A HYBRID INTELLIGENCE SYSTEM FOR FACIAL EXPRESSION RECOGNITION

A. Raouzaiou, N. Tsapatsoulis, V. Tzouvaras, G. Stamou and S. Kollias
Department of Electrical and Computer Engineering
National Technical University of Athens
Heroon Polytechniou 9, 157 73 Zographou, Greece
Phone: +30-10-7722491, Fax: +30-10-7722492
email: {araouz, ntsap, tzouvaras}@image.ntua.gr, gstam@softlab.ntua.gr, stefanos@cs.ntua.gr

ABSTRACT: This paper presents an experimental study dealing with facial expression recognition which examines the appropriateness of a hybrid intelligence architecture for subsymbolic to symbolic mapping. The facial expression recognition enhances interactivity and assists human-computer interaction issues, letting the system become accustomed to the current needs and feelings of the user. Actual application of this technology is expected in educational environments, 3D video conferencing and collaborative workplaces, online shopping and gaming, virtual communities and interactive entertainment.

KEYWORDS: facial expression recognition, hybrid intelligence system, human computer interaction, facial points.

INTRODUCTION

Availability of multimedia information anywhere and anytime is one of the key aims of ambient intelligence. On the other hand, multimedia information is increasing continuously: new data capture and sensor technologies will be generating petabytes and exabytes of data. Applications and interfaces that will be able to automatically analyse these data, exchange knowledge and make decisions in a given context, are strongly desirable. Natural and enjoyable user interactions with such applications will be based on autonomy, avoiding the need for the user to control every action, and adaptivity, so that they are contextualised and personalised, delivering the right information/decision at the right moment. The main challenges include exploration and definition of the ways that interfaces can provide users with information in an intelligent fashion:

- Taking into account users' wishes or needs, as well as the underlying physical and social context;
- Having a generic design, which, however, is able to take later into account personalization and to learn from interaction with users;
- Helping their users to make informed decisions about complex issues in real time;
- Doing so in a trusted manner, being able to explain or justify their suggestions or decisions.

In summary, in the real world, and in the continuously increasing amounts of information sources and knowledge bases, there exist: (a) Sensors and devices which collect and pre-process data; these provide raw, numerical data, and (b) knowledge bases (such as rule-based systems, ontologies) for specific tasks; these are in the form of rules, concepts and symbols. This category may also contain databases and databanks, such as data repositories which have been examined and annotated /characterized by experts.

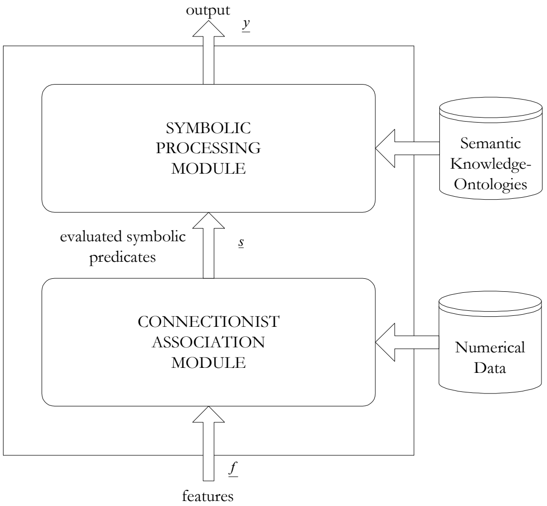
What is missing, and this is what the state-of-the-art calls for, are technologies/devices for effectively linking these two different types of symbolic and subsymbolic information, in real life situations. What we propose is that intelligence should be capable to handle both these types of information, i.e., symbolic and subsymbolic. The effectiveness of the proposed architecture is validated through a facial expression application demo.

This paper is organized as follows: In Section 1 it is presented the motivation that lead to the hybrid intelligent architecture and the overall scheme is outlined. In Section 2, the CAM (Connectionist Association Module) component is analysed by describing its operation and functionality. Section 3 presents the SPM module, giving emphasis to a new and innovative neurofuzzy network that is proposed. The operation and the functionality of the neurofuzzy network is outlined, while the rule insertion and extraction methods are analysed. In Section 5, the experimental work made on the facial expression recognition application and the results that have been produced are given. Finally, Section 6 presents conclusions are drawn.

