A Lattice Computing Extension of the FAM Neural Classifier for Human Facial Expression Recognition

Vassilis G. Kaburlasos[a], Stelios E. Papadakis[a,b], and George A. Papakostas[a]
[a]Human-Machines Interaction (HMI) Laboratory
Dept. of Industrial Informatics
TEI of Kavala, 65404 Kavala, Greece
[b]Dept. of Business Planning and Information Systems
TEI of Crete, 72100 Agios Nikolaos, Greece
Emails: vgkabs@teikav.edu.gr, spap@staff.teicrete.gr, gpapak@teikav.edu.gr

***Abstract*—This paper proposes a fundamentally novel extension, namely flrFAM, of the fuzzy ARTMAP (FAM) neural classifier for incremental real-time learning and generalization based on fuzzy lattice reasoning (FLR) techniques. FAM is enhanced, first, by a *parameter optimization* training (sub)phase and, second, by a capacity to process *partially ordered* (non)numeric data including information granules. The interest here focuses on Intervals' Numbers (INs) data, where an IN represents a distribution of data samples. We describe the proposed flrFAM classifier as a fuzzy neural network that can induce descriptive as well as *flexible* (i.e., *tunable*) decision-making knowledge (rules) from the data. This work demonstrates the capacity of the flrFAM classifier for human facial expression recognition on benchmark datasets. A novel feature extraction and knowledge-representation is based on orthogonal moments. The reported experimental results compare well with the results by alternative classifiers from the literature. The far reaching potential of FLR in Human-Machine Interaction (HMI) applications is discussed.**

***Index Terms* – Fuzzy ARTMAP, fuzzy lattice reasoning, inclusion measure, intervals' number, the lattice computing paradigm**

I. INTRODUCTION

The employment of a computational model for learning is often based on simplifying (non-realistic) assumptions, including abundant/representative data, fixed data distributions and independent data samples in order to enable rigorous analysis and design. However, far more often than not, the previous assumptions do not hold in practical applications such as climate/financial modeling, electricity demand, human-machine interaction, etc. Hence, alternative modeling approaches emerged including (concept) drift models and domain adaptation algorithms, which may engage incremental-learning and/or online-learning [12], [37]. Nevertheless, an alternative modeling approach still makes (heuristic) assumptions such as restrictive types of distributions, moreover it is typically restricted in the Euclidean space $\mathfrak{R}^N$. Against this background, there is a need for general architectures crafted in a versatile framework to enable learning from – and adapting to – an ever changing environment.

This work proposes a straightforward extension of the established *fuzzy ARTMAP* (*FAM*) neural classifier [5], [7] for incremental, on-line learning and classification of nonstationary data based on fuzzy lattice reasoning (FLR) techniques [29] in the context of the versatile lattice theory [3] – Recall that "FLR" has been defined as decision-making based on an *inclusion measure* function [25]. In particular, we extend FAM's application domain from the unit hypercube in $\mathfrak{R}^N$, where learning is pursued by inducing hyperboxes, to a general (mathematical) lattice. In conclusion, the *flrFAM classifier* emerges here for learning by inducing intervals in a general lattice including the induction of hyperboxes in the unit hypercube as a special case. An implied advantage is the widening of FAM's scope so as to deal with data semantics represented by partial order. Additional advantages for the flrFAM classifier are summarized in the following.

The proposed flrFAM classifier can learn rare patterns by addressing the "stability-plasticity" dilemma the same way as FAM does – Recall that the aforementioned dilemma states that "(a system) must be capable of plasticity in order to learn about significant new events, yet it must also remain stable in response to irrelevant or often repeated events" [4]. Moreover, the flrFAM classifier can carry out *granular computing* by processing lattice-ordered (information) granules [46] instead of processing merely *points* in $\mathfrak{R}^N$; the latter (points) are exclusively processed by FAM in the unit hypercube. Furthermore, in every data dimension, only the flrFAM classifier may optimize a tunable positive valuation (weight) function towards improving performance.

The basic "decision-making" instrument of the flrFAM classifier is an *inclusion measure* function $\sigma(.,.)$, which corresponds to both FAM's *choice (Weber)* and *match* functions as explained below. Note that, historically, inclusion measures of the form $\sigma(A,B)$ have been introduced for computing the degree of inclusion of a hyperbox A into another one B in classification applications [20]. It was then realized that the set of hyperboxes in $\mathfrak{R}^N$ is lattice-ordered; this fact has been the motivation to extend the hyperbox based approach for learning/generalization to a general lattice data domain [22].

The interest of this work is in Intervals' Numbers (INs) data, where an IN represents a distribution of samples. An IN may also be thought of as the "α-cuts representation" of a fuzzy number. In all, an IN is a mathematical object which can be interpreted either probabilistically or possibilistically [41]. INs, previously called FINs, have been studied in a series of publications. In particular, it has been shown that the set $\mathfrak{F}_1$ of INs is a (metric) lattice [21], [31] with cardinality $\aleph_1$ [24], where "$\aleph_1$" is the cardinality of the set $\mathfrak{R}$ of real numbers; moreover, the space $\mathfrak{F}_1$ is a cone in a linear space [26], [41]. INs have already been used in numerous (classification and regression) applications [21], [24], [26], [27], [30], [41] as well as for hybrid intelligence fusion [25]. Our interest here is in an flrFAM classifier application on the lattice $(\mathfrak{F}_2, \preceq)$ of Type-2 *Intervals' Numbers* as detailed below.

From an application point of view, this work focuses on a specific human-machine interaction (HMI) problem, namely human facial expression recognition. Note that a number of learning models have been proposed in human-centered recognition applications [2], [8]. Currently, static/dynamic facial expression recognition is carried out at large by "number crunching" machine learning techniques [39]. The flrFAM classifier here suggests a viable alternative for *flexible* (i.e., *tunable*) rule-based classification with a considerable potential for sound (non)numeric data fusion.

An "agglomerative" FLR classifier has been reported lately for human facial expression recognition and applied exclusively on the JAFFE benchmark [42]. Substantial differences with the work here include: First, this work details constructively a six-level hierarchy of mathematical lattices, whereas the work in [42] engages only part of the aforementioned hierarchy. Second, the work in [42] delineates an agglomerative FLR learning scheme only for structure identification such that one Type-1 IN is induced (unconditionally) per class; whereas, this work details sophisticated extensions of the FAM classifier architecture for structure identification followed by parameter optimization such that multiple Type-2 INs may be induced (conditionally) per class. Third, the work in [42] assumes one 100-dimensional features (moments) vector represented by one (non-trivial) Type-1 IN, furthermore it employs seven random data partitions for training/testing; whereas, this work assumes one 16-dimensional features (moments) vector represented by a (trivial) Type-2 IN, furthermore it employs ten random data partitions for training/testing. Fourth, the work in [42] carries out computational experiments in space $\mathfrak{F}_1^1$ engaging only two classifiers, namely *(agglomerative) FLR* and *kNN*; whereas, this work carries out computational experiments in both spaces $\mathfrak{F}_2^6$ and $\mathfrak{F}_2^{16}$ engaging seven classifiers, namely *flrFLR*, *kNN*, *LDA*, *naive Bayes*, *classification tree*, a *neural network* and *FAM* as detailed below; furthermore, the flrFAM classifier here is applied, in addition, on the RADBOUD benchmark; moreover, only this work presents statistical testing results. Fifth, only this work presents an extensive literature review with novel perspectives including an introduction of the lattice computing (LC) paradigm.

The paper is organized as follows. Section II presents a hierarchy of mathematical lattices including Intervals' Numbers (INs). Section III details the flrFAM extension of the FAM classifier. Section IV describes the human facial expression recognition problem in context. Section V presents comparative computational experiments on benchmark datasets and results including a discussion of significance. Section VI concludes by summarizing our contribution and future work.

II. A Hierarchy of Lattices

This section introduces constructively, in six steps, a hierarchy of complete lattices; in particular, each subsection presents an ever enhanced (lattice) hierarchy level. For general lattice theory notions including the definition of an *inclusion measure* function the reader may refer elsewhere [25], [30].

Assume a positive valuation[1] function $v : \mathfrak{L} \to [0, \infty)$ on a complete lattice $(\mathfrak{L}, \sqsubseteq)$ with least and greatest element O and I, respectively, such that $v(O) = 0$ and $v(I) < \infty$. Assume functions *sigma-meet* $\sigma_\sqcap : \mathfrak{L} \times \mathfrak{L} \to [0, 1]$ and *sigma-join* $\sigma_\sqcup : \mathfrak{L} \times \mathfrak{L} \to [0, 1]$ defined as follows

$$\sigma_\sqcap(x, y) = \begin{cases} 1, & \text{for } x = O \\ \frac{v(x \sqcap y)}{v(x)}, & \text{for } x \sqsupset O \end{cases} \tag{1}$$

$$\sigma_\sqcup(x, y) = \begin{cases} 1, & \text{for } x \sqcup y = O \\ \frac{v(y)}{v(x \sqcup y)}, & \text{for } x \sqcup y \sqsupset O \end{cases} \tag{2}$$

Then, both $\sigma_\sqcap(.,.)$ and $\sigma_\sqcup(.,.)$ are inclusion measures. Note that an inclusion measure function $\sigma : \mathfrak{L} \times \mathfrak{L} \to [0, 1]$ can be interpreted as a *fuzzy order* relation on a lattice $(\mathfrak{L}, \sqsubseteq)$. Hence, notations $\sigma(x, y)$ and $\sigma(x \sqsubseteq y)$ will be used interchangeably.

A. Real Numbers

The set $\mathfrak{R}$ of real numbers is a *totally-ordered*, *non-complete* lattice denoted by $(\mathfrak{R}, \leq)$, where "$\leq$" is the usual order relation of real numbers. Lattice $(\mathfrak{R}, \leq)$ can be extended to a complete lattice by including both symbols "$-\infty$" and "$+\infty$". In conclusion, the complete lattice $(\overline{\mathfrak{R}}, \leq)$ emerges, where $\overline{\mathfrak{R}} = \mathfrak{R} \cup \{-\infty, +\infty\}$, with least and greatest elements $O = -\infty$ and $I = +\infty$, respectively.

In the context of this work we will employ, in particular, a *reference set* $\mathfrak{L} \subseteq \overline{\mathfrak{R}}$ so that the totally ordered lattice $(\mathfrak{L}, \leq)$ is complete. For example, $\mathfrak{L}$ can be either $\overline{\mathfrak{R}}$ itself or a *closed* interval $[a, b] \subset \overline{\mathfrak{R}}$. In every case, $\mathfrak{L}$ includes a least element denoted by O and a greatest element denoted by I (hence $\mathfrak{L} = [O, I]$). For example, for $\mathfrak{L} = \overline{\mathfrak{R}}$ it is $O = -\infty$ and $I = +\infty$; whereas, for $\mathfrak{L} = [a, b]$ it is $O = a$ and $I = b$. The *inf* and *sup* operations in the complete lattice $(\mathfrak{L}, \leq)$ are denoted by $\wedge$ and $\vee$. Any strictly increasing function $v : \mathfrak{L} \to [0, \infty)$ is a positive valuation on $(\mathfrak{L}, \leq)$, moreover any strictly decreasing function $\theta : \mathfrak{L} \to \mathfrak{L}$ is dual isomorphic[2] on the complete lattice $(\mathfrak{L}, \leq)$. In this work, we consider *bijective* (one-to-one) functions $\theta : \mathfrak{L} \to \mathfrak{L}$ such that both $\theta(O) = I$ and $\theta(I) = O$; moreover, we consider positive valuation functions $v : \mathfrak{L} \to [0, \infty)$ such that both $v(O) = 0$ and $v(I) < \infty$.

