# An Adaptable Emotionally Rich Pervasive Computing System

*Nikolaos Doulamis*

Department of Electrical & Computer Engineering
National Technical University of Athens
9, Heroon Polytechniou street, 157 73, Zografou-Athens, Greece
ndoulam@cs.ntua.gr

## ABSTRACT

*Different people express their feelings in a different way under different circumstances. For this reason, an adaptable architecture is proposed in this paper able to automatically update its performance to a particular individual. This means that the system takes into account the specific user's characteristics and properties and adapts its performance to the specific user's needs and preferences. The architecture also takes into account the context of the environment, which significantly affects the way that people express their emotions (family, friends, working environment). As a result, the same expressions may lead to different emotional states in accordance to the specific environment to these feelings are expressed. The adaptation is performed using concepts derived from functional analysis. The presented adaptable architecture requires low memory and processing capabilities and thus it can be embedded in smart pervasive devices of low processing requirements. Experimental results on real-life databases illustrate the efficiency of the proposed scheme in recognizing the emotion of different people or even the same under different circumstances.*

## 1. INTRODUCTION

The widespread use of the internet, mobile technologies and the development of a wide variety of inexpensive smart computer equipment and devices has lead to a new dimension of interacting with the environment and coordinating these devices with each other. Furthermore, emerging standards in wireless communications enable embedded devices to intercommunicate and ubiquitously access information. These trends have led to a change from the traditional computer centered to a future human-centered information access mode known as pervasive computing [1].

Human machine communication becomes a usual activity in everyday life. In our information society more and more people have to interact with a machine, e.g., a call or information providing centre, probably distributed in various places. Making this interaction as efficient and friendly as possible, by including intelligence to the machines that interact with people, is a crucial aspect which is currently attracting R&D efforts worldwide. Spoken language interfaces to computers are a topic that has lured and fascinated engineers and speech scientists alike for over five decades.

It should, however, be mentioned that there are two channels in human interaction. One transmits explicit messages, about the content; the other transmits implicit messages about the speakers themselves. Both linguistics and technology have invested enormous efforts in understanding the first, explicit channel, but the second is much less well understood. Understanding the other party's emotions and senses, i.e. non-verbal information, is one of the key tasks associated with the second, implicit channel. The implicit channel is a major feature of human communication, and progress towards reproducing it is crucial for various applications related to human computer interaction (HCI), especially regarding user friendliness, e-inclusion issue of the elderly or other persons of special needs. .

However, humans express their feelings and emotions with a totally different way *in a different context*. We are in general more explicit in our homes than in the outdoor world. In addition, our feelings are expressed more explicitly in friends and family rather than in unknown or not-well known people. Furthermore, even in the same conditions and context, different humans react in a different way according to cultural, region, or educational characteristics. Thus, an emotion recognition system cannot be considered static for all humans and situations. Instead, it should be adapted to a particular user and context.

A facial expression emotion recognition system consists of two main modules; the face extraction and the classification module. Robust and accurate facial analysis and feature extraction has always been a complex problem that has been dealt with by posing presumptions or restrictions with respect to facial rotation and orientation, occlusion, lighting conditions and scaling. These restrictions are being eventually revoked in the literature, since authors deal more and more with realistic environments, while keeping in mind pioneering works in the field [1][2]. Methods for facial feature extraction use either stereo vision, or statistical modeling, or hierarchical image decomposition, or probabilistic theory. A more detailed of these approaches can be found in [3].

Usually for the classification, highly non-linear models are used, such as neural network architectures. Although these systems provide satisfactory results in the conditions in which the models have been trained, their performance severely deteriorates for data outside the training environment. This is due to the fact that, as we have mentioned above, different people or even the same under different circumstances express their emotions in a different way. Therefore, regardless of the model efficiency and the representativity of the samples used for model training, such systems cannot satisfactorily be applied to different people or different context than that they have been trained on.