SYSTEM ARCHITECTURE

The intelligence architecture of the system is a hybrid one, consisting of a connectionist (subsymbolic) association part and a symbolic processing part as shown in Fig. 1. In this modular architecture the *Connectionist Association Module* (CAM) provides the system with the ability of grounding the symbolic predicates (associating them with the input features), while the *Symbolic Processing Module* (SPM) implements a semantically rich reasoning process.

Let us first proceed to a more detailed description of Figure 1 architecture. The system takes as input a set of features and gives a set of recognised situations. The features are actually subsymbolic pre-processed measures, taken from external modules. Using the CAM, the set of features is associated with the set of evaluated symbolic predicates that have a semantic meaning. The above association uses a connectionist basis that has the ability to adapt its performance to its inputs, using numerical data (with the aid of supervised or unsupervised learning). SPM performs the conceptual reasoning process that finally results to the degree of which the output situations are recognised. This process can be adapted using structured and rule-based knowledge.



output
y
SYMBOLIC PROCESSING MODULE
Semantic Knowledge-Ontologies
evaluated symbolic predicates
s
CONNECTIONIST ASSOCIATION MODULE
Numerical Data
f
features

Figure 1: System Intelligence structure

There are two main reasons for using a hybrid structure like the one shown in Figure 1:

(1) Warnings / indications should be given to the human user in an informative way not provided by a neural network structure alone.

(2) Rules, describing situations are based on linguistic terms and are generally of the form “if *open_jaw_low* and *close_left_eyelid_low* then *Anger*”. It is, therefore, necessary to model somehow such kind of rules and an evident solution is the use of a neurofuzzy network. On the other hand a partitioning of the feature input space should be done in order to evaluate the symbolic predicates using the information provided by input features. But why we need a connectionist model (neural network) to a make this partitioning? Generally the internal state defined by the neural network output is not so simple to be considered as a simple fuzzy partitioning; instead the neural network performs the appropriate data clustering to provide the evaluation of the required symbolic predicates based on *numerical data*.

To summarise: we need a neurofuzzy structure to model both a priori and generally evolving knowledge, found in books, databases, provided by experts etc, expressed in terms of structured and shared knowledge (like a domain specific ontology) and a neural network structure to provide learning/adaptation capabilities.

THE CAM MODULE

CAM provides the ability of grounding the symbolic predicates (maps the input space $\boldsymbol{F}$ to the symbolic predicate space $\boldsymbol{S}$). It takes as input a set $f = [f_1, f_2, ..., f_n]$ of features and gives a set $s = [s_1, s_2, ..., s_m]$ of evaluated symbolic predicates that have a semantic meaning. The above association uses a connectionist basis that has the ability to adapt its performance to its inputs, using numerical data (with the aid of supervised or unsupervised learning). We define two phases in CAM’s lifecycle:

TRAINING PHASE

In this phase the CAM module is trained so as to be able to analyse the feature space of a particular domain. This step requires: (a) Using an appropriate set of training inputs *f*, (b) Collecting a representative set T_I of pairs (*f*, *s*) to be used for network training, and (c) Estimating a parameter set $\mathbf{W_I}$, which maps the input space ***F*** to the symbolic predicate space ***S***.

Feature analysis and problem understanding is the core of the CAM module. Its output will in general be the result of highly computationally complex analysis. It is, therefore, required that approximate heuristics must be (subsymbolically) learnt to get suitable approximate outputs. In most cases, such systems will be a personal support to their single owner; thus they must learn their habits and preferences. In the following, we will focus on classification tasks, which constitute the usual analysis problem.

In classification problems, it is required to classify each feature vector, $\underline{x}_i$, to one of, say, *p* available classes ω_j, *j*=1,2,…,*p*. A neural network classifier can be used to produce 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 $p^i_{\omega_j}$ refers to the degree of coherence of $\underline{x}_i$ to class ω_j.