B. Type-1 Intervals

Consider the complete lattice $(\mathfrak{I}_1, \subseteq)$ of *Type-1 intervals* $[a, b]$, or *intervals* for short, on a complete lattice $(\mathfrak{L}, \leq)$ of real numbers with least and greatest elements O and I,

[1] *Positive valuation* on a lattice $(\mathfrak{L}, \sqsubseteq)$ is a real function $v : \mathfrak{L} \to \mathfrak{R}$ that satisfies both $v(x) + v(y) = v(x \sqcap y) + v(x \sqcup y)$ and $x \sqsubset y \Rightarrow v(x) < v(y)$.

[2] Let $(\mathfrak{K}, \sqsubseteq)$ and $(\mathfrak{L}, \sqsubseteq)$ be lattices. A function $\theta : \mathfrak{K} \to \mathfrak{L}$ here is called *dual isomorphic* iff both "$x \sqsubset y \Leftrightarrow \theta(x) \sqsupset \theta(y)$" and "$\theta$ onto $\mathfrak{L}$".

respectively. Recall that an interval is defined as $[a,b] \doteq \{x : a \le x \le b\}$. Moreover,

$$[a,b] \cap [c,d] = [a \vee c, b \wedge d] \text{ and } [a,b] \dot{\cup} [c,d] = [a \wedge c, b \vee d]$$

Note that if $a \vee c > b \wedge d$ then $[a \vee c, b \wedge d] = \emptyset$; in words, if $a \vee c > b \wedge d$ then we assume that the intersection $[a,b] \cap [c,d]$ is the empty set ($\emptyset$). We remark from [22] that a preferable (in computing) representation for the least element $O_{\mathfrak{I}1} = \emptyset$ in lattice $(\mathfrak{I}_1, \subseteq)$ is $O_{\mathfrak{I}1} = [I, O]$.

Consider a (strictly increasing) positive valuation function $v : \mathfrak{L} \to [0, \infty)$, furthermore consider a (strictly decreasing) dual isomorphic function $\theta : \mathfrak{L} \to \mathfrak{L}$. Then, function $v_1 : \mathfrak{L} \times \mathfrak{L} \to [0, \infty)$ given by $v_1([a,b]) = v(\theta(a)) + v(b)$ is a positive valuation on lattice $(\mathfrak{L} \times \mathfrak{L}, \ge \times \le)$ [25]. Furthermore, based on equations (1) and (2) two inclusion measures $\sigma_\cap : \mathfrak{I}_1 \times \mathfrak{I}_1 \to [0,1]$ and $\sigma_{\dot{\cup}} : \mathfrak{I}_1 \times \mathfrak{I}_1 \to [0,1]$ can be introduced by $\sigma_\cap(x,y) = \sigma_\sqcap(x, x \cap y)$ and $\sigma_{\dot{\cup}}(x,y) = \sigma_\sqcup(x,y)$, respectively, on the complete lattice $(\mathfrak{I}_1, \subseteq)$ as it will be shown elsewhere.

Functions $\theta(.)$ and $v(.)$ can be selected in different ways. In the context of this work, we select a pair of functions $v(x)$ and $\theta(x)$ so as to satisfy equality "$v_1([x,x]) = v(\theta(x)) + v(x) = Constant$" required by a "standard" fuzzy lattice reasoning (FLR) scheme [25], [28], [29]. For instance, such pairs of functions $v(x)$ and $\theta(x)$ include, first, $v(x) = px$ and $\theta(x) = Q - x$, where $p, Q > 0$, $x \in [0, Q]$ and, second, $v_s(x) = \frac{A}{1 + e^{-\lambda(x-\mu)}}$ and $\theta(x) = 2\mu - x$, where $A, \lambda \in \mathfrak{R}_0^+$, $\mu, x \in \mathfrak{R}$. In pariticular, it follows, first, $v_1([x,x]) = pQ$ and, second, $v_1([x,x]) = A$, respectively.

C. Type-2 Intervals

A *Type-2 interval* is defined as an interval of Type-1 intervals. Consider the complete lattice $(\mathfrak{I}_2, \subseteq)$ of Type-2 intervals on a complete lattice $(\mathfrak{L}, \le)$ of real numbers with least and greatest elements O and I, respectively. Recall that $[[a_1,a_2],[b_1,b_2]] \cap [[c_1,c_2],[d_1,d_2]] =$

$$[[a_1,a_2] \dot{\cup} [c_1,c_2], [b_1,b_2] \cap [d_1,d_2]], \text{ and}$$

$[[a_1,a_2],[b_1,b_2]] \dot{\cup} [[c_1,c_2],[d_1,d_2]] =$

$$[[a_1,a_2] \cap [c_1,c_2], [b_1,b_2] \dot{\cup} [d_1,d_2]].$$

We remark that a preferable representation for the least element $O_{\mathfrak{I}2} = \emptyset$ in lattice $(\mathfrak{I}_2, \subseteq)$ is $O_{\mathfrak{I}2} = [[O,I],[I,O]]$.

Consider a (strictly increasing) positive valuation function $v : \mathfrak{L} \to [0, \infty)$ as well as a (strictly decreasing) dual isomorphic function $\theta : \mathfrak{L} \to \mathfrak{L}$. Recall that function $v_1 : \mathfrak{L} \times \mathfrak{L} \to [0, \infty)$ given by $v_1(a,b) = v(\theta(a)) + v(b)$ is a positive valuation. Furthermore, function $\theta_1 : \mathfrak{L} \times \mathfrak{L} \to \mathfrak{L} \times \mathfrak{L}$ given by $\theta_1(a,b) = (b,a)$ is dual isomorphic. Therefore, function $v_2 : \mathfrak{L} \times \mathfrak{L} \times \mathfrak{L} \times \mathfrak{L} \to [0, \infty)$ given by $v_2([[a_1,a_2],[b_1,b_2]]) = v(a_1) + v(\theta(a_2)) + v(\theta(b_1)) + v(b_2)$ is a positive valuation on lattice $(\mathfrak{L} \times \mathfrak{L} \times \mathfrak{L} \times \mathfrak{L}, \le \times \ge \times \ge \times \le)$. In conclusion, based on (1) and (2) inclusion measures $\sigma_\cap : \mathfrak{I}_2 \times \mathfrak{I}_2 \to [0,1]$ and $\sigma_{\dot{\cup}} : \mathfrak{I}_2 \times \mathfrak{I}_2 \to [0,1]$ can be introduced by $\sigma_\cap(x,y) = \sigma_\sqcap(x, x \cap y)$ and $\sigma_{\dot{\cup}}(x,y) = \sigma_\sqcup(x,y)$, respectively, on the complete lattice $(\mathfrak{I}_2, \subseteq)$ of Type-2 intervals.

D. Type-1 Intervals' Numbers (INs)

Consider the following definition.

Definition 2.1: A *Type-1 Intervals' Number (IN)* is a function $F : [0,1] \to \mathfrak{I}_1$ which satisfies

$$\begin{aligned} &F(0) = I_{\mathfrak{I}1}, \\ &\quad h_1 \le h_2 \Rightarrow F(h_1) \supseteq F(h_2), \\ &\quad \forall P \subseteq [0,1] : \cap_{h \in P} F(h) = F\left(\bigvee P\right). \end{aligned}$$

We will denote the set of INs by $\mathfrak{F}_1$ and equip it with an order relationship $\preceq$ such that $F \preceq G \Leftrightarrow (\forall h \in [0,1] : F(h) \subseteq G(h))$. Furthermore, we will denote an IN by a capital letter in italics, e.g. $\mathfrak{F}_1 \ni F = F(h) = [a_h, b_h]$, $h \in [0,1]$. In practice, an IN is interpreted as an information *granule*. It turns out that $(\mathfrak{F}_1, \preceq)$ is a complete lattice whose least element $\emptyset$ is preferably represented as $O_{\mathfrak{F}1} = O(h) = [I, O]$, $h \in [0,1]$.

Definition 2.1 implies that an IN can be represented by a set of intervals; that is, its *interval-representation*. In addition, an IN can, equivalently, be represented by a membership function; that is, the *membership-function-representation* [25].

E. Type-2 Intervals' Numbers (INs)

Another information *granule* of interest is an interval $[U, W]$ of Type-1 INs U and W, where interval $[U, W]$ by definition equals $[U, W] \doteq \{X \in \mathfrak{F}_1 \colon U \preceq X \preceq W\}$. In the latter sense we say that X is *encoded* in $[U, W]$. Interval $[U, W]$ is called *Type-2 IN*. It follows the complete lattice $(\mathfrak{F}_2, \preceq)$ of Type-2 INs. We remark that the least (empty) interval $\emptyset$ is preferably represented in computing as $O_{\mathfrak{F}2} = O(h) = [[O,I],[I,O]]$, where $h \in [0,1]$. A Type-2 IN will be denoted by a double-line capital letter, e.g. $\mathbb{F} \in \mathfrak{F}_2$.

The lattice $(\mathfrak{F}_2, \preceq)$ *join* operation is demonstrated in Fig.1. In particular, Fig.1(a) shows trivial Type-2 INs $\mathbb{C}_1 = [C_1, C_1]$, $\mathbb{C}_2 = [C_2, C_2]$ and $\mathbb{C}_3 = [C_3, C_3]$. Fig.1(b) displays the join $\mathbb{C}_1 \curlyvee \mathbb{C}_2 = [C_1 \curlywedge C_2, C_1 \curlyvee C_2]$ in its membership-function-representation. Note that, since Type-1 INs C_1 and C_2 overlap, the Type-1 IN $C_1 \curlywedge C_2$ is not empty. More specifically, it is $(C_1 \curlywedge C_2)(h) \neq \emptyset$, for $h \in [0, 0.6471]$; nevertheless, for $h \in (0.6471, 1]$, it is $(C_1 \curlywedge C_2)(h) = \emptyset$. Fig.1(c) displays the join $\mathbb{C}_1 \curlyvee \mathbb{C}_2$ in the (equivalent) interval-representation. Fig.1(d) displays the join $\mathbb{C}_2 \curlyvee \mathbb{C}_3 = [C_2 \curlywedge C_3, C_2 \curlyvee C_3]$ in its membership-function-representation. Note that, since Type-1 INs C_2 and C_3 do not overlap, the Type-1 IN $C_2 \curlywedge C_3$ is empty, that is $(C_2 \curlywedge C_3)(h) = \emptyset$, for all $h \in [0,1]$.

We point out that there are similarities as well as differences between Type-1/2 INs and Type-1/2 fuzzy sets [30].