To overcome the aforementioned problems, in this paper, we present an adaptable non-linear model for recognizing emotional states of a user based on their facial expressions. The main advantage of the proposed system is its ability to adapt its performance to the specific way that a user expresses his/her feelings. The adaptation is performed with a computationally efficient algorithm, which can be applied to smart embedded devices of low processing and memory requirements. For the adaptation a set of representatives samples is used which characterize the context of the current environment. Then, an adaptive algorithm is implemented which updates the model parameters to the current conditions. More specifically, the adaptation is performed so that a) the system response satisfies the current context as much as possible, while simultaneously a minimal modification of the already acquired knowledge of the model is obtained.

## 2. THE ADAPTABLE EMOTIONALLY RICH PERVASIVE COMPUTING ARCHITECTURE

An Emotionally Rich (ER)- pervasive computing system should be able to a) perform collection of signals from different sensors (speech, bio-signals or facial expressions), b) pre-process and extract features, c) analyze and understand the extracted feature data d) make decisions and finally e) manage the interaction with their users. Therefore, the core of an ER pervasive computing system is the module responsible for feature extraction and processing, and the classification module, which maps the extracted features to the actual human emotional states.

Let us assume in the following that some features have been extracted from a human face analysis and included in a feature vector say, $\mathbf{f}_i$ where index *i* corresponds to the ith human individual. Let us also assume that *p* humans' emotions are recognized by the proposed ER- pervasive computing architecture and thus each vector $\mathbf{f}_i$ is classified to one of the *p* available classes $\omega_j$, $j=1,2,\ldots,p$, i.e., emotions. As it can be shown in several psychological paper, six emotions are basic; the happiness, sadness, fear, anger, surprise, and disgust [4]. It is not in doubt that these basic emotions are key points of reference and can be described as 'archetypal emotions', which reflects the fact that they are undeniably the obvious examples of an emotion state. New studies attempt to define a set of terms covering a wider range of emotions without becoming unmanageable [1].

Let us denote in the following as $\mathbf{y}(\mathbf{f}_i)$, a *p*-dimensional vector, the elements of which $p_{ek}^{(i)}$ contains the degrees of coherence of vector $\mathbf{f}_i$ to one of the kth emotional state among the *p* available,

$$\mathbf{y}(\mathbf{f}_i) = [p_{e1}^{(i)}\ p_{e2}^{(i)} \cdots p_{ep}^{(i)}]^T \tag{1}$$

Let us assume in the following, *for simplicity purposes and clarification reasons*, a two-class classification problem, where the two classes $\omega_1$, $\omega_2$ refer, for example, to two emotional states, such as happiness and sadness. Extension to multiple emotional states can be performed in a similar way. In this case the model output is scalar, say $y(\mathbf{f}_i)$ instead of a vector (output values close to one (1) correspond to the first emotional state, while values close to zero to the second emotional state.).

Then, using concepts derived from functional analysis, it can be proved that any continuous non-linear function can be expressed as a parametric relation of known functional components $\Phi_l(\cdot)$ within any degree of accuracy. Then,

$$y(\mathbf{f}_i) \approx \sum_{l=1}^{L} v_l \cdot \Phi_l\left(\sum_{k=1}^{P} w_{k,l}(f_{i,k})\right) \tag{2}$$

where $f_{i,k}$ refers to the kth element of feature vector $\mathbf{f}_i$, the $v_l$ and $w_{k,l}$ to the model parameters (coefficients), and $Q$ the size of the feature vector $\mathbf{f}_i$. Variable *L* corresponds to the approximation order of the model. The most familiar class of functional components $\Phi_l(\cdot)$ is the sigmoid functions.

The model parameters $v_l$ and $w_{k,l}$ are estimated using a set of representative samples and then by applying a training algorithm. Let us denote as $S = \{(\mathbf{f}_1, d_1), \cdots, (\mathbf{f}_N, d_N)\}$ this set, where *N* is the total number of samples and $d_i$ the desired output vector for the ith human individual. The values of $d_i$ is equal to 1 for the first emotional state and 0 for the second emotional state. Let us also denote as $\mathbf{w}$ a vector, which contains all parameters $v_l$ and $w_{k,l}$ of equation (2), i.e., $\mathbf{w} = [\cdots v_i \cdots w_{kl} \cdots]^T$.