Let us first consider that a neural-network-based system has been created by a service provider and is being supplied to customers, so as to be used, for facial expression recognition; it is able to classify each image, or video shot of a human to one, or more specific categories, such as happy, angry or neutral. The neural classifier has obtained the necessary knowledge to perform the task, having been trained with a carefully designed and selected training set, say, $S_b = \left\{ \left(\underline{x}'_1, \underline{d}'_1 \right), \cdots, \left(\underline{x}'_{m_b}, \underline{d}'_{m_b} \right) \right\}$, where vectors $\underline{x}'_i$ and $\underline{d}'_i$ with $i = 1, 2, \cdots, m_b$ denote the *i*-th input and corresponding desired output vectors; this set being also provided to the customers. Let us then consider that a specific customer includes the system in his/her own PC and starts to use it, so as to have a more friendly interaction with it. Since the neural classifier includes all existing knowledge about facial anatomy and facial gestures, it will be able to analyze them so as to conclude about human specific expressions. It is, however, known that the system will be facing its true owner, and thus should adapt to its owner's specific characteristics and behavior, while keeping up with its former knowledge.

REFINEMENT / ADAPTATION PHASE

In this phase further training is performed so as to make the appropriate adjustments to the artefact in order to meet both its user peculiarities and any slowly changing conditions of the operating environment. This phase refers mainly to the adjustment of the $\mathbf{W_I}$ parameters to a more precise set $\mathbf{W_R}$ which corresponds better to the particular human user and which maps ***F'*** to ***S'***. This step requires the collection of a representative set T_R of user related pairs ($\underline{f}$, $\underline{s}$) to be used for refinement (retraining) of the network knowledge. Adaptation should be performed according to some constraints. Let p_I be a measure of the classification performance of the $\mathbf{W_I}$ network w.r.t training set T_I, p_R the corresponding measure of the $\mathbf{W_R}$ network w.r.t training set T_R, p_{RI} the performance measure of the $\mathbf{W_R}$ network w.r.t the initial training set T_I, p_{IR} the performance measure of the $\mathbf{W_I}$ network w.r.t the initial training set T_R. Then the following conditions should be true:

- $\left\| W_I - W_R \right\| < \varepsilon$, which allows only a small perturbation of the initial domain modeller. This is an absolute requirement since a radical changing to $\mathbf{W_I}$ parameters would lead to inefficient input to symbolic predicates mapping (ε is a small threshold).
- $p_R > p_{IR}$, which means that fine-tuning of the domain modeller should lead to a better performance w.r.t data related to its human user.
- $p_I \geq p_{RI}$, because $p_I < p_{RI}$ would imply that the initial training phase had not been sufficient and the further training has led to a better global mapping between inputs and symbolic predicates, and not to an adjustment to its particular user characteristics.
- $\left\| p_I - p_{RI} \right\| < \delta$, which ensures that the refinement will not significantly depreciate the performance of the domain modeller (δ is a small threshold).

In the case where the internal states *s* of the hybrid system are known and available data do exist adaptation/refinement is handled through the approach described below; otherwise the adaptation procedure requires the co-operation between the CAM and SPM modules.

Let us proceed with the first case: CAM adaptation through retraining. Let vector $\mathbf{W_R}$ include all weights of the network before retraining, and $\mathbf{W_I}$ the new weight vector which is obtained through retraining. A retraining set S_c is assumed to be extracted from the current operational situation composed of, say, m_c feature vectors; $S_c = \{(\underline{x}_1, \underline{d}_1), \cdots, (\underline{x}_{m_c}, \underline{d}_{m_c})\}$ where $\underline{x}_i$ and $\underline{d}_i$ with $i = 1,2,\cdots,m_c$ correspond to the *i*-th input and desired output retraining data. The retraining algorithm should compute the new network weights $\mathbf{W_I}$, by minimizing the following error criterion with respect to the weights,

$$E_a = E_{c,a} + \eta E_{f,a} \tag{2}$$

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

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 consisting of weights $\mathbf{W_I}$, corresponding to input vectors $\underline{x}_i$ and $\underline{x}'_i$ respectively. Similarly $\underline{z}_b(\underline{x}_i)$ would represent the output of the network, consisting of weights $\mathbf{W_R}$, when accepting vector $\underline{x}_i$ at its input. Parameter η is a weighting factor accounting for the significance of the current training set compared to the former one and $\|\circ\|_2$ denotes the L_2 -norm.

In most real life applications, training set S_c is initially unknown; consequently selection of S_c, as well as detection of the need to retrain should be provided to the system, either through user interaction, or automatically, when this is possible [1]. Each time that retraining is performed, new network weights are estimated taking into account both the current information (data in S_c) and the former knowledge (data in S_b). Further details regarding the retraining method can be found at [1], [2] and [3].

