case of retraining, index k is reset to zero while index N is incremented by one.

Such an approach detects with high accuracy the retraining time instances both in cases of abrupt and gradual changes of the operational environment since the comparison is performed between the current error difference $e(k,N)$ and the one obtained right after retraining, i.e., $e(0,N)$. In an abrupt operational change, error $e(k,N)$ will not be close to $e(0,N)$; consequently, $a(k,N)$ exceeds threshold T and retraining is activated. In case of a gradual change, error $e(k,N)$ will gradually deviate from $e(0,N)$ so that the quantity $a(k,N)$ gradually increases and retraining is activated at the frame where $a(k,N)>T$.

Network retraining can be instantaneously executed each time the system is put in operation by the user. Thus, the quantity $a(0,0)$ initially exceeds threshold T and retraining is forced to take place.

5 Application to Multimodal Emotion Analysis

5.1 How to Combine Modalities

While evaluating the user's emotional state, information on one modality can be used to disambiguate information on the other ones. Two obvious approaches exist of fusing information from different cues: the first is to integrate information at the signal or feature level, whereas the second is to process information and make a decision independently on each modality and finally fuse those decisions at semantic level.

For the first strategy, namely fusion at the signal level, to be meaningful, two conditions must be satisfied: first, modalities must have features that can be handled in a similar way and second the modalities must be synchronized. Such is the case in the combined speech and lip movement analysis. The obvious disadvantages of treating inputs on the signal level include the requirement of large amounts of training data, and the inability to combine the fusion process with possible knowledge about the internal mechanisms present in physical multimodal understanding.

On the other hand, fusion on the decision level, can be applied to modalities which have different time scale characteristics; in this case timing in each modality can be different not only on the frequency of feature extraction but also on the time interval where each decision is valid. For example, an audio prosodic feature concerning some milliseconds of speech could reveal a specific emotional speaker disposition, while the presence of a facial expression could have to be detected for several seconds before it reveals a specific underlying emotion. Decision-level fusion offers several advantages over feature-level fusion. Firstly, each modality is treated independently therefore, they can be both separately trained and their integration does not require excessive computation. A disadvantage of this method is the fact that it does not support mutual disambiguation: using information from one modality to enhance or reject information coming from the other.

In the current approach, a novel technique is proposed, based on the above described adaptive neural network retraining detection. In particular, the proposed approach is applied separately, but synchronised, to the two modalities. The performance of each unimodal classifier is monitored through the decision mechanism of section 4. Whenever a deterioration of performance in one modality is detected, the other one, if still successful, is used to provide the desired outputs for retraining the modality where the problem occurred. The experimental study is presented next.

5.2 The Experimental Study

In this work, we analyzed naturalistic data from the EU IST HUMAINE Network of Excellence [5] naturalistic database. The database includes persons driven to real emotional discourse, being annotated in valence and activity terms by several experts. Both facial expression information in the form of MPEG-4 features [9], and prosodic audio features were extracted from the same data and feature-level classification was employed. Our main synchronization unit has been chosen to be audio tunes, i.e. for the video analysis MPEG-4 FAPs have been extracted on each video frames both at the location of tunes and at the location of silence between tunes (a tune being the portion of the pitch contour that lies between two audio pause boundaries) [11,14]. We observed that in the majority of the cases from a subjective point of view, a tune defined with audio pauses of at least 150 ms seems to be a good segmentation at the sentence level.

Regarding training, testing and performance evaluation of automatic recognizers of multimodal data, a frequent problem is the absence of labelling on separate modalities. The work here is really at its infancy: there are only one or two annotated naturalistic databases, and those have not been annotated separately on each modality, i.e. having human experts produce an emotional annotation by watching only one modality at a time. Moreover, there is the question of the labelling synchronization: when dealing with tune segments, is it proper to reduce continuous labeling to tune labeling instead of first defining tunes and then labeling them?

5.3 Extraction of Visual Features

At first face detection is performed using nonparametric discriminant analysis with a Support Vector Machine (SVM) [6], which classifies face and non-face areas by reducing the training problem dimension to a fraction of the original with negligible loss of classification performance. The face detection step provides us with a rectangle head boundary which includes the whole face area. The latter is segmented roughly using static anthropometric rules [1] into three overlapping rectangle regions of interest which include both facial features and facial background; these three feature-candidate areas include the left eye/eyebrow, the right eye/eyebrow and the mouth. Continuing, we utilize these areas to initialize the feature extraction process. Facial feature extraction performance depends on head pose, thus head pose needs to be detected and the head restored in the upright position; in this work we are mainly concerned with roll rotation, since it is the most frequent rotation encountered in real life video sequences.

Head pose is estimated through the detection of the left and right eyes in the corresponding eye candidate areas. After locating the eyes, we can estimate head roll rotation by calculating the angle between the horizontal plane and the line defined by the eye centers. For eye localization we propose an efficient technique using a feed-forward back propagation neural network with a sigmoidal activation function. The multi-layer perceptron (MLP) we adopted employs Marquardt-Levenberg learning [8] while the optimal architecture obtained through pruning has two 20 node hidden layers and 13 inputs.