### 2.1 Adaptable Emotionally Rich Architecture

As we have mentioned above, each human individual has his/her own way for expressing his/her emotions. Thus, direct application of the model of (2), in which the model parameters are considered constant, would *not* work well, since the model cannot be adapted to the different ways that humans use to express their emotions. To overcome this difficulty, we equip the proposed pervasive architecture with an adaptable mechanism, which can update the system response with respect to the current way that the individual use to express their emotions.

In this way, a new set of parameters, say $\mathbf{w}_a$, should be created for each emotional state, which are capable of updating the response of the ER pervasive architecture to the current actual way that the users express their emotions. To perform the adaptation, a new set of representative samples, say $S_c = \{(\mathbf{f}'_1, d'_1), \cdots, (\mathbf{f}'_M, d'_M)\}$ should be created, which has a similar form to the set $S$. Then, the new parameters $\mathbf{w}_a$ are estimated by minimizing the following error criterion,

$$\mathbf{w}_a = \arg\min_{\mathbf{w}} E = \frac{1}{2}\sum_{i=1}^{N}(y_{\mathbf{w}}(\mathbf{f}_i) - d_i)^2 \quad \text{for all data in } S \tag{3}$$

and

$$y_{\mathbf{w}_a}(\mathbf{f}_i) = d'_i \quad i = 1,\ldots,M, \quad \text{for all data in } S_c \tag{4}$$

In equations (3,4) we added the dependence of the model output $y_{\mathbf{w}}(\mathbf{f}_i)$ on the model parameters **w**. Equations (3,4) indicate that the new model parameters are estimated so that i) the *current data*, as expressed by the set $S_c$, *are satisfied as much as possible*, while simultaneously ii) *a minimal degradation* of the previous obtained knowledge is derived.

### 2.2 Optimal Estimation of the Model Parameters

In this section, we describe a novel, fast and reliable algorithm for estimating the new model parameters $\mathbf{w}_a$. Initially, we assume that a small perturbation of the model parameters before the adaptation is enough to achieve good classification performance. Then,

$$\mathbf{w}_a = \mathbf{w} + \Delta\mathbf{w} \tag{5}$$

where $\Delta\mathbf{w}$ is a small incremental vector. This assumption leads to an analytical and tractable solution for estimating $\mathbf{w}_a$, since it permits linearization of functional components $\Phi_l(\cdot)$ of (2) using a first order Taylor series expansion.

It can be shown in [5] that linearization of (4) with respect to the weight increments $\Delta\mathbf{w}$ is equivalent to a set of linear equations

$$\mathbf{c} = \mathbf{A} \cdot \Delta\mathbf{w} \tag{6}$$

where vector $\mathbf{c}$ and matrix **A** are appropriately expressed in terms of the previous model parameters $\mathbf{w}$ [5].

The size of vector $\mathbf{c}$ is smaller than the number of unknown weights $\Delta\mathbf{w}$, since in general a small number *M*, of current data are available. Thus, many solutions exist for Eq. (6), since the number of unknowns is much greater than the respective number of equations. Uniqueness, however, is imposed by an additional requirement, which takes into consideration the previous network knowledge. Among all possible solutions that satisfy (6), the one which causes a minimal degradation of the previous model knowledge is selected as the most appropriate. This is expressed by equation (3).

It can be shown [5] that (3) takes the form of

$$E = \frac{1}{2}(\Delta\mathbf{w})^T \cdot \mathbf{K}^T \cdot \mathbf{K} \cdot \Delta\mathbf{w} \tag{7}$$

where the elements of matrix **K** are expressed in terms of the previous network weights $\mathbf{w}$ and the data in the set $S$. Thus, the problem results in the minimization of (7) subject to constraints of (6).