THE NEUROFUZZY ARCHITECTURE (SPM MODULE)

Fuzzy systems are numerical model-free estimators. While neural networks encode sampled information in a parallel-distributed framework, fuzzy systems encode structured, empirical (heuristic) or linguistic knowledge in a similar numerical framework. Although they can describe the operation of the system in natural language with the aid of human-like if-then rules, they do not provide the highly desired characteristics of learning and adaptation. The use of neural networks in order to realize the key concepts of a fuzzy logic system enriches the system with the ability of learning and improves the subsymbolic to symbolic mapping. Neural network realization of basic operations of fuzzy logic, such as fuzzy complement, fuzzy intersection and fuzzy union, can be implemented in terms of the activation function of neurons to provide fuzzy logic inference.

One of the widely used ways of constructing fuzzy inference systems is the method of approximate reasoning which can be implemented on the basis of compositional rule of inference. Different criteria have been proposed for the approximate reasoning to satisfy [4]. The most useful is that of the perfect recall. Fuzzy inference systems that satisfy the perfect recall criterion can be implemented with the aid max-min compositions of fuzzy relations[5]. The need for more general research lead to the representation of fuzzy inference systems on the basis of generalised Sup-t-norm and Inf-u-norm compositions [6].

Let us now proceed to a more detailed description of the neurofuzzy architecture. As previously explained the Sup-t and Inf-u compositions of fuzzy relations are the key issues of this network and generally of fuzzy set theory. This type of neuron is referred to as compositional neuron.

The general structure of a conventional neuron is described by the equation:

$y = a(\sum_{i=1}^{n} w_i x_i + \vartheta)$, where α is non-linearity, ϑ is threshold and wi are the weights that can change on-line with the aid of a learning process.

There are four types of composition neurons, the Sup-t, the Inf-u and the corresponding adjoints of them. The Sup-t and the Inf-u operators are used for forward direction (normal phase) and the corresponding adjoints for the backward direction (learning phase). We will only report the equations, which describe the neurons and not go further explaining the four operators/neurons [7].

The Sup-t compositional neuron has the same structure as the conventional neuron. It is described by the equation:

$y = a(\underset{j \in N_n}{Sup}\, t(x_i, w_i))$, where *t* is a fuzzy intersection operator (a t-norm) and a is the activation function.

The Inf-u compositional neuron is described by the equation:

$y = a(\underset{j \in N_n}{Inf}\, u(x_i, w_i))$, where *u* is a fuzzy union operator (an s-norm) and *a* is the activation function.

$$a(x)=\begin{cases}0, & x\in(-\infty,0)\\ x, & x\in[0,1]\\ 1, & x\in(0,+\infty)\end{cases}$$

which is widely used in neural networks

The proposed architecture is a two-layer neural network of compositional neurons. The first layer consists of the Inf-u neurons and the second layer consists of the Sup-t neurons. $W^1_{n\times k}$ is the weight matrix of the first layer and $W^2_{k\times m}$ is the weight matrix of the second layer(Figure 2).