Our interest here focuses on inclusion measure $\sigma_{\dot{\curlyvee}} : \mathfrak{F}_2 \times \mathfrak{F}_2 \to [0,1]$ given as [22]

$$\sigma_{\dot{\curlyvee}}(\mathbb{E}_1, \mathbb{E}_2) = \int_0^1 \sigma_{\dot{\cup}}(\mathbb{E}_1(h), \mathbb{E}_2(h))dh \tag{3}$$

F. Extensions to More Dimensions

An N-tuple IN of Type-1/2 will be indicated by an "over right arrow". More specifically, a Type-1 IN will be denoted

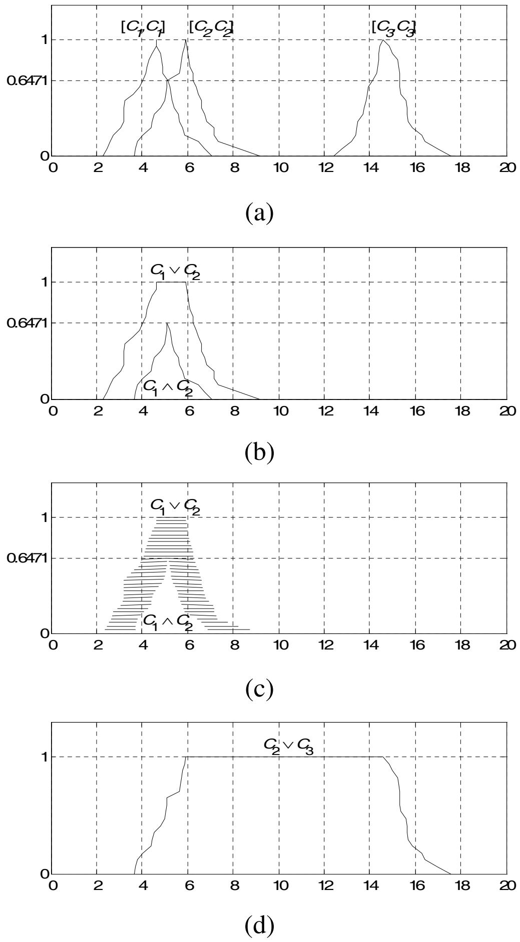


Fig. 1. Demonstrating the lattice *join* ($\curlyvee$) operation between trivial Type-2 INs. (a) Trivial Type-2 INs $[C_1, C_1] = \mathbb{C}_1$, $[C_2, C_2] = \mathbb{C}_2$ and $[C_3, C_3] = \mathbb{C}_3$. (b) Type-2 IN $\mathbb{C}_1 \curlyvee \mathbb{C}_2 = [C_1 \curlywedge C_2, C_1 \curlyvee C_2]$ is shown in its membership-function-representation. (c) Type-2 IN $\mathbb{C}_1 \curlyvee \mathbb{C}_2 = [C_1 \curlywedge C_2, C_1 \curlyvee C_2]$ is shown again, this time in its (equivalent) interval-representation for $L = 32$ different levels spaced uniformly over the interval $[0, 1]$ on the vertical axis. (d) Type-2 IN $\mathbb{C}_2 \curlyvee \mathbb{C}_3 = [C_2 \curlywedge C_3, C_2 \curlyvee C_3] = [\emptyset, C_2 \curlyvee C_3]$.

by $\overrightarrow{E} = (E_1, \ldots, E_N) \in (\mathfrak{F}_1^N, \preceq)$, whereas a Type-2 IN will be denoted by $\overrightarrow{\mathbb{E}} = (\mathbb{E}_1, \ldots, \mathbb{E}_N) \in (\mathfrak{F}_2^N, \preceq)$.

The previous has shown how to define inclusion measure functions on lattice $(\mathfrak{F}_2, \preceq)$. The latter functions can be extended to the product lattice $(\mathfrak{F}_2^N, \preceq)$ by inclusion measure function $\sigma_\wedge : \mathfrak{L} \times \mathfrak{L} \rightarrow [0, 1]$ given as follows

$$\sigma_\wedge((x_1, \ldots, x_N), (y_1, \ldots, y_N)) = \min_{i \in \{1,\ldots,N\}} \sigma_i(x_i, y_i) \quad (4)$$

III. A Fuzzy Lattice Reasoning (FLR) Extension of the FAM Neural Classifier

This section details the flrART scheme for clustering followed by the flrFAM scheme for classification.

A. The flrART Scheme for Clustering

Fig.2 displays the flrART neural architecture for clustering in lattice $(\mathfrak{I}_1^N, \subseteq)$ inspired from *fuzzy ART* [6].

Algorithm 1 describes the flrART scheme for clustering in the interval lattice data domain $(\mathfrak{I}_1^N, \subseteq)$.

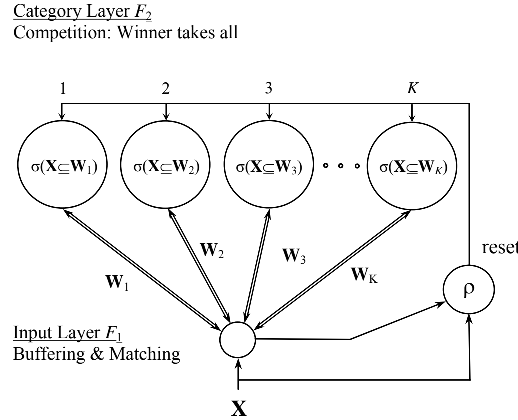


Fig. 2. The flrART neural architecture for clustering, where an input pattern $\mathbf{X}$ is in the lattice $(\mathfrak{I}_1^N, \subseteq)$ of intervals.

Algorithm 1 flrART Clustering

1: Assume a set $C \subset 2^{\mathfrak{I}_1^N}$; $K = |C|$; a user-defined *vigilance parameter* $\rho \in [0, 1]$;
2: **for** $i = 1$ to $i = n$ **do**
3:   Consider the next input datum $\mathbf{X}_i \in \mathfrak{I}_1^N$;
4:   $S \doteq C$;
5:   $J \doteq \underset{\substack{j \in \{1,\ldots,|S|\} \\ \mathbf{W}_j \in S}}{\texttt{argmax}} \{\sigma(\mathbf{X}_i \subseteq \mathbf{W}_j)\}$;
6:   **while** $(S \neq \{\})$.and.$(\sigma(\mathbf{W}_J \subseteq \mathbf{X}_i) < \rho)$ **do**
7:     $S \doteq S \setminus \{\mathbf{W}_J\}$;
8:     $J = \underset{\substack{j \in \{1,\ldots,|S|\} \\ \mathbf{W}_j \in S}}{\texttt{argmax}} \{\sigma(\mathbf{X}_i \subseteq \mathbf{W}_j)\}$;
9:   **end while**
10:   **if** $S = \{\}$ **then**
11:     $C \doteq C \cup \{\mathbf{X}_i\}$;
12:     $K \doteq K + 1$;
13:   **else**
14:     $\mathbf{W}_J \doteq \mathbf{W}_J \,\dot{\cup}\, \mathbf{X}_i$;
15:   **end if**
16: **end for**

The complexity of Algorithm 1 is determined by its two (nested) loops: The outer (**for**) loop repeats exactly n times such that, each time, the inner (**while**) loop repeats $O(n)$ times. Hence, the complexity of the flrART scheme for clustering is quadratic $O(n^2)$ in the number n of the input data.

Algorithm 1 is an extension of *fuzzy ART* [6] as explained in the following. An interval $\mathbf{W}_i \in \mathfrak{I}_1^N$, where $i \in \{1, \ldots, K\}$ corresponds to a "category" of *fuzzy ART*. Moreover, in *fuzzy ART*'s terminology, the set S holds all the "set" categories. Competition among the "set" categories takes place in step 5, as well as in step 8, where the index J of the winner category is computed. In particular, flrART's function $\sigma(\mathbf{X}_i \subseteq \mathbf{W}_j)$ corresponds to *fuzzy ART*'s *choice (Weber)* function such that the flrART calculates, in parallel, the degree of inclusion of an input datum $\mathbf{X}_i$ to each "set" category $\mathbf{W}_j \in S$. Furthermore, flrART's *match criterion* is the following inequality: $\sigma(\mathbf{W}_J \subseteq \mathbf{X}_i) \geq \rho$, implicit in step 6, where the winner category $\mathbf{W}_J$ calculates its degree of inclusion to the input datum $\mathbf{X}_i$. In conclusion, if the winner category $\mathbf{W}_J$ does not satisfy the *match criterion* then the winner category $\mathbf{W}_J$ is "reset" in step 7 by removing it (the $\mathbf{W}_J$) from the set S of the "set" categories. Otherwise, the winner category $\mathbf{W}_J$ is enhanced in step 14 by the lattice join operation $\mathbf{W}_J \doteq \mathbf{W}_J \dot{\cup} \mathbf{X}_i$ so as to include the input datum $\mathbf{X}_i$. Note that the set C in step 1 is, typically, empty; nevertheless, it could be $C = \{\mathbf{W}_1, \ldots, \mathbf{W}_K\}$, where $\mathbf{W}_k \in \mathfrak{I}_1^N$ for $k \in \{1, \ldots, K\}$. Furthermore, note that $|C|$ denotes the cardinality of set C. We point out that for an empty set $S = \{\}$ the corresponding input datum $\mathbf{X}_i \in \mathfrak{I}_1^N$ is memorized.

Some technical differences between flrART and *fuzzy ART* are summarized next. First, *fuzzy ART* employs, in particular, inclusion measure $\sigma_\cap(\mathbf{W}_j \subseteq \mathbf{X}_i)$ as *choice (Weber)* function. In fact, there is also a (small positive) parameter value α in the denominator of *fuzzy ART*'s choice (Weber) function, which has the following form $\frac{v(\mathbf{X}_i \cap \mathbf{W}_j)}{\alpha + v(\mathbf{W}_j)}$. Nevertheless, parameter α can be omitted as detailed in [20], [28]. Second, *fuzzy ART* assumes exclusively (as well as implicitly) the positive valuation $v(x) = x$ together with the dual isomorphic function $\theta(x) = 1 - x$ for normalized input patterns; the latter is assumed by *fuzzy ART*'s *complement coding* technique [6], [7]. Third, *fuzzy ART* employs inequality "$\sigma_\cap(\mathbf{X}_i \subseteq \mathbf{W}_J) \geq \rho$" as a *match criterion*. A critical advantage for inclusion measure $\sigma_{\dot\cup}(.,.)$ over $\sigma_\cap(.,.)$ is that only $\sigma_{\dot\cup}(.,.)$ is non-zero outside a category support; in other words, only $\sigma_{\dot\cup}(.,.)$ enables generalization beyond category support.

B. The flrFAM Scheme for Classification

Fig.3 displays the flrFAM neural architecture for classification inspired from the fuzzy-ARTMAP, or FAM for short [7]. That is, a synergy of two flrART modules for clustering, namely FLR$_a$ and FLR$_b$, interconnected via the MAP field F^{ab} whose operation is described next. During training, a pair $(\mathbf{X}, \ell(\mathbf{X})) \in \mathfrak{I}_1^N \times B$ is presented, where B is a set of category labels. Module FLR$_a$ clusters the input data $\mathbf{X}$, whereas module FLR$_b$ clusters the corresponding labels $\ell(\mathbf{X})$. Since we typically assume $\rho_b = 1$ it follows that module FLR$_b$ memorizes each label $\ell(\mathbf{X})$. Note that a category label is typically represented by a binary pattern of 0s and a single 1. The intermediate MAP field F^{ab} implements a function $\ell : \mathfrak{I}_1^N \to B$ that maps clusters (intervals) in FLR$_a$ to labels in FLR$_b$. A pair $(\mathbf{W}_k, \ell(\mathbf{W}_k))$, stored in the MAP field F^{ab}, is interpreted as rule $\mathcal{R}$: "if $\mathbf{W}_k$ then $\ell(\mathbf{W}_k)$", symbolically $\mathcal{R} : \mathbf{W}_k \to \ell(\mathbf{W}_k)$, induced from the training data.