We apply the network separately on the left and right eye-candidate face regions. For each pixel in these regions the 13 inputs to the neural network are the luminance Y, the Cr & Cb chrominance values and the 10 most important DCT coefficients (with zigzag selection) of the neighboring 8x8 pixel area. The MLP has two outputs, one for each class, namely eye and non-eye, and it has been trained with more than 100 hand-made eye masks that depict eye and non-eye area in random frames from the ERMIS and HUMAINE [5] databases, in images of diverse quality, resolution and lighting conditions.

Eyes are located with the aid of the aforementioned network while this information is also combined with other feature detectors in a fusion process, to create facial feature masks, i.e. binary maps indicating the position and extent of each facial feature. The left, right, top and bottom–most coordinates of the eye and mouth masks, the left, right and top coordinates of the eyebrow masks as well as the nose coordinates, are used to define the considered feature points (FPs).

For the nose and each of the eyebrows, a single mask is created. On the other hand, since the detection of eyes and mouth can be problematic in low-quality images, a variety of methods is used each resulting in a different mask. In total, we have four masks for each eye and three for the mouth. These masks have to be calculated in near-real time, thus avoiding utilizing complex or time-consuming feature extractors. The use of the afore-mentioned neural network greatly serves this scope. The feature extractors developed for this work are described in [4].

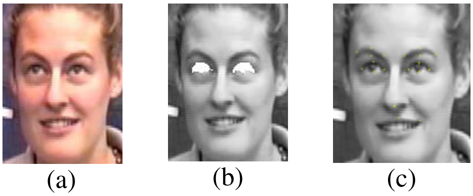

Fig. 1. (a) original frame, (b) final mask for the eyes, (c) detected feature points from the mask

Eyebrows are detected with a procedure involving morphological edge detection and feature selection using data from [1]. Nose detection is based on nostril localization. Nostrils are easy to detect due to their low intensity. Connected objects (i.e. nostril candidates) are labeled based on their vertical proximity to the left or right eye, and the best pair is selected according to its position, luminance and geometrical constraints from [1].

Since, as was already mentioned, the detection of a mask using the applied methods can be problematic, all detected masks have to be validated against a set of criteria. Each one of the criteria examines the masks in order to decide whether they have acceptable size and position for the feature they represent. This set of criteria consists of relative anthropometric measurements, such as the relation of the eye and eyebrow vertical positions, which when applied to the corresponding masks produce a value in the range [0,1] with zero denoting a totally invalid mask. More information about the used expression profiles can be found in [9].

5.4 Extraction of Audio Features

The features used in this work are exclusively based on prosodic features. We consider here features related to pitch and rhythm. All information related to emotion that one can extract from pitch is probably not only in these features, but the motivation of this approach is in the desire to develop and use a higher level of speech prosody analysis than the usual pitch features used in previous studies.

We analyzed each tune with a method employing prosodic representation based on perception called 'Prosogram'. Prosogram is based on a stylization of the fundamental frequency data (contour) for vocalic (or syllabic) nuclei. It gives globally for each voiced nucleus a pitch and a length. According to a 'glissando threshold' in some cases we don’t get a fixed pitch but one or more lines to define the evolution of pitch for this nucleus. This representation is in a way similar to the 'piano roll' representation used in music sequencers. This method, based on the Praat environment, offers the possibility of automatic segmentation based both on voiced part and energy maxima. From this model/representation stylization we extracted several types of features: pitch interval based features, nucleus length features and distances between nuclei.

In musical theory, ordered pitch interval is the distance in semitones between two pitches upwards or downwards. For instance, the interval from C to G upward is 7, but the interval from G to C downwards is −7. Using integer notation (and eventually modulo 12) ordered pitch interval, *ip*, may be defined, for any two pitches *x* and *y*, as:

$$\begin{aligned} ip\left\langle y,x\right\rangle &= x-y \\ ip\left\langle x,y\right\rangle &= y-x \end{aligned} \tag{1}$$

In this study we considered pitch intervals between successive voiced nuclei. For any two pitches *x* and *y*, where *x* precedes *y*, we calculate the interval $ip\left\langle x,y\right\rangle = y-x$, then deduce the following features.

For each tune, feature (f1) is the minimum of all the successive intervals in the tune. In a similar way, we extract the maximum (f2), the range (absolute difference between minimum and maximum) (f3), of all the successive intervals in each tune. Using the same measure, we also deduce the number of positive intervals (f4) and the number of negative intervals (f5). Using the absolute value, a measure equivalent to the unordered pitch interval in music theory, we deduce a series of similar features: minimum (f6), maximum (f7), mean (f8) and range (f9) of the pitch interval. Another series of features is also deduced from the ratio between successive intervals, here again maximum (f10), minimum (f11), mean (f12) and range (f13) of these ratios give the related features. In addition to the aforementioned features, the usual pitch features have also been used such as fundamental frequency minimum (f14), maximum (f15), mean (f16) and range (f17). The global slope of the pitch curve (f18), using linear regression, has also been added.