The error function defined by (7) is convex since it is of squared type, while the constraints of (6) are linear equalities. Thus, the solution should lie on the hyper-surface defined by (6) and simultaneously minimize the error function given in (7). The gradient projection method is used in this paper to solve this problem. The gradient projection method starts from a feasible point and moves in a direction, which decreases $E$ and simultaneously satisfies the constraints; a point is called feasible, if it satisfies all constraints. The weights are adapted as follows:

$$\Delta\mathbf{w}(n+1) = \Delta\mathbf{w}(n) + \mu(n)\mathbf{h}(n) \quad (8)$$

where $n$ is the iteration index and $\mu(n)$ is a scalar that determines the rate of convergence. Using the methodology of [6] we can estimate vector $\mathbf{h}(n)$ as

$$\mathbf{h}(n) = -\mathbf{P}\nabla E = -\mathbf{Q}\Delta\mathbf{w} \quad (9)$$

$$\text{with } \mathbf{P} = \mathbf{I} - \mathbf{A}^T(\mathbf{A}\ \mathbf{A}^T)^{-1}\mathbf{A} \quad \text{and} \quad \mathbf{Q} = \mathbf{P}\mathbf{K}^T\mathbf{K} \quad (10)$$

using $\nabla E$ computed from (9). The computational complexity required to independently update each network weight is proportional to the number of network weights.

### 2.3 Computational Complexity of Adaptation Process

The computational complexity of the adaptation algorithm, in contradiction to the generally long training periods of the initial model parameters, is very small. In particular, the adaptation process is separated in two phases; the initialization phase where matrix $\mathbf{Q}$, which projects the negative gradient of function $E$ onto the surface defined by the constraints of (6), is estimated, and the iteration phase where the weight increments $\Delta\mathbf{w}$ are updated. The iteration phase involves a simple multiplication of matrix **Q,** of size $N_w \times N_w$, by vector $\Delta\mathbf{w}$, of size $N_w \times 1$, where $N_w$ denotes the number of model parameters. For a typical value of $N_w$ equal to 500, the product $\mathbf{Q}\cdot\Delta\mathbf{w}(n)$ in (9) requires few msec to be executed (~30 ms on a P-II PC). The number of iterations $n$ that the gradient projection method requires to derive the optimal solution is in general small (8-12 iterations are usually sufficient to obtain a weight increment close to the optimal one). Thus, the computational cost of the iteration phase is about half a sec.

In the initialization phase, matrix $\mathbf{Q}$ is estimated as a product of matrix $\mathbf{P}$ and matrices $\mathbf{K}^T\cdot\mathbf{K}$. The main computational load of $\mathbf{P}$ is the estimation of the inverse matrix $(\mathbf{A}\cdot\mathbf{A}^T)^{-1}$. However, the size of $\mathbf{A}\cdot\mathbf{A}^T$ is equal to the number of constraints, which is usually very small since it refers to the number of representative data of the current environment. Consequently, the computational complexity for matrix $(\mathbf{A}\cdot\mathbf{A}^T)^{-1}$ is significantly low (in the order of few msec). Finally, computation of $\mathbf{P}\cdot\mathbf{K}^T\cdot\mathbf{K}$ requires about few seconds for a typical network size of 500 weights. Thus, the total cost of retraining is small (of order of few seconds), allowing the efficient use of the proposed scheme to real-life interactive multimedia systems.

## 3. EMOTION RECOGNITION USING FACIAL EXPRESSIONS

Apart from the adaptation mechanism, required for updating the performance of a system with respect to the current actual way that human individuals use to express their emotions and feelings, the identification of the human facial features with an automatic way is also the other issue of interest. This means that the proposed ER- pervasive system should be able to extract facial features so as to derive cues about the users' emotional states.

Analysis of the emotional expression of a human face requires a number of pre-processing steps which attempt to detect or track the face, to locate characteristic facial regions such as eyes, mouth and nose on it, to extract and follow the movement of facial features, such as characteristic points in these regions, or model facial gestures using anatomic information about the face [7] [8]. Most of the above techniques are based on a well known system for describing "all visually distinguishable facial movements", called the Facial Action Coding System (FACS) [4]. FACS is an anatomically oriented coding system, based on the definition of "action units" of a face that cause facial movements and it has been implemented in the framework of MPEG-4 standard [9].