The flrFAM training (learning) phase consists of two subphases, namely *structure identification* subphase and *parameter optimization* subphase. Algorithm 2 describes the *structure identification* subphase towards computing categories (clusters), i.e. hyperboxes in a lattice $(\mathfrak{I}_1^N, \subseteq)$. In particular, Algorithm 2 is a staightforward extension of FAM's learning algorithm [7] such that *fuzzy ART* modules ART$_a$ and ART$_b$ correspond to modules FLR$_a$ and FLR$_b$, respectively. Note that there is a single parameter, namely *baseline* vigilance $\overline{\rho_a} \in [0, 1]$, in the header "flrFAMstr($\overline{\rho_a}$)" of Algorithm 2. During training, parameter $\overline{\rho_a}$ may increase by a small positive

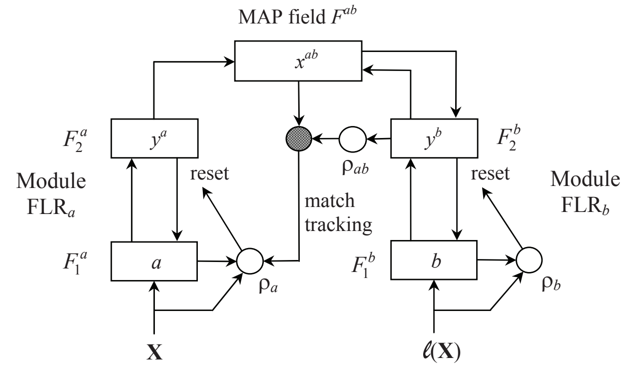


Fig. 3. The flrFAM neural architecture for classification, where $\mathbf{X} \in (\mathfrak{I}_1^N, \subseteq)$ and $\ell(\mathbf{X})$ is the category label of $\mathbf{X}$.

number ε (in steps 7 and 13) so as to resolve category contradiction. The set C_a in step 1 of Algorithm 2 is, typically, empty; nevertheless, it could be $C_a = \{\mathbf{W}_1, \ldots, \mathbf{W}_K\}$, where $\mathbf{W}_k \in \mathfrak{I}_1^N$ for $k \in \{1, \ldots, K\}$. The complexity of Algorithm 2 is determined by its two (nested) loops, likewise as the complexity of Algorithm 1 above. In conclusion, the complexity of flrFAM training for structure identification is quadratic $O(n_{trn}^2)$ in the number n_{trn} of the training data.

Algorithm 3 describes the *parameter optimization* subphase of flrFAM training (learning) in lattice $(\mathfrak{I}_1^N, \subseteq)$. Such a subphase does not exist in FAM [7]. The objective in this subphase is to optimize the parameters: baseline vigilance $\overline{\rho_a}$ and $A_1, \lambda_1, \mu_1, \ldots, A_N, \lambda_N, \mu_N$ in both the (sigmoid) positive valuation and the dual isomorphic function in every data dimension – Apparently, if we assume a different (parametric) positive valuation function then the corresponding parameters will have to be optimized. The "heart" of Algorithm 3 is a GENETIC optimization (step 30) of all the parameters in each of the N_p individual flrFAM classifiers per genetic algorithm generation. An individual flrFAM classifier in Algorithm 3 carries out structure identification in step 6 with a single parameter ($\overline{\rho_a}$). To avoid overtraining, the fitness Q_k of an individual flrFAM classifier is computed based on both training and validation data. The corresponding success rates S_{trn} and S_{val}, computed in steps 11 and 18, respectively, are jointly employed in step 21 towards computing the fitness Q_k, where $b_s \in [0, 1]$ is a user-defined *balancing factor for success* [30].

We point out that the categories (clusters) of an individual flrFAM classifier are induced, during the *structure identification* subphase, from the training data alone; moreover, the learned knowledge (categories) remains permanently in the system and may be updated, any time, by a system input (see in Algorithm 2, step 22). There is no pruning here. Note that, typically, an flrFAM classifier learns all its training data. All the parameter values of an individual flrFAM classifier are optimizable, during the *parameter optimization* subphase, using both the training data and the validation data. In conclusion, an "optimal" flrFAM classifier is computed in the sense that it learns well the training data, moreover it retains a capacity for generalization based on a balanced combination of the training data and the validation data.

The capacity of the aforementioned "optimal" flrFAM classifier for generalization is demonstrated by the success rate S_{tst} on the testing dataset in Algorithm 4.

Algorithm 2 flrFAMstr($\overline{\rho_a}$): flrFAM Training (Learning) – *Structure Identification* subphase

Assume, a set $C_a \subset 2^{\mathfrak{I}_1^N}$ in module FLR$_a$; $K = |C_a|$; a *baseline* vigilance parameter $\overline{\rho_a} \in [0,1]$; a small positive number ε; a set $B = \{b_1, \ldots, b_L\}$ of category labels; the vigilance parameter $\rho_b = 1$; a map $\ell : \mathfrak{I}_1^N \to B$ on C_a;
for $i = 1$ to $i = n_{trn}$ **do**
Consider the training datum $(\mathbf{X}_i, \ell(\mathbf{X}_i)) \in \mathfrak{I}_1^N \times B$;
$S \doteq C_a$;
$J \doteq \underset{\substack{j \in \{1,\ldots,|S|\} \\ \mathbf{W}_j \in S}}{\texttt{argmax}} \{\sigma(\mathbf{X}_i \subseteq \mathbf{W}_j)\}$;
if $\ell(\mathbf{W}_J) \neq \ell(\mathbf{X}_i)$ **then**
$\overline{\rho_a} = \sigma(\mathbf{W}_J \subseteq \mathbf{X}_i) + \varepsilon$;
end if
while $(S \neq \{\})$.and.$(\sigma(\mathbf{W}_J \subseteq \mathbf{X}_i) < \overline{\rho_a})$ **do**
$S \doteq S \setminus \{\mathbf{W}_J\}$;
$J \doteq \underset{\substack{j \in \{1,\ldots,|S|\} \\ \mathbf{W}_j \in S}}{\texttt{argmax}} \{\sigma(\mathbf{X}_i \subseteq \mathbf{W}_j)\}$;
if $\ell(\mathbf{W}_J) \neq \ell(\mathbf{X}_i)$ **then**
$\overline{\rho_a} = \sigma(\mathbf{W}_J \subseteq \mathbf{X}_i) + \varepsilon$;
end if
end while
if $S = \{\}$ **then**
$C_a \doteq C_a \cup \{\mathbf{X}_i\}$; $K \doteq K + 1$;
if $\ell(\mathbf{X}_i) \notin B$ **then**
$B \doteq B \cup \{\ell(\mathbf{X}_i)\}$; $L \doteq L + 1$;
end if
else
$\mathbf{W}_J \doteq \mathbf{W}_J \dot{\cup} \mathbf{X}_i$;
end if
end for

For $\sigma = \sigma_\cap$, $v(x) = x$ and $\theta(x) = 1 - x$ in the unit hypercube, Algorithms 1, 2 and 4 describe the classic FAM.

The applicability of the flrFAM classifier can be extended to a general product lattice $\mathfrak{L}_1 \times \cdots \times \mathfrak{L}_N$ including the lattice $(\mathfrak{F}_2^N, \preceq)$ of Type-2 INs as a special case.

IV. Human Facial Expression Recognition

Human-Machine Interaction (HMI) is an emerging application domain of general interest that includes anthropocentric computing, cognitive robotics, etc. The last decade has witnessed a growing interest in anthropocentric computing, that is computing such that a human is directly involved in the computation, e.g. emotion and/or facial expression recognition, human activity recognition, etc. [10], [40]. Even though an assortment of computational modeling techniques have been proposed, it is recognized that the area lacks general mathematical modeling techniques [1].

A. The Lattice Computing (LC) Paradigm

It has been argued lately that a major reason for the existence of different information processing paradigms is the

Algorithm 3 flrFAMpar: flrFAM Training (Learning) – *Parameter Optimization* subphase

A user defines the integers $N_G > 0$ and $N_p > 0$ as well as $b_s \in [0,1]$. Let $cntr = 0$, $Q_{prev} = 0$;
Randomize parameters (i) *baseline* vigilance $\overline{\rho_a} \in [0,1]$ and (ii) $A_i \in [0,100]$, $\lambda_i \in [0,10]$ and $\mu_i \in [-10,10]$ for both one sigmoid positive valuation $v_s(x; A_i, \lambda_i, \mu_i) = A_i / \left(1 + e^{-\lambda_i (x - \mu_i)}\right)$ and one dual isomorphic function $\theta_i(x) = 2\mu_i - x$ per data dimension $i \in \{1, \ldots, N\}$;
while $cntr \leq N_G$ **do**
for $k = 1$ to $k = N_p$ **do**
Let $S_{trn} = S_{val} = 0$;
flrFAMstr($\overline{\rho_a}$);
for $i = 1$ to $i = n_{trn}$ **do**
Consider training datum $(\mathbf{X}_i, \ell(\mathbf{X}_i)) \in \mathfrak{I}_1^N \times B$;
$J \doteq \underset{\substack{j \in \{1,\ldots,|C_a|\} \\ \mathbf{W}_j \in C_a}}{\texttt{argmax}} \{\sigma(\mathbf{X}_i \subseteq \mathbf{W}_j)\}$;
if $\ell(\mathbf{W}_J) = \ell(\mathbf{X}_i)$ **then**
Update the training data success rate S_{trn};
end if
end for
for $i = 1$ to $i = n_{val}$ **do**
Consider validation datum $(\mathbf{X}_i, \ell(\mathbf{X}_i)) \in \mathfrak{I}_1^N \times B$;
$J \doteq \underset{\substack{j \in \{1,\ldots,|C_a|\} \\ \mathbf{W}_j \in C_a}}{\texttt{argmax}} \{\sigma(\mathbf{X}_i \subseteq \mathbf{W}_j)\}$;
if $\ell(\mathbf{W}_J) = \ell(\mathbf{X}_i)$ **then**
Update the validation data success rate S_{val};
end if
end for
$Q_k \doteq b_s S_{trn} + (1 - b_s) S_{val}$;
end for
$J \doteq \underset{k \in \{1,\ldots,N_p\}}{\texttt{argmax}} \{Q_k\}$;
if $Q_J = Q_{prev}$ **then**
$cntr \doteq cntr + 1$;
else
$cntr \doteq 0$;
end if
$Q_{prev} \doteq Q_J$;
GENETIC optimization of the N_p individual flrFAM classifiers' parameters $\overline{\rho_a}, A_1, \lambda_1, \mu_1, \ldots, A_N, \lambda_N, \mu_N$;
end while

Algorithm 4 flrFAMtst: flrFAM Testing (Generalization) phase

Assume, a set $C_a = \{\mathbf{W}_1, \ldots, \mathbf{W}_K\} \subset 2^{\mathfrak{I}_1^N}$ in module FLR$_a$; a set $B = \{b_1, \ldots, b_L\}$ of category labels in module FLR$_b$; a map $\ell : \mathfrak{I}_1^N \to B$ on C_a;
for $i = 1$ to $i = n_{tst}$ **do**
Consider the next testing datum $(\mathbf{X}_i, b_i) \in \mathfrak{I}_1^N \times B$;
$J \doteq \underset{\substack{j \in \{1,\ldots,|C_a|\} \\ \mathbf{W}_j \in C_a}}{\texttt{argmax}} \{\sigma(\mathbf{X}_i \subseteq \mathbf{W}_j)\}$;
The testing datum $\mathbf{X}_i$ is classified in category $\ell(\mathbf{W}_J)$;
end for
Compute the overall testing data success rate S_{tst};

need to cope with disparate types of data including matrices of numbers, (distribution) functions, sets, set partitions, logic values, relations, (strings of) symbols, etc. In conclusion, motivated by the fact that popular types of data (including the aforementioned ones) are lattice-ordered, a unified modeling and knowledge-representation has been proposed based on mathematical lattice theory [22], [23].

