2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)
July 6-11, 2014, Beijing, China

Lattice Computing (LC) Meta-representation for Pattern Classification

George A. Papakostas and Vassilis G. Kaburlasos

Abstract— **This paper compares two alternative feature data meta-representations using Intervals' Numbers (INs) in the context of the *Minimum Distance Classifier (MDC)* model. The first IN meta-representation employs one IN per feature vector, whereas the second IN meta-representation employs one IN per feature per class. Comparative classification experiments with the standard minimum distance classifier (MDC) on two benchmark classification problems, regarding face/facial expression recognition, demonstrate the superiority of the aforementioned second IN meta-representation. This superiority is attributed to an IN's capacity to represent discriminative, all-order data statistics in a population of features.**

I. Introduction

An important part of each modern intelligent system, able to interact with its environment and to take decisions about its content, is the process of *pattern classification*. This processing gives human-like intelligent abilities to the system being incorporated, in terms of perception, understanding and adaptability. Based on this mechanism several systems have been proposed in the past for video surveillance [1], security [2], entertainment [3], biomedicine [4], etc.

In order to accomplish the pattern classification procedure, the patterns have to be represented appropriately and fed on a specific classification model called *classifier*. The knowledge about the patterns is commonly represented by numerical feature vectors which for the case of image patterns reduce the dimensionality of the useful information. Considering the case of image patterns, several features have been used such as shape descriptors [5], moment features [6][7], histograms of measures [8][9], etc. The formed feature vectors are used in a next stage to train a classification model e.g. neural network, k-NN, Naive-Bayes classifiers, providing the overall system with the decision capability.

A different approach in representing the patterns and classifying them, showing promising results has been proposed into the Lattice Computing (LC) paradigm [10]-[13].

In this context, the initial feature vectors constituting an approximation of the images are transformed to Interval's Numbers (INs). This meta-representation enables the usage of some useful tools defined in lattice space, such as distance and similarity measures, able to distinguish the patterns of each class. Based on these distance and similarity measures, traditional classifier models have been adopted [10]-[14] and applied in pattern classification applications with success.

Considering the task of pattern classification by using lattice computing, this work investigates the impact of the INs construction to the classification accuracy of the designed classifier. To this end, two possible construction strategies are studied by highlighting the main properties of each design strategy. Moreover, through appropriate experiments useful conclusions, regarding the discrimination capability of the constructed INs types, are drawn. The outcomes of this study will constitute a useful guide to everyone who desires to solve a pattern classification problem into the lattice computing paradigm.

This paper is organized as follows: Section II summarizes the theory of the INs meta-representation defined inside the lattice computing framework. In this section the two possible INs construction strategies are analyzed. Section III describes the design principles of a MDC classifier in the LC context. Section IV presents some experimental results with well-known benchmark datasets. Finally, section V summarizes the main conclusions of this study.

II. Interval's numbers (INs)

A lattice is a partially ordered set $(L,\sqsubseteq)$. In a lattice any two elements $x, y \in L$ have both a greatest lower bound (*infimum*), denoted by $x \wedge y$ and a least upper bound (*supremum*), denoted by $x \vee y$. The $(L,\sqsubseteq)$ lattice is called *totally-ordered* if every (x, y) pair satisfies either $x \sqsubseteq y$ or x⊐ y. Furthermore, the $(L,\sqsubseteq)$lattice is called *complete* iff each of its subsets X has both an *infimum* and a *supremum* in L [15].

Considering the above definitions several tools have been developed in the LC framework for solving difficult engineering problems. Our interest focuses on a complete lattice of Intervals' Numbers (INs) [10] defined hereafter.

George A. Papakostas and Vassilis G. Kaburlasos are with the Department of Computer and Informatics Engineering, Eastern Macedonia and Thrace Institute of Technology, Human Machines Interaction (HMI) Laboratory, 65404 Kavala, Greece (e-mail: gpapak@teikav.edu.gr, vgkabs@teikav.edu.gr).

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.

978-1-4799-2072-3/14/$31.00 ©2014 IEEE

A. Data Representation by Interval's Numbers (INs)

A simple well established algorithm (CALCIN) exists [10][13] for the calculation of INs as a data representation utility. Herein, the INs calculation procedure is briefly described, for the general case of a population of data samples (measures).

Suppose a vector $\vec{\mathbf{x}}$ of real (pair wise different) data samples, e.g. $\vec{\mathbf{x}}=(x_1,\ldots,x_n)$. Two entries x_i, x_j of the vector $\vec{\mathbf{x}}$ are called successive iff there is no other entry $x_k, k\{1,...,n\}$ with $x_i \wedge x_j < x_k < x_i \vee x_j$ ($\wedge$: *min*, $\vee$: *max*).