Various results have been presented regarding classification of archetypal expressions of faces, based on features or points mainly extracted from the mouth, eyes and eyebrow areas of the faces. These results indicate that facial expressions can be used to perceive, at some extend, a person's emotional state. Facial anatomy as well as social issues make the emotional analysis of faces to be more or less user dependent. The accuracy of facial feature movement estimation that is required to discern a large gamut of emotional states is much higher than when dealing with a small number of categories; this makes the former situation much more difficult [1] [3].

### 3.1 Face Analysis and Facial Feature Extraction

Detection of the position and shape of the mouth, the eyes, the eyelids, eyebrows, and wrinkles as well as extraction of feature related to them, require a preceding stage for *face detection and tracking*. Most face tracking techniques, which are appropriate for *real-time processing,* use colour as a clue for detection of facial regions. In this paper face detection is performed using chrominance information as described in the following. This is due to the fact that the chrominance components are directly available in the MPEG compressed domain and thus minimal decoding of compressed data is required, resulting in an algorithm of high computational efficiency

#### *3.1.1 Face Detection and Evaluation*

The two-chrominance components of a color image are used for efficiently performing human face detection. The selection of the two chrominance components is due to the fact that the face region occupies a very small region of the chrominance space [7].Thus, image regions, whose respective chrominance values are located at this small region, can be considered as face blocks. On the contrary, regions of chrominance with values located far from this region correspond to non-face regions.

The histogram of chrominance values corresponding to a face region is initially modeled by a Gausssian probability density function (pdf), thus [8],

$$P(\mathbf{x}\,|\,\Omega_f) = \frac{\exp(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_f)^T\cdot\boldsymbol{\Sigma}_f^{-1}\cdot(\mathbf{x}-\boldsymbol{\mu}_f))}{2\pi\cdot|\boldsymbol{\Sigma}|^{1/2}} \quad (11)$$

where $\mathbf{x}=[u\ v]^T$ is a 2x1 vector containing the mean chrominance components $u$ and $v$ of an examined block, $\boldsymbol{\mu}_f$ is the 2x1 mean vector of a face area and $\boldsymbol{\Sigma}$ is the 2x2 variance matrix of the pdf having the form. Parameters $\boldsymbol{\mu}_f$ and $\boldsymbol{\Sigma}$ are estimated based on a set of several face images and using the maximum likelihood algorithm.

After modeling the distribution of the chrominance components for face regions [i.e., estimating the parameters $\boldsymbol{\mu}_f$ and $\boldsymbol{\Sigma}$ of (15)], we can compute for each image area, the probability of belonging to a face or to a non-face class. Equation (11) indicates that an image region, say $R_i$ belongs to the face area, if the respective probability of its chrominance values, $P(\mathbf{x}(R_i)\,|\,\Omega_f)$ is high.

However, since the above mentioned process takes into consideration only color information, the final outcome may contain non-face areas whose chrominance components are similar to the face one, e.g., the human hands. For this reason, an evaluation step is required to discard non-face areas, which have been classified as face ones. The evaluation step is based on a template matching algorithm. In particular, the human face is approximated as a rectangular of specific aspect ratio.

### 3.2 Facial Feature and Gesture Analysis

Detection of the position and shape of the mouth, of the eyes, of the eyelids, of wrinkles and extraction of features related to them are targets of this step. A powerful technique to achieve this goal follows a hierarchical search [3] [8]. In particular, the basic features, i.e., eyes, nose and mouth are located first, while the more detailed features, such as parts of the eyes, nose, mouth, eyebrows, etc., are located relative to the basic feature locations. The first step that follows face detection is to normalise the face image so that the eyes are in a horizontal line with a fixed distance. The approximate locations of the other basic features are known from statistics of mean and variance, relative to the nose position, gathered on a training database. Using Fisher discriminate and distance from feature space templates, the locations with high probability to contain features are derived. Locations of more-detailed features are estimated similarly, relative to the nearest basic feature locations. Verification follows using collocation statistics, so that locations that are inconsistent with the other detected ones are pruned.