The term "Lattice Computing (LC)" has been proposed as a Computational Intelligence branch that develops algorithms in $(\mathfrak{R}, \vee, \wedge, +)$, where $\mathfrak{R}$ is the set of real numbers [14], [15], [16]. This work proposes the term "Lattice Computing (LC) paradigm" for denoting an evolving collection of tools and mathematical modeling methodologies with a capacity to process disparate types of (lattice ordered) data per se including logic values, numbers, sets, symbols, graphs, etc.

In the aforementioned sense HMI, including anthropocentric computing, emerges as a promising application domain for the LC paradigm. More specifically, IN-based LC techniques may combine (numeric) machine learning techniques with (semantic) rule-based interpretations as shown below.

B. *The Pattern Recognition Problem*

Humans may interact with computers by hand gestures, facial expressions, speech or combinations of them. Among those interactions, facial expressions are especially interesting also because they can fairly easily represent human emotions. Hence, facial expressions have already been used in interactive computer games as indicators of the player's intention and/or satisfaction [49], in patient monitoring for pain detection [18], in sign language communication systems [38], etc.

A critical information-processing module in any electronic system for recognizing facial expressions is a classifier. Facial expression recognition can be cast as a pattern recognition problem, where a facial expression has to be recognized among a number of known facial expressions including, for example, happiness, sadness, surprise, fear, pain etc. Towards the aforementioned (recognition) objective "feature extraction" is typically pursued in a data preprocessing step.

Several feature extraction alternatives on digital images have been proposed in the literature including wavelet features [45], facial attributes [19], Gabor features [17] and Zernike moments [32]. Action units (AUs), i.e. the smallest visually discernible facial movements, are especially popular features [47]. In this work we employ *orthogonal moments*, that is an invertible image transform [44] known for its effectiveness in potentially rotation-scale-translation (RST) invariant pattern recognition applications [43]. Even though specific moments (Zernike) have already been employed for facial expression recognition [32], to the authors' best knowledge, this is the first joint/comparative employment of different moments features for human facial expression recognition.

V. Experiments and Results

We carried out a number of human facial and emotional expression recognition experiments by the flrFAM classifier as described in this section.

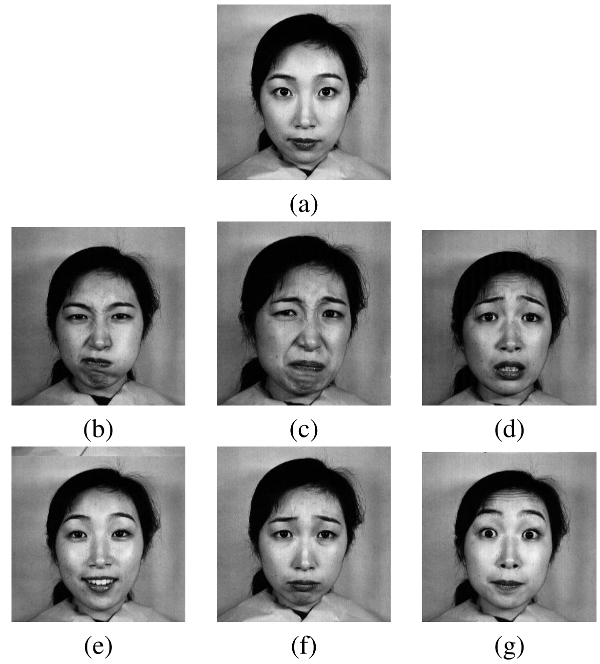


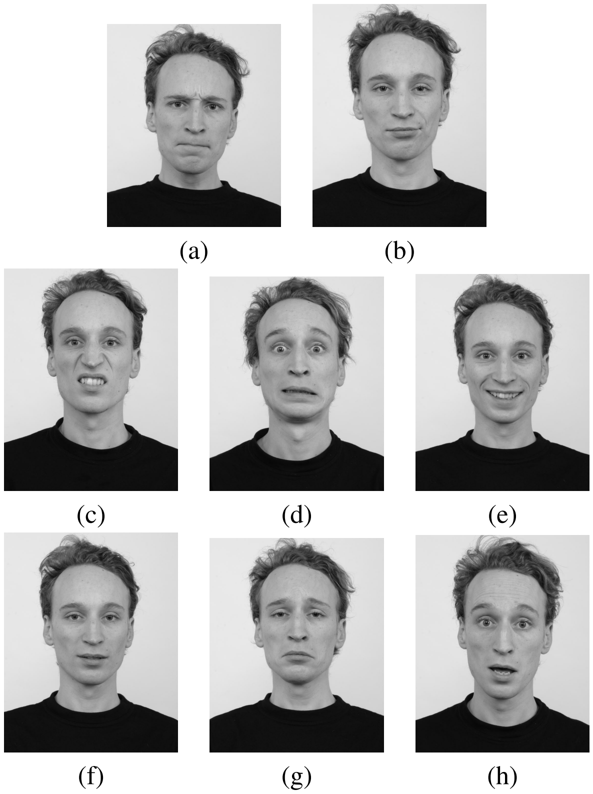
Fig. 4. Seven different facial expressions, from the JAFFE benchmark data set, including (a) "neutral", (b) "angry", (c) "disgusted", (d) "fear", (e) "happy", (f) "sad", and (g) "surprise".

A. *Benchmark Datasets*

Two facial expression recognition benchmark datasets were engaged. First, the JAFFE dataset [34] including 213 frontal images (with 256×256 pixels per image) of 10 different persons corresponding to seven common human facial expressions, namely "neutral" (30), "angry" (30), "disgusted" (29), "fear" (32), "happy" (31), "sad" (31) and "surprise" (30) regarding Japanese female subjects (Fig.4). Second, the RADBOUD dataset [33] including $67 \times 8 = 536$ frontal images (with 681×1024 pixels per image) corresponding to 8 common emotional expressions, namely "angry" (67), "contemptuous" (67), "disgusted" (67), "fear" (67), "happy" (67), "neutral" (67), "sad" (67) and "surprise" (67) regarding Caucasian and Moroccan subjects both male and female (Fig.5). A number within parentheses above, indicates the number of images per facial/emotional expression.

B. *Data Preprocessing and Feature Extraction*

In an initial "data preprocessing" step we removed irrelevant image content such as background/hair by, first, applying the Viola-Jones face detector [48] so as to separate the head region from the background and, second, by masking the face with an ellipse so as to remove the hair and include as much facial information as possible. In a final "data preprocessing" step we used the latter (face) segment for feature extraction by the method of orthogonal moments. Six kinds of moments, namely Zernike, Pseudo-Zernike, Fourier-Mellin, Legendre, Tchebichef and Krawtchouk moments [44] were computed up to order 6 and 5 (for order 5 we kept only the first 16 moments) for Zernike and Pseudo-Zernike moments, respectively, and up to order 3 for all other moments. In each case, a 16-dimensional feature vector (including 16 moments of a kind)



(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 5. Eight different emotional expressions, from the RADBOUD benchmark data set, including (a) “Angry”, (b) “Contemptuous”, (c) “Disgusted”, (d) “Fear”, (e) “Happy”, (f) “Neutral”, (g) “Sad”, and (h) “Surprise”.

was computed per image. The induction of a Type-1 IN from a vector of real numbers was carried out as detailed in [30].

C. Computational Experiments

We carried out a number of experiments with different classifiers on either 16- or 96- dimensional (feature) vectors that represented an image. More specifically, a 16-dimensional (feature) vector included 16 moments of a kind regarding either Zernike or Pseudo-Zernike or Fourier-Mellin or Legendre or Tchebichef or Krawtchouk moments, *separately*; whereas, a $6 \times 16 = 96$-dimensional (feature) vector was produced by *concatenating* six 16-dimensional (feature) vectors for the six aforementioned kinds of moments, respectively.

We employed a number of classifiers including the k-Nearest-Neighbor (kNN) [17] with $k = 1$, Linear Discriminant Analysis (LDA) [9], Naive Bayes [32], Classification Trees [11], feedforward Neural Networks [35] and FAM [7], all implemented in the MATLAB 7.8.0 integrated development environment (IDE). Moreover, we employed the flrFAM classifier implemented in the C++ programming language.

In our classification experiments, a different facial/emotional expression corresponded to a different class. We randomly partitioned the data in three mutually disjoint sets: one for training, one for validation and another one for testing. More specifically, for the JAFFE benchmark the datasets for training, validation and testing included 184, 7 and 22 images, respectively; whereas, for the RADBOUD benchmark they included 472, 10 and 54 images, respectively. We repeated the aforementioned (random) data partition 10 times. Care was taken so that all different classes be represented fairly in the datasets for training, validation and testing. Every experiment was repeated 10 times using the same (random) data partitions for all classifiers. We point out that three dataset partitions (i.e., for training, validation and testing) were employed only by the Neural Network and the flrFAM classifiers; whereas, the remaining classifiers employed jointly the training dataset and the validation dataset for training.

1) Experiments with 96-dimensional feature vectors: All the classifiers were applied in the Euclidean space $\mathfrak{R}^{96}$ but the LDA classifier which could not be applied for numerical reasons due to the large input data dimension (96) comparatively to the total number of the training data. For the Neural Network classifier an optimal number of hidden layer neurons was estimated by “trial-and-error” to 50. The flrFAM classifier was applied by representing an image by a 6-dimensional trivial Type-2 IN $\overrightarrow{\mathbb{E}} = [\overrightarrow{E}, \overrightarrow{E}]$, where a Type-1 IN in $\overrightarrow{E} \in \mathfrak{F}_1^6$ was induced from a 16-tuple of numeric (feature) data that corresponded to a moment kind.

2) Experiments with 16-dimensional feature vectors: All the classifiers were applied in space $\mathfrak{R}^{16}$. In particular, a Neural Network classifier was applied with an optimal number of hidden layer neurons estimated by “trial-and-error” to 16. The flrFAM classifier was applied by representing an image by a 16-dimensional trivial Type-2 IN $\overrightarrow{\mathbb{E}} = [\overrightarrow{E}, \overrightarrow{E}]$, where a trivial Type-1 IN $\overrightarrow{E} \in \mathfrak{F}_1^{16}$ was induced from the corresponding feature vector data. Hence, the flrFAM computed “hyperboxes” for an *upper* Type-2 IN envelope, whereas the corresponding *lower* Type-2 IN envelope was the empty set.