Moreover, a strictly-increasing, cumulative real function $c:\Re\rightarrow\Re_0^+$ is computed from the entries of vector $\vec{\mathbf{x}}$ by, first, defining

$$c(x_i)=\frac{1}{n}\left|\left\{x_j : j\in\{1,...,n\} \text{ and } x_j\le x_i\right\}\right|,\ i\in\{1,...,n\} \qquad (1)$$

where $|S|$ denotes the cardinality of the set S; finally, function $c:\Re\rightarrow\Re_0^+$ is defined by straight-line connecting any two points $(x_i,c(x_i))$ and $(x_j,c(x_j))$, where x_i,x_j are successive entries of vector $\vec{\mathbf{x}}$. Obviously, there is a unique real number x_0 such that $c(x_0)=0.5$. In conclusion, an IN is calculated from function $c(\cdot)$ such as for values less-than or equal-to x_0 the corresponding IN envelope function is $2c(x)$, whereas for values larger than x_0 the corresponding IN envelope function is $2(1-c(x))$. An IN envelope is represented by a user defined number N_h equally spaced intervals (*h-cuts*) from *h=0* to *h=1* and thus any population of data samples in vector $\vec{\mathbf{x}}$ can be represented by N_h intervals stored in a N_h x2 size matrix. A commonly used number of intervals is 32 (*N*N_h*=32*) by resulting to a 32x2 length representation. It is worth noting that the IN meta-representation includes all order data statistics of the initial population.

B. Construction of INs for pattern classification

The calculation of an IN, as described in the previous section, takes into account only the distribution of the data samples inside the population. However, for the case of pattern classification problems an additional requirement has to be also satisfied. For this case it is essential to describe the prototype of each class in a sense that patterns of the same class are close to their prototype and far away from the prototypes of the other classes. This need also agrees with the Fisher criterion [16] considering the intra-class and inter-class separability of the classes.

Therefore, it is evident that special attention has to be paid when the INs are constructed from populations of features (feature vectors), with several populations for each class. In order to investigate this process the three artificial patterns $\{p_1^k, p_2^k, p_3^k\}$ for the k^{th} class of Fig. 1 are considered.

	f_1	f_2	f_3	f_4
p_1^k	1.34	0.67	2.11	4.32
p_2^k	1.26	0.59	2.04	4.38
p_3^k	1.29	0.71	2.07	4.28

Figure 1. Artificial patterns of the kth class.

Each pattern is described by a feature vector having four features $\{f_1,f_2,f_3,f_4\}$.

It is note that each class is represented by a number of patterns described by equally sized feature vectors. The aforementioned patterns will be used to investigate two different INs calculation approaches described hereafter.

1) Approach 1 - INs per Feature Vector (pFV)

Considering that each pattern is represented by a feature vector of specific length (equal to the number of used features), the simplest way to construct an IN is by tackling each feature vector as a population of samples. This approach is called *INs per Feature Vector (pFV)* for the rest of this work. In this case the classes' prototypes are selected to be a specific IN that better describes the patterns distribution. The corresponding INs constructed by this approach for the case of the patterns of Fig. 1 are depicted in Fig.2. These INs are derived by applying the CALCIN algorithm (Eq.(1)) to the data rows of Fig.1.

The ability of IN's to provide an IN meta-representation of any size for data vector $\vec{\mathbf{x}}$, make this meta-representation appealing instrument to deal with big (numeric) data, such as in Alzheimer's disease detection [17][18], where initial feature vector sizes are of the order of several hundreds or even thousands.

Therefore, it can be claimed that this INs construction strategy performs a dimensionality reduction to the initial feature vectors. This functionality is achieved by the following rule of thumb "$N_h \ll (DataSize/2)$" where *DataSize* is the size of the initial population data samples. Although, the *Datasize* of the examined test patterns are 4 and a number of N_h =32 *h-cuts* is usually computed [12]-[15], several N_h numbers up to *DataSize* can be studied for the cases of long feature vectors.