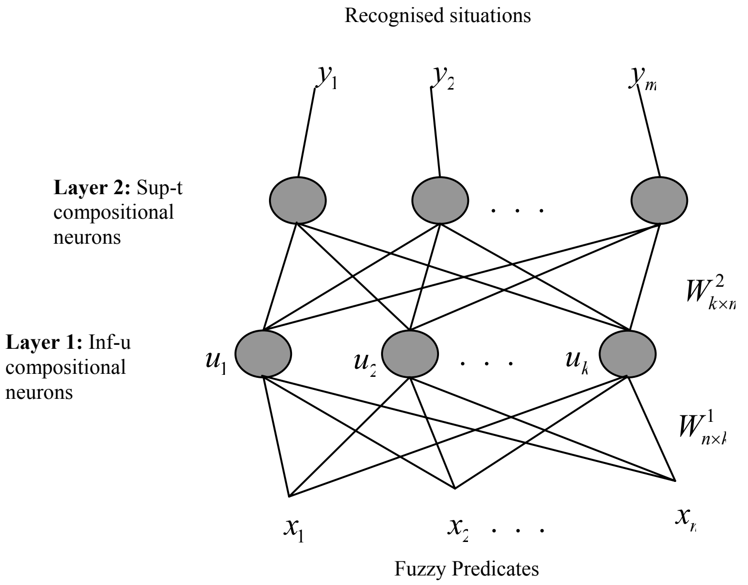


Figure 2: The two-layer neurofuzzy architecture

LEARNING ALGORITHM

Using a traditional minimisation algorithm (for example steepest descent) to implement learning in the network, we cannot take advantage of the specific character of the problem. Moreover, the nonlinear response of the compositional neuron could lead to the error local minimum. In this approach, we take advantage of this fact and propose a learning algorithm specialised in this type of neurofuzzy networks. The algorithm is based on a more sophisticated credit assignment. The problem of the credit assignment has been mentioned as the main problem of any learning algorithm. The proposed algorithm 'blames' the neurons of the network using the knowledge about the topographic structure of the neurofuzzy network. As explained before, the learning algorithm is based on the adjoints operators of the Sup-t and Inf-u dual operators. Each layer has its own learning algorithm. In layer 1, learning is implemented through the adjoint operator of the Inf-u operator. In layer two, the adjoint operator of Sup-t implements the learning process. The proposed learning algorithm converges independently for each neuron [7].

EXPERIMENTAL RESULTS

In order to validate the hybrid intelligence architecture we conducted an experimental study dealing with facial expression recognition. The basic motivation for examining this particular application stems from several studies for facial expression modelling, analysis and synthesis that are based on image/video features. For example, both FACS, through the Action Units (AU), and MPEG-4, through the use of the Facial Animation Parameters (FAP), use intermediate states to characterize facial expressions. Intermediate states refer to the fact that no low-level image/video features (pixel values, motion vectors, colour histograms) are used directly for modelling the expressions. Instead the estimation of AUs or FAPs based on image features is left to the particular implementation.

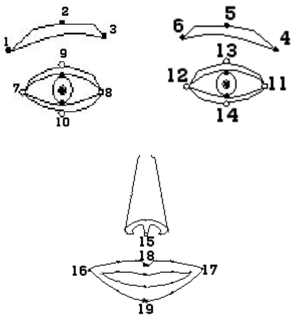


Figure 3: Facial Points used for the distances definition

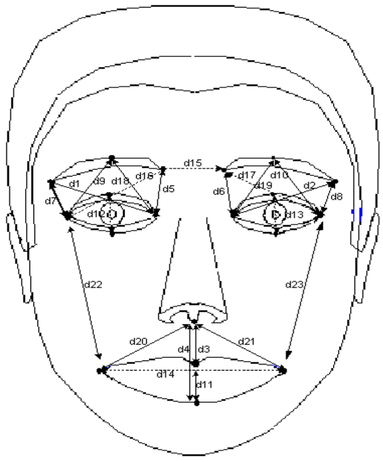


Figure 4: Facial Distances

EXPERIMENT SETUP

In the current study we consider as features (inputs to the CAM module) a set of 23 distances illustrated in Figure 4 and summarized in Table 1. As internal states (output of CAM-input to SPM) we used activation levels of FAPs that are related with facial expression formation (see Table 2). Finally, the output of the SPM module corresponds to the degree to which the observed image is related with a particular archetypal expression. The linguistic rules through which the facial expressions are described through FAPs can be found at [8]). We used the databases PHYSTA [9] for the training set and the EKMAN database [11] for the evaluation test. The coordinates of the points shown in Figure 3 have been marked by hand for 300 images in the training set and 110 images in the test set.

TRAINING THE CAM MODULE

CAM module is feedforward Neural Network which consists of 23 inputs, 324 hidden neurons and 48 outputs. For the training of this network we split it into 17 sub-networks each of which correspond to one FAP (see Table 2) Therefore, CAM consists of 17 independent NNs (actually FAPs cannot be considered totally independent but at this stage we made this adoption in order to deal with CAM output dimension).