In an N-dimensional flrFAM classification experiment (for either $N = 6$ or $N = 16$), an inclusion measure ($\sigma = \sigma_{\dot{\cup}}$) was computed in the product lattice $(\mathfrak{F}_2^N, \preceq)$ using equations (3) and (4). All descriptor values were normalized. A Type-1 IN was represented with $L = 32$ intervals spaced evenly from $h = 0$ to $h = 1$ included.

Regarding parameter optimization by a genetic algorithm, the phenotype of an *individual* (flrFAM classifier) consisted of specific values for 3 sigmoid function $v_s(x; A_i, \lambda_i, \mu_i)$ parameters A_i, λ_i and μ_i per data dimension $i \in \{1, \ldots, N\}$. An additional parameter was the baseline vigilance $\overline{\rho_a}$. Hence, a total number of $3N + 1$ parameters was binary-encoded in the chromosome of an individual. We included $N_p = 25$ individuals per generation. The genetic algorithm was enhanced by the *microgenetic hill-climbing* operator and, in addition, both *elitism* and *adaptive crossover/mutation* rates were implemented [41]. A *balancing factor for success* $b_s = 0.5$ (see Algorithm 3, step 21) was employed. The genetic algorithm was left to evolve until no improvement was observed in the fitness (Q_J) of the best individual for $N_G = 30$ generations in a row. Then, the testing data were applied once and the *testing data percentage success rate* (or, equivalently, *generalization rate*) S_{tst} was recorded.

Table I displays the “minimum (min)”, “maximum (Max)”, “average (ave)” and “standard deviation (std)” statistics of the generalization rate (%) regarding the JAFFE benchmark dataset in 10 computational experiments for a number of classifiers and the aforementioned six kinds of moments *con-*

TABLE I
GENERALIZATION RATE (%) STATISTICS REGARDING THE JAFFE TESTING DATA IN 10 COMPUTATIONAL EXPERIMENTS USING SEVERAL CLASSIFIERS AND SIX DIFFERENT KINDS OF MOMENTS, CONCATENATED

Classifier name	min	Max	ave	std
kNN (k=1)	40.91	94.74	67.68	15.82
Naive Bayes	18.18	52.63	36.80	10.03
Classification Tree	31.82	47.37	40.02	5.67
Neural Network (50)	18.18	59.09	37.27	13.52
FAM	50.00	90.00	68.87	13.49
flrFAM	50.00	86.36	69.54	12.31

TABLE II
GENERALIZATION RATE (%) STATISTICS REGARDING THE RADBOUD TESTING DATA IN 10 COMPUTATIONAL EXPERIMENTS USING SEVERAL CLASSIFIERS AND SIX DIFFERENT KINDS OF MOMENTS, CONCATENATED

Classifier name	min	Max	ave	std
kNN (k=1)	22.22	46.30	35.74	7.51
Naive Bayes	35.19	57.41	48.15	7.04
Classification Tree	27.78	40.74	34.07	4.20
Neural Network (50)	11.11	64.81	45.74	15.81
FAM	27.77	44.44	37.40	6.03
flrFAM	35.18	50.00	43.14	4.86

catenated, whereas Table II displays the corresponding statistics regarding the RADBOUD benchmark dataset. Likewise, Table III displays "min", "Max", "ave" and "std" statistics of the generalization rate (%) regarding the JAFFE benchmark dataset in 10 experiments using various classifiers and the aforementioned six kinds of moments *separately*, whereas Table IV displays the corresponding statistics regarding the RADBOUD benchmark dataset.

The computation of any kind of 16 moments took around 0.5 minute per image. A full classification experiment for one image data partition took around: 1 minute for each one of the kNN, LDA, Naive Bayes and Classification Tree classifiers; 2 minutes for the FAM classifier; 4 minutes for the Neural Network classifier; 61 minutes for the flrFAM classifier due mainly to the computationally expensive genetic algorithm optimization (see in Algorithm 3, step 30). Note that without any optimization, flrFAM was as fast as FAM.

Our computational experiments with the flrFAM on the 96-dimensional feature vectors of the JAFFE dataset induced the set of rules shown in Fig.6. In particular, Fig.6 displays one 6-tuple Type-2 IN per class as follows. The first six columns of the 7×7 Table in Fig.6 (excluding the header) display Type-2 INs corresponding to Zernike (MOMS_Z), Pseudo-Zernike (MOMS_PZ), Fourier-Mellin (MOMS_FM), Legendre (MOMS_L), Tchebichef (MOMS_T) and Krawtchouk (MOMS_K) moments, respectively; the seventh column displays the corresponding class name. For instance, the first row of the 7×7 Table in Fig.6 displays a data-induced "Type-2 6-tuple IN" granule for the class (facial expression) ANGRY, the second row displays the corresponding granule for class DISGUSTED, etc. We point out that the lower/upper envelope $U/W \in \mathfrak{F}_1$ of a Type-2 IN $\mathbb{E} = [U, W]$ in Fig.6 is indicated in bold (black) color, whereas all the *encoded* Type-1 INs are indicated in light (red) color within a Type-2 IN. A similar set of rules was induced by the flrFAM from the 96-dimensional feature vectors of the RADBOUD benchmark dataset.

TABLE III
GENERALIZATION RATE (%) STATISTICS REGARDING THE JAFFE TESTING DATA IN 10 COMPUTATIONAL EXPERIMENTS USING SEVERAL CLASSIFIERS AND SIX DIFFERENT KINDS OF MOMENTS, SEPARATELY

CLASSIFIER NAME Moment Type	min	Max	ave	std
kNN (k=1)				
1) Zernike	50.00	95.45	80.37	13.04
2) Pseudo-Zernike	45.45	90.91	78.57	13.69
3) Fourier-Mellin	45.45	90.91	73.98	14.52
4) Legendre	63.64	100.00	75.91	10.06
5) Tchebichef	63.64	100.00	75.00	11.19
6) Krawtchouk	40.91	95.45	66.24	16.36
LDA				
1) Zernike	40.91	68.18	52.49	9.73
2) Pseudo-Zernike	40.91	59.09	51.24	7.62
3) Fourier-Mellin	31.82	72.73	53.82	11.96
4) Legendre	36.36	77.27	53.18	10.51
5) Tchebichef	36.36	77.27	53.18	10.51
6) Krawtchouk	27.27	54.54	41.46	9.85
NAIVE BAYES				
1) Zernike	22.73	50.00	32.39	8.83
2) Pseudo-Zernike	22.73	45.45	30.89	6.77
3) Fourier-Mellin	27.27	50.00	41.36	7.25
4) Legendre	9.09	40.91	27.18	8.30
5) Tchebichef	9.09	36.36	25.81	7.11
6) Krawtchouk	18.18	54.54	32.61	13.30
CLASSIFICATION TREE				
1) Zernike	27.27	54.55	40.57	8.29
2) Pseudo-Zernike	18.18	50.00	32.18	10.86
3) Fourier-Mellin	13.64	45.45	33.22	10.10
4) Legendre	22.73	45.45	32.90	7.63
5) Tchebichef	13.64	50.00	28.38	11.96
6) Krawtchouk	22.73	45.45	32.61	7.35
NEURAL NETWORK (16)				
1) Zernike	9.09	50.00	29.18	14.54
2) Pseudo-Zernike	4.55	63.64	33.48	19.45
3) Fourier-Mellin	13.63	68.18	37.13	19.06
4) Legendre	9.09	100.00	32.88	25.24
5) Tchebichef	9.09	59.09	39.40	16.58
6) Krawtchouk	4.55	50.00	25.00	15.34
FAM				
1) Zernike	50.00	95.45	79.00	12.14
2) Pseudo-Zernike	50.00	90.90	74.90	12.22
3) Fourier-Mellin	36.36	86.36	63.54	15.19
4) Legendre	45.45	95.45	72.27	12.57
5) Tchebichef	54.54	95.45	72.72	10.71
6) Krawtchouk	40.90	85.00	63.45	16.67
flrFAM				
1) Zernike	59.09	95.45	83.63	10.53
2) Pseudo-Zernike	54.54	90.90	79.08	12.52
3) Fourier-Mellin	50.00	86.36	75.90	13.04
4) Legendre	59.09	95.45	77.72	11.82
5) Tchebichef	63.63	95.45	77.26	10.92
6) Krawtchouk	45.45	95.45	69.54	15.00

D. *Significance of the Results*

Based on 10 experiments for 10 (random) data partitions, respectively, we evaluated all classifiers *pairwise* using the one-sided "matched pairs" statistical t test with $df = 9$ degrees of freedom. The null hypothesis H_0: "the two classifiers (in a pair) give similar results" was tested versus the alternative hypothesis H_a: "the second classifier (in a pair) improves classification performance". For each evaluation we computed the P-value of the statistic $t = (\overline{x} - 0)/(s/\sqrt{n})$ for $n = 10$, where $\overline{x}$ is the sample average of differences in generalization accuracy and s is the corresponding standard deviation. We worked at 5% level of significance.

Table V presents our results for the JAFFE 96-dimensional

TABLE IV
GENERALIZATION RATE (%) STATISTICS REGARDING THE RADBOUD TESTING DATA IN 10 COMPUTATIONAL EXPERIMENTS USING SEVERAL CLASSIFIERS AND SIX DIFFERENT KINDS OF MOMENTS, SEPARATELY

CLASSIFIER NAME Moment Type	min	Max	ave	std
kNN (k=1)				
1) Zernike	33.33	51.85	41.30	6.05
2) Pseudo-Zernike	33.33	50.00	41.30	5.79
3) Fourier-Mellin	35.19	55.56	44.81	5.71
4) Legendre	33.33	48.15	40.37	5.44
5) Tchebichef	31.48	51.85	40.93	6.44
6) Krawtchouk	22.22	46.30	35.56	7.45
LDA				
1) Zernike	37.04	57.41	48.15	7.20
2) Pseudo-Zernike	35.19	59.26	47.59	8.86
3) Fourier-Mellin	46.30	66.67	55.37	6.84
4) Legendre	42.59	59.26	49.26	5.53
5) Tchebichef	42.59	59.26	49.26	5.53
6) Krawtchouk	27.78	50.00	41.30	7.51
NAIVE BAYES				
1) Zernike	37.04	55.56	45.37	7.57
2) Pseudo-Zernike	33.33	61.11	43.89	8.24
3) Fourier-Mellin	37.04	59.26	45.93	7.03
4) Legendre	33.33	53.70	41.67	6.13
5) Tchebichef	33.33	53.70	41.85	6.12
6) Krawtchouk	16.67	40.74	27.22	7.56
CLASSIFICATION TREE				
1) Zernike	22.22	37.04	29.07	5.92
2) Pseudo-Zernike	24.07	40.74	32.04	5.52
3) Fourier-Mellin	24.07	44.44	33.52	6.56
4) Legendre	22.22	42.59	32.59	6.49
5) Tchebichef	12.96	50.00	28.70	10.23
6) Krawtchouk	20.37	44.44	27.96	6.67
NEURAL NETWORK (16)				
1) Zernike	16.67	51.85	29.26	11.41
2) Pseudo-Zernike	5.56	38.89	29.07	9.92
3) Fourier-Mellin	18.52	61.11	46.67	12.52
4) Legendre	24.07	55.56	41.11	10.10
5) Tchebichef	16.67	53.70	35.19	14.48
6) Krawtchouk	16.67	46.30	31.11	11.31
FAM				
1) Zernike	24.07	42.59	34.44	7.20
2) Pseudo-Zernike	31.48	48.14	37.40	5.71
3) Fourier-Mellin	33.33	48.14	40.18	4.36
4) Legendre	31.48	50.00	42.77	6.73
5) Tchebichef	33.33	55.55	43.88	7.20
6) Krawtchouk	22.22	42.59	32.59	6.30
flrFAM				
1) Zernike	35.18	50.00	42.03	5.45
2) Pseudo-Zernike	35.18	48.14	41.84	5.17
3) Fourier-Mellin	37.03	53.70	43.14	5.24
4) Legendre	37.03	48.14	42.21	4.07
5) Tchebichef	33.33	51.85	41.66	5.80
6) Krawtchouk	24.07	48.14	37.40	7.18