## 4. EXPERIMENTAL RESULTS

An experimental study has been conducted in the emotion recognition case using well known databases [1] such as the MediaLab database, the database of the UCSF Human Interaction Lab, and the database with natural video sequences developed in [10]. The initial training set ( $S$ ) includes faces of various people expressing six archetypal – type emotions, i.e. anger, sadness, happiness, disgust, fear and surprise, as well as their neutral state. More than one thousand images have been incorporated in this set. We then assumed that these data have been used to train the non-linear model. A second data set has been created with 20 images, portraying a specific end user, in each of the above seven emotional states. This data pool serves as retraining set ( $S_c$ ) for the weight adaptation procedure. Following adaptation, the product will be ready for use by its end user; a third data set composed of 170 other images with expressions of the same end user constitutes the test material ( $S_t$ ), on which the performance of the adaptive non-linear model is to be examined.

Using the above data sets, two kinds of experiments were carried out. In the first, the scenario of the proposed adaptive strategy was tested. The second experiment is used for comparison purposes. More specifically, the sample of the retraining set $S_c$ was used together with the initial set $S$ to train the non-linear classifier. The performance of the two substitutional solutions are collated and compared in the following figures.

Let us denote by $y$ the output of the initially trained classifier (i.e., using samples of the initially training set $S$ ), and by $y_c$ the output of the model after its adaptation (i.e., using samples of the retraining set $S_c$ ). We also denote as $y_o$ the output of the model the parameters of which have been estimated using samples of the union of the sets $S$ and $S_c$ .

The results for the "happy" expression are presented in Figures 2-3; In particular, Figure 2(a) compares the performance of the classifier that was trained with the set $S$ (i.e., the output $y$ ) with the output $y_c$ over a small subset of the initial set $S$ . On the other hand Figure 2(b) compares the outputs $y$ and $y_c$ over the samples of the set $S_c$ . Let us note that Figure 2(a) and 2(b) are in one sense complementary as expected; personalization of the classifier leads $y_c$ to a lower performance on the initial set $S$ , while the classification efficiency on data of set $S_c$ is improved. However, since the adaptation is performed with the minimal degradation of the already obtained model knowledge, the performance of $y_c$ on the set $S$ retains at some degree the original model integrity [see Figure 2(a)], while it is evident that the reverse statement is not valid, i.e. the output $y$ does not achieve acceptable performance rates on the set of the current environment $S_c$ , verifying the fact that model adaptation is required [see Figure 2(b)]. Figure 2(c) presents the outputs $y$ and $y_c$ over the samples of the test data of set $S_t$ . The effectiveness of the adaptation procedure is clear: the output of $y_c$ approaches unit values, which is not the case with the output of $y$ .

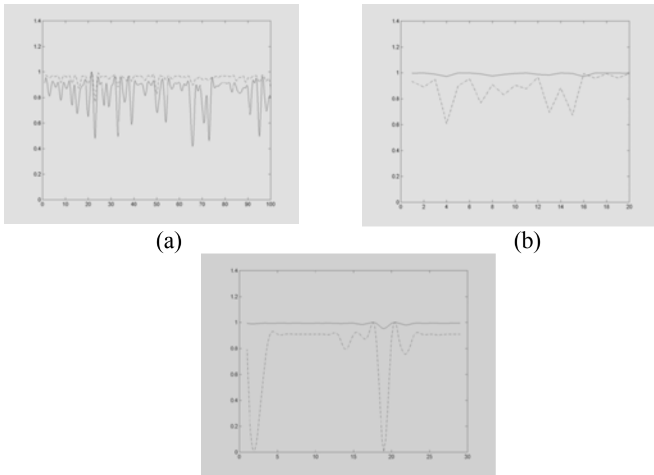
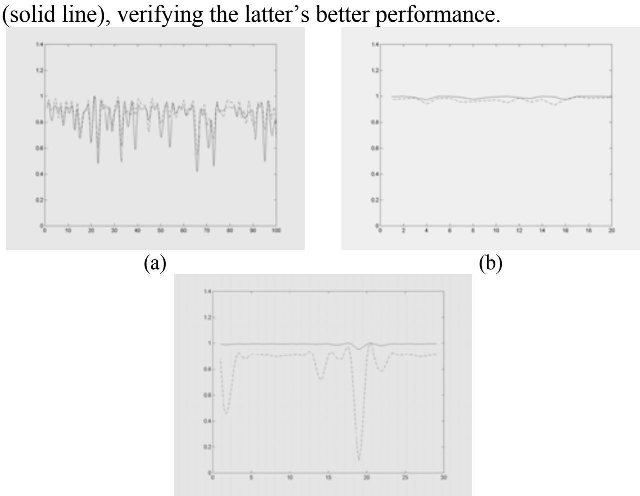