Distances between facial points
d_1=d(p_1,p_8), d_2=d(p_4,p_{12}), d_3=d(p_{15},p_{18}), d_4=d(p_{15},p_{19}), d_5=d(p_3,p_8), d_6=d(p_6,p_{12}), d_7=d(p_1,p_7), d_8=d(p_4,p_{11}), d_9=d(p_2,p_7), d_{10}=d(p_5,p_{11}), d_{11}=d(p_{18},p_{19}), d_{12}=d(p_9,p_{10}), d_{13}=d(p_{13},p_{14}), d_{14}=d(p_{16},p_{17}), d_{15}=d(p_3,p_6), d_{16}=d(p_3,p_7), d_{17}=d(p_2,p_8), d_{18}=d(p_1,p_8), d_{19}=d(p_5,p_{12}), d_{20}=d(p_{15},p_{16}), d_{21}=d(p_{15},p_{17}), d_{22}=d(p_7,p_{16}), d_{23}=d(p_{11},p_{17})

Table 1: Facial distances used as features, *d(p_i,p_j)* is the Euclidean distance between facial points p_i and p_j - see also Figures 3, 4.

OPERATION EXAMPLE

Let us provide an example of the performance of the overall system. Input is *image001* of the Ekman database, showing happy expression. The feature vector, shown in Figure 5, express the deviation of the various distances, of Table 3, w.r.t the neutral case of the same person (*image006*). Values closed to one illustrate insignificant change of the corresponding distances. Figure 6 shows the output of the CAM module, which is translated as: *open_jaw*-> Medium, *lower_t_midlip*-> Low, *raise_b_midlip*->VeryLow, …, *raise_r_cornerlip_o*->High.

Figure 7, presents the activation level of each of the 41 rules that have been inserted in the SPM module, while Figure 8 shows the degree of belief that the observed, through the input vector, expression corresponds to the seven archetypal emotions (*Anger*->0, *Sadness*->0.12, *Joy*->0.64, *Disgust*->0, *Fear*->0, *Surprise*->0, *Neutral*->0).

Table 3 illustrates the confusion matrix of the mean degree of beliefs (*not the classification rates),* for each of the archetypal emotions *anger*, *joy*, *disgust*, *surprise* and the *neutral* condition, computed over the EKMAN dataset. We did not include the emotions *sadness* and *fear* due to the difficulty on constructing efficient rules for them based on the distances of Table 1. However, the output of the system provides degree of beliefs for sadness and fear also based on a few preliminary rules. It is observed that the expression *surprise* presents lower mean degree of belief than the other expressions. At a first glance this seems not reasonable since in the majority of studies that deal with expression recognition surprise is considered as the most recognizable emotion through the facial activity. However, in our approach rules describing surprise consist of several conditions (see for example row 3 in Table 4) that should be hold simultaneously. Failing of one condition leads to a lower degree of belief in the output. On the other hand, the same reasoning explains the fact that surprise cases are never misclassified (see column 5 of Table 3).

Table 4 shows the more often activated rule for each of the above expressions.

FAP name	*Primary distance*	*Other distances*	*States (VL-VeryLow, L-Low, M-Medium, H-High)*
Squeeze_l_eyebrow (F_{37})	d_2	d_6, d_8, d_{10}, d_{17}, d_{19}, d_{15}	L, M, H
Squeeze_r_eyebrow (F_{38})	d_1	d_5, d_7, d_9, d_{16}, d_{18}, d_{15}	L, M, H
Lower_t_midlip (F_4)	d_3	d_{11}, d_{20}, d_{21}	L, M
Raise_b_midlip (F_5)	d_4	d_{11}, d_{20}, d_{21}	VL, L, H
Raise_l_I_eyebrow (F_{31})	d_6	d_2, d_8, d_{10}, d_{17},d_{19}, d_{15}	L, M, H
Raise_r_I_eyebrow (F_{32})	d_5	d_1, d_7, d_9, d_{16}, d_{18}, d_{15}	L, M, H
Raise_l_o_eyebrow (F_{35})	d_8	d_2, d_6, d_{10}, d_{17}, d_{19}, d_{15}	L, M, H
Raise_r_o_eyebrow (F_{36})	d_7	d_1, d_5, d_9, d_{16}, d_{18}, d_{15}	L, M, H
Raise_l_m_eyebrow (F_{33})	d_{10}	d_2, d_6, d_8, d_{17}, d_{19}, d_{15}	L, M, H
Raise_r_m_eyebrow (F_{34})	d_9	d_1, d_5, d_7, d_{16}, d_{18}, d_{15}	L, M, H
Open_jaw (F_3)	d_{11}	d_4	L, M, H
close_left_eye (F_{19}, F_{21})	d_{13}	-	L, H
close_right_eye (F_{20}, F_{22})	d_{12}	-	L, H
Wrinkles_between_eyebrows (F_{37}, F_{38})	d_{15}	d_1, d_2, d_5, d_6, d_7, d_8, d_9, d_{16}, d_{17}, d_{18}, d_{19}	L, M, H
Raise_l_cornerlip_o (F_{53})	d_{23}	d_3, d_4, d_{11}, d_{20}, d_{21}, d_{22}	L, M, H
Raise_r_cornerlip_o (F_{54})	d_{22}	d_3, d_4, d_{11}, d_{20}, d_{21}, d_{23}	L, M, H
widening_mouth (F_6, F_7)	d_{11}	d_3, d_4, d_{14}	L, M, H