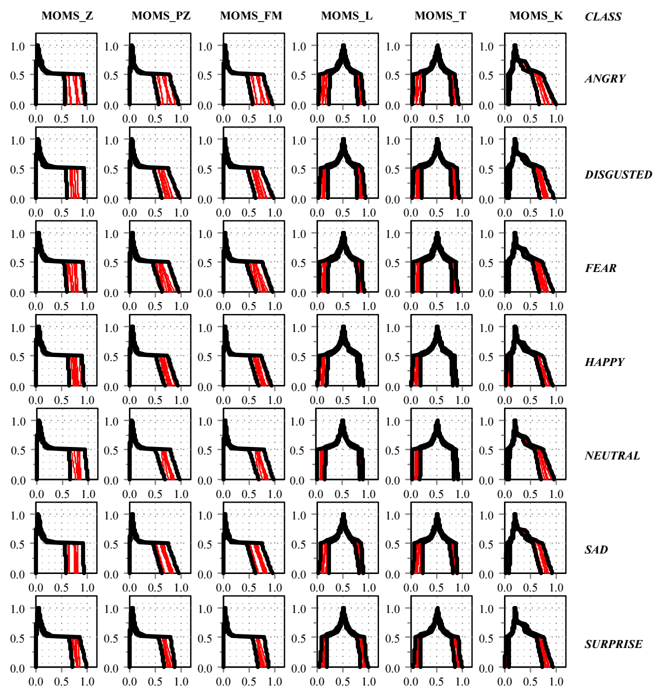


Fig. 6. A row of the 7×7 Table above (excluding the header) displays one 6-dimensional Type-2 IN induced for each of the seven human facial expressions (classes) of the JAFFE benchmark dataset. One Type-2 IN corresponds to one kind of moments. At the end of a row, the corresponding class name is shown.

TABLE V
P-VALUES OF THE ONE-SIDED "MATCHED PAIRS" STATISTICAL t TEST WITH $df = 9$ DEGREES OF FREEDOM FOR PAIRWISE CLASSIFIER EVALUATION ON THE JAFFE 96-DIMENSIONAL FEATURE VECTORS

Classifier	kNN	NBayes	CTree	NN (50)	FAM	flrFAM
kNN		0	0.0001	0.0003	0.1991	0.3217
NBayes			0.1178	0.4663	0	0
CTree				0.2962	0	0
NN (50)					0.0001	0
FAM						0.4290

data. In particular, a comparison of the testing data accuracy of the flrFAM (69.54%) with the kNN (67.68%) and FAM (68.87%) classifiers resulted in $t = 0.4796$ and $t = 0.1842$, which implied $P = 0.3217$ and $P = 0.4290$, respectively. Hence, the null hypothesis H_0 could not be rejected; in other words, the flrFAM appears to perform as well as either classifier kNN or FAM. Moreover, a comparison of flrFAM with the Naive Bayes (36.80%), Classification Tree (40.02%) and Neural Network (37.27%) classifiers resulted in $t = 8.1986$, $t = 8.2653$ and $t = 6.6391$, which practically implied $P = 0$. Hence, the null hypothesis H_0 could not be accepted; in other words, the flrFAM appears to improve the generalization rate. Furthermore, for the RADBOUD 96-dimensional data, the Naive Bayes, Neural Network and flrFAM classifiers produced the best (statistically significant) generalization rates.

We repeated the previous experiments for both the JAFFE and the RADBOUD 96-dimensional data such that a population of 16 data, corresponding to a moment kind, was replaced by its *first order statistic*, namely its *average*. We recorded an average performance drop by up to 40% and 20% for the JAFFE and the RADBOUD, respectively. We attributed the aforementioned drop to the loss of "discriminatory" information. In particular regarding flrFAM, note that an IN advantage is its representation of all order data statistics [24], [25], [26].

We carried out additional statistical hypothesis testing to evaluate, pairwise, different kinds of moments for each classifier. For the JAFFE 16-dimensional data, the kNN, FAM and flrFAM classifiers with (Pseudo-)Zernike moments produced the highest generalization rates. Moreover, for the RADBOUD 16-dimensional data, the LDA classifier with Fourier-Mellin moments produced the highest generalization rates followed by the kNN, Naive Bayes, Neural Network and flrFAM classifiers also with Fourier-Mellin moments as well as by the FAM classifier also with Tchebichef moments. It was confirmed that

TABLE VI
AUC VALUES FOR THREE CLASSIFIERS AND 96-DIM FEATURE VECTORS FOR JAFFE CLASSES "NEUTRAL" (C1), "ANGRY" (C2), "DISGUSTED" (C3), "FEAR" (C4), "HAPPY" (C5), "SAD" (C6) AND "SURPRISE" (C7)

Classifier	c1	c2	c3	c4	c5	c6	c7
kNN (k=1)	0.87	0.82	0.78	0.87	0.89	0.81	0.78
FAM	0.90	0.82	0.80	0.89	0.90	0.82	0.79
flrFAM	0.80	0.75	0.78	0.92	0.94	0.80	0.77

no specific kind of moments is is globally preferable.

The flrFAM classifier application in the JAFFE problem on 16-dimensional vectors produced better generalization rates than its application on 96-dimensional vectors; that is, keeping different moment features in different dimensions improves flrFAM's generalizability compared to mingling different moment features in a single dimension. The latter improvement was not confirmed in the RADBOUD problem, where no statistically significant difference was mostly recorded between the 16- and 96-dimensional vector representations.

We studied the confusion of different classifiers. First, we present our average (confusion) results in 10 experiments on 10 random data partitions regarding the 96-dimensional feature vectors of the JAFFE problem. It turned out that the kNN classifier learns well the classes "neutral" (61.81%), "fear" (60.42%) and "happy" (61.36%), whereas it learns the remaining classes in the range 51%-58%; the largest error is the 24.31% confusion of class "sad" with class "surprise". The FAM classifier learns well the classes "neutral" (67.36%), "disgusted" (64.03%), "happy" (67.42%) and "surprise" (66.57%), whereas it learns the remaining classes in the range 54%-58%; the largest error is the 25% confusion of class "angry" with class "disgusted". The flrFAM classifier learns well the classes "neutral" (63.19%), "fear" (71.53%), "happy" (76.52%) and "surprise" (65.19%), whereas it learns the remaining classes in the range 47%-58%; the largest error is the 22.22% confusion of class "sad" with class "surprise". The remaining classifiers typically confused a class to over 50%. Second, we confirmed that classification results deteriorated considerably for the RADBOUD benchmark. More specifically, even though all classifiers recognized class "neutral" well in the range 62%-87%, they typically confused any other class to over 50%. Note that likewise confusion results were recorded for all 16-dimensional feature vector data in both the JAFFE and the RADBOUD classification problems.

To further demonstrate a classifier system performance, we computed Receiver Operating Characteristics (ROC) curves. Each ROC curve computation was based on a few tens of "false-positive, true-positive" pairs of points. For lack of space, we display only the corresponding Area Under Curve (AUC) values [13] in Table VI for the three "best performing" classifiers regarding the 96-dimensional JAFFE data. In particular, a Table VI cell entry is the average of 10 AUC values for 10 random data partitions. Note that the nearest a Table VI entry is to 1, the better the corresponding classifier (generalization) performance. Table VI shows that the best performance was attained by either classifier FAM or flrFAM.

Next, we give a measure of comparison of our techniques with alternative ones. Note that a number of facial expression recognition schemes have been reported in the literature mostly for the JAFFE [17], [32], [45] rather than for the RADBOUD benchmark [19]. More specifically, first, the works in [17], [32] and [45] have reported a maximum classification rate of 89.67%, 92.8% and 95.71%, respectively, using different machine-learning classification schemes, different specific features as well as different training/testing datasets. Second, the work in [19] has reported a maximum classification rate of 93.96% using a Fuzzy Inference System (FIS) with human-defined initial rules, different features and 414 frontal images regarding six basic emotions and two gaze directions. Apparently, the maximum (Max) classification rates reported in Tables I and III for the JAFFE benchmark compare well with the aforementioned results from the literature. Moreover, it appears that all aforementioned 414 frontal images of the RADBOUD benchmark were employed in [19] for testing. Given both the sizes of our data sets for training and testing (i.e. around 90% and 10%, respectively) and the fact that the flrFAM typically learns all its training data, it follows that the flrFAM here can outperform the classifier in [19].

The flrFAM classifier performed as good as the FAM or the kNN classifier (for $k = 1$) because they operate on the same principle: The kNN decides based on the distance of a testing datum from the nearest (labeled) training datum, whereas the (flr)FAM classifier decides based on the inclusion of a testing datum into a (labeled) category induced from the training data. A unique advantage of the flrFAM classifier is the induction of *flexible* (i.e., *tunable*) descriptive decision-making knowledge (rules) as shown in Fig.6, which (Fig.6) also indicates that the flrFAM can be interpreted as a fuzzy neural classifier. Moreover, since Type-2 INs are involved, this work paves the way for sound extensions of FAM to Type-2 FISs [36].

VI. CONCLUSION

This work has introduced the novel flrFAM neural classifier as a Lattice Computing (LC) extension of the *fuzzy ARTMAP* (*FAM*) neural classifier for real-time learning and classification of nonstationary data followed by an application to facial expression recognition. Comparative computational experiments have demonstrated the viability of our proposed techniques.

The work here emphasized an application of the flrFAM classifier to (static) human facial expression recognition. Advantages include the induction of *flexible* (i.e., *tunable*) rules computable by machine learning techniques as well as the capacity for granular computing so as to cope with data uncertainty/ambiguity. An additional advantage is flrFAM's capacity for (non)numeric data fusion based, rigorously, on data semantics represented by partial-order.

Future work plans include extensions to dynamic (video) human recognition applications engaging, as well, additional types of data such as voice, etc.

ACKNOWLEDGMENT

This work has been supported, in part, by the European Union (Social Fund) and Greek national resources under the framework of the "Archimedes III: Funding of Research Groups in TEI of Athens" project of the "Education & Lifelong Learning" Operational Programme.