**Figure 2.** "Happy" expression. (a) Performance of networks $y$ (dashed) and $y_c$ (solid) over set $S$ . (b) Performance of $y$ (dashed) and $y_c$ (solid) over set $S_c$ . (c) Performance of networks $y$ (dashed) and $y_c$ (solid) over set $S_t$ . Horizontal axis shows the sample numbering, while vertical axis shows the relevance to the specific class.

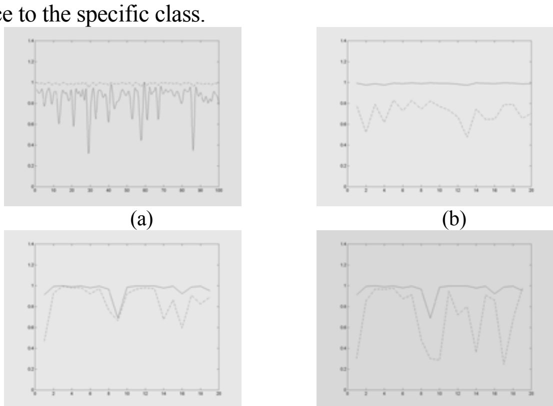
Figures 3(a)-(c) compare the performance of the classifier that has been trained from scratch using both the sets $S$ and $S_c$ , i.e., the output $y_o$ . It can be seen that the performances of $y_o$ and $y$ do not significantly differ. Figure 3(c) points out that $y_o$ has not adapted itself well enough to the new environment $S_t$ . Such an observation seems quite reasonable, since the number of samples of the retraining set $S_c$ is much smaller than the one of the initial set. Moreover, training from scratch might urge the $y_o$ to converge to a point in the weight parameter space, which is far from the one of the $y$ .

Results for the "angry" expression are presented in Figure 4. Figures 4(a)-(c) present similar results as the Figures 4(a)-(c). As it may be seen from Figure 4(b), the performance of $y$ over $S_c$ is even lower than in the case of the "happy" expression. This may be attributed to the lower activation nature of the "angry" expression with respect to the activation-evaluation space [1]. In Figure 4(d) the performance of the $y_o$ (dashed line) is compared versus the $y$ (solid line), verifying the latter's better performance.



(a) (b) (c)

**Figure 3.** "Happy" expression. (a) Performance of networks $y_o$ (dashed) and $y_c$ (solid) over set $S$. (b) Performance of networks $y_o$ (dashed) and $y_c$ (solid) over set $S_c$. (c) Performance of networks $y_o$ (dashed) and $y_c$ (solid) over set $S_t$. Horizontal axis shows the sample numbering, while vertical axis shows the relevance to the specific class.



(a) (b) (c) (d)

**Figure 4.** "Angry" expression. (a) Performance of networks $y$ (dashed) and $y_c$ (solid) over set $S$. (b) Performance of networks $y$ (dashed) and $y_c$ (solid) over set $S_c$. (c) Performance of networks $y$ (dashed) and $y_c$ (solid) over set $S_t$. (d) Performance on networks $y_o$ (dashed) and $y_c$ (solid) over set $S_t$. Horizontal axis in (a)-(d) shows the sample numbering.