Table 2: Training the CAM module: Each row corresponds to a feedforward NN; therefore the CAM consists of 17 NNs each of which has less than 10 inputs (distances), less than 30 hidden neurons and less than 4 outputs (states)

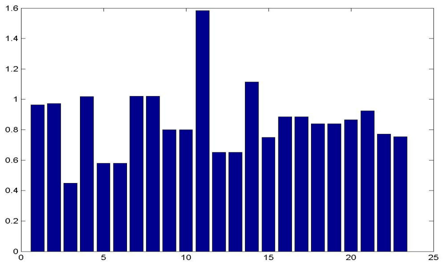


Figure 5: Example of a feature vector feeding the CAM

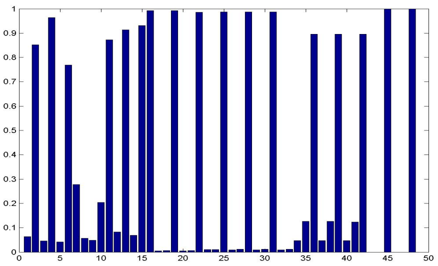


Figure 6: An instance of CAM's output

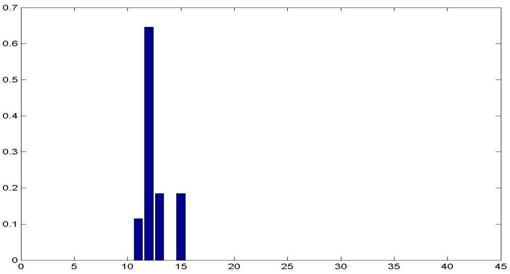


Figure 7: Activation of each of the 41 rules

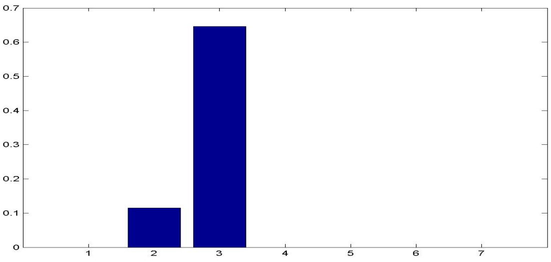


Figure 8: The output of SPM

	Anger	**Joy**	**Disgust**	**Surprise**	**Neutral**
Anger	0.611	0.01	0.068	0	0
Joy	0.006	0.757	0.009	0	0.024
Disgust	0.061	0.007	0.635	0	0
Surprise	0	0.004	0	0.605	0.001
Neutral	0	0.123	0	0	0.83