REFERENCES

[1] D. Aarno and D. Kragic, "Motion intention recognition in robot assisted applications," *Robotics and Autonomous Systems*, vol. 56, no. 8, pp. 692–705, 2008.

[2] J. Andreu and P. Angelov, "Real-time human activity recognition from wireless sensors using evolving fuzzy systems," in *2010 IEEE Intl. Conf. on Fuzzy Systems (FUZZ)*, Barcelona, Spain, Jul. 18-23, 2010, pp. 1–8.

[3] G. Birkhoff, *Lattice Theory*. Providence, RI: American Mathematical Society, 1967, Colloquium Publications, vol. 25.

[4] G.A. Carpenter and S. Grossberg, "A massively parallel architecture for a self-organizing neural pattern recognition machine," *Computer Vision, Graphics, and Image Processing*, vol. 37, no. 1, pp. 54–115, 1987.

[5] G.A. Carpenter, S. Grossberg, and J.H. Reynolds, "A fuzzy ARTMAP nonparametric probability estimator for nonstationary pattern recognition problems," *IEEE Transactions on Neural Networks*, vol. 6, no. 6, pp. 1330–1336, 1995.

[6] G.A. Carpenter, S. Grossberg, and D.B. Rosen, "Fuzzy ART: fast stable learning and categorization of analog patterns by an adaptive resonance system," *Neural Networks*, vol. 4, no. 6, pp. 759–771, 1991.

[7] G.A. Carpenter, S. Grossberg, N. Markuzon, J.H. Reynolds, and D.B. Rosen, "Fuzzy ARTMAP: a neural network architecture for incremantal supervised learning of analog multidimensional maps," *IEEE Trans. on Neural Networks*, vol. 3, no. 5, pp. 698–713, 1992.

[8] R. Chattopadhyay, S. Chakraborty, V.N. Balasubramanian, and S. Panchanathan, "Optimization-based domain adaptation towards person-adaptive classification models," in *IEEE Computer Society Proc. 10th Intl. Conf. on Machine Learning and Applications (ICMLA, 2011)*, Honolulu, HI, Dec. 18-21, 2011, pp. 476–483.

[9] X.-W. Chen and T. Huang, "Facial expression recognition: a clustering-based approach," *Pattern Recognition Letters*, vol. 24, no. 9-10, pp. 1295–1302, 2003.

[10] F. Courtemanche, E. Aïmeur, A. Dufresne, M. Najjar, and F. Mpondo, "Activity recognition using eye-gaze movements and traditional interactions," *Interacting with Computers*, vol. 23, no. 3, pp. 202–213, 2011.

[11] F. Dornaika, E. Lazkano, and B. Sierra, "Improving dynamic facial expression recognition with feature subset selection," *Pattern Recognition Letters*, vol. 32, no. 5, pp. 740–748, 2011.

[12] R. Elwell and R. Polikar, "Incremental learning of concept drift in nonstationary environments," *IEEE Transactions on Neural Networks*, vol. 22, no. 10, pp. 1517–1531, 2011.

[13] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006.

[14] M. Graña, "A brief review of lattice computing," in *Proceedings of the World Congress on Computational Intelligence (WCCI) 2008, FUZZ-IEEE Program*, , Hong Kong, China, 1-6 June 2008, pp. 1777–1781.

[15] M. Graña (Ed.), Special issue on: Lattice computing and natural computing, *Neurocomputing*, vol. 72, no. 10-12, pp. 2065–2066, 2009.

[16] M. Graña, D. Chyzhyk, M. García-Sebastián, and C. Hernández, "Lattice independent component analysis for functional magnetic resonance imaging," *Information Sciences*, vol. 181, no. 10, pp. 1910–1928, 2011.

[17] W. Gu, C. Xiang, Y.V. Venkatesh, D. Huang, and H. Lin, "Facial expression recognition using radial encoding of local Gabor features and classifier synthesis," *Patt. Recogn.*, vol. 45, no. 1, pp. 80–91, 2012.

[18] Z. Hammal and M. Kunz, "Pain monitoring: a dynamic and context-sensitive system," *Patt. Recogn.*, vol. 45, no. 4, pp. 1265–1280, 2012.

[19] M. Ilbeygi and H. Shah-Hosseini, "A novel fuzzy facial expression recognition system based on facial feature extraction from color face images," *Eng. Applic. of Artif. Intel.*, vol. 25, no. 1, pp. 130–146, 2012.

[20] V.G. Kaburlasos, *Adaptive Resonance Theory With Supervised Learning and Large Database Applications*. Ph.D. Dissertation, Univ. Nevada, Reno, 1992, USA Library of Congress-Copyright Office.

[21] V.G. Kaburlasos, "FINs: lattice theoretic tools for improving prediction of sugar production from populations of measurements," *IEEE Trans. on Syst., Man & Cybern. - B*, vol. 34, no. 2, pp. 1017–1030, 2004.

[22] V.G. Kaburlasos, *Towards a Unified Modeling and Knowledge-Representation Based on Lattice Theory*. Heidelberg, Germany: Springer, 2006, ser. Studies in Computational Intelligence, vol. 27.

[23] V.G. Kaburlasos (Ed.), Special issue on: Information engineering applications based on lattices, *Information Sciences*, vol. 181, no. 10, pp. 1771–1773, 2011.

[24] V.G. Kaburlasos and A. Kehagias, "Novel fuzzy inference system (FIS) analysis and design based on lattice theory," *IEEE Transactions on Fuzzy Systems*, vol. 15, no. 2, pp. 243–260, 2007.

[25] V.G. Kaburlasos and T. Pachidis, "A lattice-computing ensemble for reasoning based on formal fusion of disparate data types, and an industrial dispensing application," *Information Fusion*, (to be published)

[26] V.G. Kaburlasos and S.E. Papadakis, "Granular self-organizing map (grSOM) for structure identification," *Neural Networks*, vol. 19, no. 5, pp. 623–643, 2006.

[27] V.G. Kaburlasos and S.E. Papadakis, "A granular extension of the fuzzy-ARTMAP (FAM) neural classifier based on fuzzy lattice reasoning (FLR)," *Neurocomputing*, vol. 72, no. 10-12, pp. 2067–2078, 2009.

[28] V.G. Kaburlasos and V. Petridis, "Fuzzy lattice neurocomputing (FLN) models," *Neural Networks*, vol. 13, no. 10, pp. 1145–1169, 2000.

[29] V.G. Kaburlasos, I.N. Athanasiadis, and P.A. Mitkas, "Fuzzy lattice reasoning (FLR) classifier and its application for ambient ozone estimation," *Intl. J. of Approximate Reasoning*, vol. 45, no. 1, pp. 152–188, 2007.

[30] V.G. Kaburlasos, S.E. Papadakis, and A. Amanatiadis, "Binary image 2D shape learning and recognition based on lattice-computing (LC) techniques," *J. Math. Imag. & Vis.*, vol. 42, no. 2-3, pp. 118-133, 2012.

[31] Ath. Kehagias, "Some remarks on the lattice of fuzzy intervals," *Information Sciences*, vol. 181, no. 10, pp. 1863–1873, 2011.

[32] S.M. Lajevardi and Z.M. Hussain, "Higher order orthogonal moments for invariant facial expression recognition," *Digital Signal Processing*, vol. 20, no. 6, pp. 1771–1779, 2010.

[33] O. Langner, R. Dotsch, G. Bijlstra, D.H.J. Wigboldus, S.T. Hawk, and A. van Knippenberg, "Presentation and validation of the Radboud faces database," *Cognition & Emotion*, vol. 24, no. 8, pp. 1377–1388, 2010.

[34] M.J. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba, "Coding facial expressions with Gabor wavelets," in *IEEE Computer Society Proc. 3rd Intl. Conf. on Automatic Face and Gesture Recognition (FG '98)*, Nara, Japan, April 14-16, 1998, pp. 200–205.

[35] L. Ma and K. Khorasani, "Facial expression recognition using constructive feedforward neural networks," *IEEE Transactions on Systems, Man & Cybernetics - Part B*, vol. 34, no. 3, pp. 1588–1595, 2004.

[36] J.M. Mendel, "Type-2 fuzzy sets and systems: an overview," *IEEE Computational Intelligence Magazine*, vol. 2, no. 1, pp. 20–29, 2007.

[37] L.L. Minku and X. Yao, "DDD: a new ensemble approach for dealing with concept drift," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 4, pp. 619–633, 2012.

[38] T.D. Nguyen and S. Ranganath, "Facial expressions in American sign language: tracking and recognition," *Pattern Recognition*, vol. 45, no. 5, pp. 1877–1891, 2012.

[39] M. Pantic, Keynote Speech on "Machine understanding of human behaviour," *16th Intl. Conf. on Knowledge-Based and Intelligent Information & Engineering Systems (KES-2012)*, San Sebastian, Spain, 10-12 September, 2012.

[40] M. Pantic, A. Nijholt, A. Pentland, and T.S. Huang, "Human-centred intelligent human-computer interaction (HCI2): how far are we from attaining it?," *International Journal of Autonomous and Adaptive Communications Systems*, vol. 1, pp. 168–187, 2008.

[41] S.E. Papadakis and V.G. Kaburlasos, "Piecewise-linear approximation of non-linear models based on probabilistically/possibilistically interpreted intervals' numbers (INs)," *Information Sciences*, vol. 180, no. 24, pp. 5060–5076, 2010.

[42] S.E. Papadakis, V.G. Kaburlasos, and G.A. Papakostas, "Fuzzy lattice reasoning (FLR) classifier for human facial expression recognition," in *Proc. 10th Intl. Conf. on Uncertainty Modeling in Knowledge Engineering and Decision Making (FLINS)*, Istanbul, Turkey, 26-29 Aug 2012. World Scientific on Comp. Eng. and Info. Science 7, pp. 633–638.

[43] G.A. Papakostas, E.G. Karakasis, and D.E. Koulouriotis, "Novel moment invariants for improved classification performance in computer vision applications," *Pattern Recognition*, vol. 43, no. 1, pp. 58–68, 2010.

[44] G.A. Papakostas, D.E. Koulouriotis, and E.G. Karakasis, "A unified methodology for efficient computation of discrete orthogonal image moments," *Information Sciences*, vol. 179, no. 20, pp. 3619–3633, 2009.

[45] F.Y. Shih, C.-F. Chuang, and P.S.P. Wang, "Performance comparisons of facial expression recognition in JAFFE database," *Intl. J. of Pattern Recognition and Artificial Intelligence*, vol. 22, no. 3, pp. 445–459, 2008.

[46] P. Sussner and E.L. Esmi, "Morphological perceptrons with competitive learning: Lattice-theoretical framework and constructive learning algorithm," *Information Sciences*, vol. 181, no. 10, pp. 1929–1950, 2011.

[47] M.F. Valstar, M. Mehu, B. Jiang, M. Pantic, and K. Scherer, "Meta-analysis of the first facial expression recognition challenge," *IEEE Transactions on Systems, Man & Cybernetics - Part B*, vol. 42, no. 4, pp. 966–979, 2012.

[48] P. Viola and M.J. Jones, "Robust real-time face detection," *Intl. Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[49] C. Zhan, W. Li, P. Ogunbona, and F. Safaei, "A real-time facial expression recognition system for online games," *International Journal of Computer Games Technology*, Article ID 542918, 7 pages, 2008.