Table I summarizes the results shown in Figures 2-4. The difference of performance between $y$ and $y_c$ on $S_t$, indicate that adaptation was efficient.

## 5. CONCLUSIONS

In this paper, we describe an Emotionally Rich (ER)-pervasive computing system. The system is able to recognize the emotional states of a user based on his/her facial expressions. Since, however, different people express their emotions in a different way, an efficient ER- pervasive computing system should be adapted so that its performance is updated to a specific user as well as to the current environmental conditions. This is addressed in this paper, by proposing an adaptable non-linear classifier. The adaptable model has the capability of updating its performance to the current context, i.e., to the way that a particular user express their feelings. The adaptable mechanism has been designed to be of low memory and processing requirements so that the proposed scheme can be applied in a smart embedded device. Experimental results on real-life emotional databases, as obtained within the environment of European R&D projects, show the good performance of the proposed scheme in recognizing the humans' emotions outperforming the static non-adaptable case.

**Acknowledgement:** The author would like to thank Dr. G. Votsis for his help in the experimental results.

**Table I.** Performance of networks $y$, $y_c$ and $y_o$ over all data categories for the "happy" and "angry" emotional state.

| | Mean value on $S$ | Mean value on $S_c$ | Mean value on $S_t$ |
|---|---|---|---|
| **Expression: Happy** | | | |
| $y$ | 0.981 | 0.896 | 0.895 |
| $y_c$ | 0.914 | 0.992 | 0.994 |
| $y_o$ | 0.926 | 0.971 | 0.929 |
| $\lvert y-y_c \rvert$ | 0.068 | 0.114 | 0.102 |
| **Expression: Angry** | | | |
| $y$ | 0.984 | 0.715 | 0.707 |
| $y_c$ | 0.901 | 0.989 | 0.939 |
| $y_o$ | 0.952 | 0.902 | 0.831 |
| $\lvert y-y_c \rvert$ | 0.084 | 0.274 | 0.246 |

## 6. REFERENCES


[1] R. Cowie, E. Douglas-Cowie, N. Tsapatsoulis, Y. Votsis, S. Kollias, W. Fellenz and J.Taylor, "Emotion Recognition and Human Computer Interaction," *IEEE Signal Processing Magazine,* No.1, pp. 32-80, January 2001.

[2] R. Chellappa, C.L. Wilson, S. Sirohey, "Human and machine recognition of faces: a survey," *Proc. of the IEEE*, Vol. 83, No.5, pp. 705–740, May 1995.

[3] G. Votsis, A. Drosopoulos and S. Kollias, "A Modular Approach to Facial Feature Segmentation on Real Sequences," *Signal Processing: Image Comm.,* Vol. 18, No. 1, pp. 97-89, 2003.

[4] P. Ekman and W. Friesen, The Facial Action Coding System. Consulting Psychologists Press, San Francisco, CA, 1978.

[5] N. Doulamis, A. Doulamis and S. Kollias, "On-Line Retrainable Neural Nets: Improving Performance of Neural Networks in Image Analysis Problems," *IEEE Trans. on Neural Networks,* Vol. 11, No 1, pp. 137-155, 2000.

[6] D. J. Luenberger, *Linear and non Linear Programming.* Addison-Wesley 1984.

[7] H. Wang and S. Chang, "A Highly Efficient System for Automatic Face Region Detection in MPEG Video Sequences", *IEEE Trans. on CSVT.*, vol. 7, no.4, pp. 615-628, August 1997.

[8] N. D. Doulamis, A. D. Doulamis and S. D. Kollias, "Improving the Performance of MPEG Encoding at Low Bit Rates Using Adaptive Neural Networks," *Journal of Real Time Imaging*, Academic Press, Vol. 6, No. 5, pp. 327- 345, October 2000.

[9] F. Pereira, "MPEG-4: Why, What, How and When?," *Image Communication,* Vol. 15, Nos. 4-5, pp. 271-279, January 2000.

[10] EC TMR Project "PHYSTA: Principled Hybrid Systems: Theory and Applications," http://www.image.ece.ntua.gr/physta.