Table 3: Results in images of different expressions

Expressions	Rule more often activated (% of examined photos)
Anger	[*open_jaw_low, lower_top_midlip_medium, raise_bottom_midlip_high, raise_left_inner_eyebrow_low, raise_right_inner_eyebrow_low, raise_left_medium_eyebrow_low, raise_right_medium_eyebrow_low, squeeze_left_eyebrow_high, squeeze_right_eyebrow_high, wrinkles_between_eyebrows_high, raise_left_outer_cornerlip_medium, raise_right_outer_cornerlip_medium*] (47%)
Joy	[*open_jaw_high, lower_top_midlip_low, raise_bottom_midlip_verylow, widening_mouth_high, close_left_eye_high, close_right_eye_high*] (39%)
Disgust	[*open_jaw_low, lower_top_midlip_low, raise_bottom_midlip_high, widening_mouth_low, close_left_eye_high, close_right_eye_high, raise_left_inner_eyebrow_medium, raise_right_inner_eyebrow_medium, raise_left_medium_eyebrow_medium, raise_right_medium_eyebrow_medium, wrinkles_between_eyebrows_medium*] {33%)
Surprise	[*open_jaw_high, raise_bottom_midlip_verylow, widening_mouth_low, close_left_eye_low, close_right_eye_low, raise_left_inner_eyebrow_high, raise_right_inner_eyebrow_high, raise_left_medium_eyebrow_high, raise_right_medium_eyebrow_high, raise_left_outer_eyebrow_high, raise_right_outer_eyebrow_high, squeeze_left_eyebrow_low, squeeze_right_eyebrow_low, wrinkles_between_eyebrows_low*] (71%)
Neutral	[*open_jaw_low, lower_top_midlip_medium, raise_left_inner_eyebrow_medium, raise_right_inner_eyebrow_medium, raise_left_medium_eyebrow_medium, raise_right_medium_eyebrow_medium, raise_left_outer_eyebrow_medium, raise_right_outer_eyebrow_medium, squeeze_left_eyebrow_medium, squeeze_right_eyebrow_medium, wrinkles_between_eyebrows_medium, raise_left_outer_cornerlip_medium, raise_right_outer_cornerlip_medium*] (70%)

Table 4: Activated rules

CONCLUSIONS

This work aimed at presenting the concept of a hybrid intelligence architecture for the facial expression recognition application. The facial expression recognition task is by no means trivial and it cannot be tackled using image/video data and especially distances within the human face. Texture, motion and paralinguistic measures are also required for this purpose [10]. However, we have shown that the proposed architecture is appropriate for subsymbolic to symbolic mapping, which in future HCI applications that will be based on emotion understanding, will be critical. As far as the facial expression recognition application is concerned our proposal stills be valuable. It is easy to add more rules, enhance the low level features that are currently used to describe the rules (i.e., include facial texture, facial feature movement, gestures etc.) beyond facial distances, or add other modalities (for example, those that are based on speech, physical measurements) etc. The authors are currently working towards this direction in the framework of the ERMIS project (http://www.image.ntua.gr/ermis).

REFERENCES:

[1]. 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.

[2]. N. Doulamis, A. Doulamis, G. Votsis, N. Tsapatsoulis and S. Kollias, "An Adaptive Framework for Multimedia Information Retrieval and Emotionally Rich Human Computer Interaction," *IEEE Transactions Multimedia*, submitted 2001.

[3]. D. C. Park, M. A. EL-Sharkawi, and R. J. Marks II, "An Adaptively Trained Neural Network," *IEEE Trans. on Neural Networks,* vol. 2, pp. 334-345, 1991.

[4]. S. Tzafestas, S. Raptis, G. B. Stamou, "A general neurofuzzy cell structure for fuzzy inference", *Math. Comp. On Fuzzy systems,* vol. 2, pp 956-960, San Francisco.

[5]. A. Di Nola, W. Pedrycz, S. Sessa, E. Sanchez, "Fuzzy relational equations and their applications to knowledge engineering", *Kluwer Academic Publishers*, Dordrecht, 1989.

[6]. Y. Akiyama, T. Abe, T. Mitsunaga, H. Koga, " A conceptual study of max-* composition on the correspondence of base spaces and its applications in determining fuzzy relations", *Japanese journal of fuzzy theory and systems,* vol. 2, pp. 113-132, 1991.

[7]. G. B. Stamou, S. G. Tzafestas, "Neural fuzzy relational systems with a new learning algorithm", *Mathematics and computers in simulation,* pp. 301-304, 2000.

[8]. N. Tsapatsoulis, A. Raouzaiou, S. Kollias, R. Cowie and E. Douglas-Cowie, "Emotion Recognition and Synthesis based on MPEG-4 FAPs," in *MPEG-4 Facial Animation*, Igor Pandzic, R. Forchheimer (eds), John Wiley & Sons, UK, 2002.

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

[10]. 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, January 2001.

[11]. P. Ekman and W. Friesen, *The Facial Action Coding System*, Consulting Psychologists Press, San Francisco, CA, 1978 (*http://www.paulekman.com)*