As was previously said, each segment (voiced "nucleus" if it is voiced) of this representation has a length, and this has also been used in each tune to extract features related to rhythm. These features are, as previously, maximum (f19), minimum (f20), mean (f21) and range (f22). Distances between segments have also been used as features and the four last features we used are maximum (f23), minimum (f24), mean (f25) and range (f26) of these distances.

5.5 Adaptive Multimodal Emotion Analysis

In our study, we tested the proposed neural-network-based adaptive classification, evaluation and retraining procedure on the multimodal data sets that were described above. More than 100 tunes of speech and 1000 video frames showing four personalities reacting to an emotion provoking environment named SAL (Sensitive Artificial Listener) developed in the framework of the IST NoE Humaine. The goal was to classify each instant of visual and speech input to one of the quadrants of the emotional wheel, which measures emotion based on a 2-D representation, where dimensions correspond to activation and evaluation of interaction.

While the basic classification rates for each input modality (speech, face) were close to 67%, by implementing the retraining procedure, whenever a change of personality or a lower performance measure was detected, and relying on the cues provided by both modalities, the classification rate was raised to 79%, which illustrates the ability of the proposed method to take advantage of multimodal analysis for improving the obtained results in emotion analysis and classification problems.

6 Conclusions

A novel neural network on line retraining procedure has been proposed in this paper, which is appropriate for real life analysis of multimedia applications. Illustration of the method's ability to achieve multimodal emotion recognition is given in this paper using naturalistic audio and visual data, created in the HUMAINE IST Network of Excellence (2004-2008). The proposed approach is based on neural network architectures which examine each input modality, monitoring the performance of the classification operation and provide a measure of confidence on the achieved accuracy. Whenever this measure gets unacceptable, an efficient on-line retraining of the network knowledge takes place, using the gradient projection method and combining input from all modalities under investigation. Extensive studies are currently under implementation, for further evaluation of the method capabilities.

References

1. J.W. Young, Head and face anthropometry of adult U.S. civilians, FAA Civil Aeromedical Institute, 1993.
2. A. Doulamis, N.Doulamis and S. Kollias, On-line Retrainable Neural Networks: Improving the Performance of Neural Networks in Image Analysis Problems, IEEE Transactions on Neural Networks, vol. 11, no 1, pp. 137-157, 2000.
3. A. Krog, J. Vedelsby, Neural network ensembles, cross validation and active learning, in Tesauro G., Touretzky D., Leen T. (Eds) Advances in neural information processing systems 7, pp. 231-238, Cambridge, MA. MIT Press, 1995.
4. S. Ioannou, A. Raouzaiou, V. Tzouvaras, T. Mailis, K. Karpouzis and S. Kollias, Emotion recognition through facial expression analysis based on a neurofuzzy network, Special Issue on Emotion: Understanding & Recognition, Neural Networks, Elsevier, Volume 18, Issue 4, Pages 423-435, 2005.
5. HUMAINE, Human-Machine Interaction Network on Emotion IST-2002-2.3.1.6 (http://emotion-research.net/)
6. R. Fransens, Jan De Prins, SVM-based Nonparametric Discriminant Analysis, An Application to Face Detection, Ninth IEEE International Conference on Computer Vision Volume 2, October 13 - 16, 2003
7. S. Kollias and D. Anastassiou. “An adaptive least squares algorithm for the efficient training of artificial neural networks”. IEEE Transactions on Circuits and Systems, Volume: 36 , Issue: 8 , Aug. 1989 pp:1092 – 1101
8. M.T. Hagan and M. Menhaj, "Training feedforward networks with the Marquardt algorithm". IEEE Transactions on Neural Networks, vol. 5, no. 6, pp. 989-993, 1994.
9. A. Raouzaiou, N. Tsapatsoulis, K. Karpouzis and S. Kollias, “Parameterized facial expression synthesis based on MPEG-4”, EURASIP Journal on Applied Signal Processing, Vol. 2002, No. 10, pp. 1021-1038, Hindawi Publishing Corporation, October 2002.
10. Baldonado, M., Chang, C.-C.K., Gravano, L., Paepcke, A.: The Stanford Digital Library Metadata Architecture. Int. J. Digit. Libr. 1 (1997) 108–121

11. Mertens, Piet: The Prosogram: Semi-Automatic Transcription of Prosody based on a Tonal Perception Model. in B. Bel & I. Marlien (eds.) Proceedings of Speech Prosody 2004, Nara (Japan), 23-26 March. (ISBN 2-9518233-1-2)
12. Cowie R., Douglas-Cowie E., Tsapatsoulis N., Votsis G., Kollias S., Fellenz W., Taylor J., Emotion Recognition in Human-Computer Interaction, IEEE Signal Processing Magazine, 2001.
13. R. W. Picard, Affective Computing, MIT Press, Cam-bridge, MA, 2000.
14. R. Cowie, E. Douglas-Cowie, Automatic statistical analysis of the signal and prosodic signs of emotion in speech. Proceedings of the 4th International Conference of Spoken Language Processing (pp. 1989–1992). 1996, Philadelphia, USA.