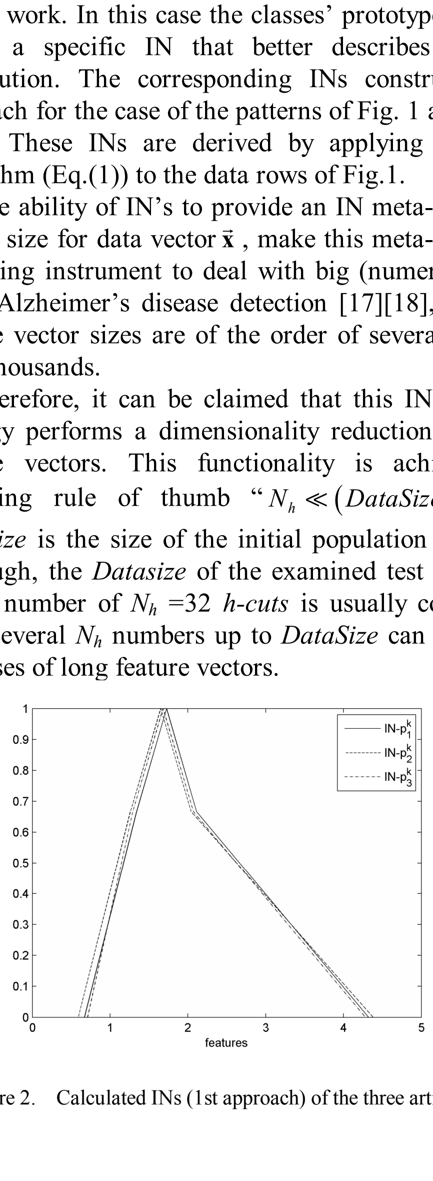

Figure 2. Calculated INs (1st approach) of the three artificial patterns.

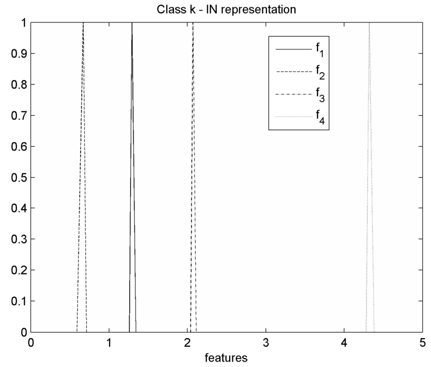

Figure 3. Calculated INs (2nd approach) of the three artificial patterns.

2) Approach 2 -INs per Feature (pF)

The aforementioned per feature vector INs construction, is useful for high length initial feature vectors, since it applies a dimensionality reduction simultaneously. However, from the pattern classification point of view this reduction is in some sense "*blind"* since it does not consider the classes' separability to perform the reduction. Moreover, a closer look at the constructed INs of Fig.2 gives the additional information of the existence of a small tolerance among the patterns of the same class. This tolerance is inherited by the initial feature vectors and corresponds to the different conditions under which these features were derived. Such variations are responsible for the degradation of classifiers' accuracy.

Another alternative of INs meta-representation design is the construction of an individual IN per feature of all the patterns, called *INs per Feature (pF)*. In this way, the resulted INs enclose all the variations imposed to each feature, increasing their robustness to different acquisition conditions.

Following this approach, an IN is constructed for each feature thus the number of INs representing each class's prototype is equal to the length of the used feature vector. Surely, it is not efficient to apply this approach for high dimension feature vectors, but the generalization capability of the classifier model is expected to be improved.

The corresponding INs of the artificial patterns (Fig.1) are illustrated in Fig.3. From this figure it is obvious that the constructed INs are more narrow compared to the INs of Fig.2 due to the fact that they describe the distribution of each feature rather the entire feature vectors of the patterns.

C. Discussion

To summarize, each one of the above different INs meta-representation design alternatives shows some useful properties making them appropriate for specific applications. Based on the above theoretical analysis the *pFV* approach is applicable for high length feature vectors, while the *pF* is more efficient for small feature vectors. The *pFV* scheme is expected to be less accurate in terms of classification performance, while the *pF* approach is more robust to varying acquisition conditions.

In the next sections, these two design schemes will be studied under the same classification configuration, in classifying the patterns of two benchmark classification datasets.

III. LC-Based Classification

Since the introduction of the LC, a lot of work has been reported in reformulating some traditional classifier models to operate according to LC framework [10]-[14]. Among these models the *Minimum Distance Classifier (MDC)*, which is equivalent with the 1-NN classifier, is selected for this study due to its simplicity.

An extension of the MDC in the LC context making use of the INs representation is described in this section. A traditional MDC assigns a test sample (S) to the class (k^*) represented by the prototype pattern P_{k^*} such that P_{k^*} corresponds to the minimum distance; symbolically,

$$k^* = \arg\min_k \{ Dist(P_k, S) \} \tag{2}$$

where P_k is the single prototype pattern of class k and *Dist(.,.)* is a distance function that quantifies the "apartness" of P_k and *S*.

A. LC-based Minimum Distance Classifier (LC-MDC)

The above MDC classifier can be reformulated in the LC framework by incorporating a distance (metric) over the meta-representation of the features as INs. In this case both a class prototype pattern P_k and a test sample S are in the Cartesian product $\mathbf{F}^N$, where $\mathbf{F}$ is the lattice of INs. Such a distance is defined as follows:

Definition 1. A distance (metric) [19] d in a nonempty set A is a real function $d : A \times A \to [0, +\infty)$ which satisfies the following conditions, $\forall x, y, z \in A$:

C1. $d(x,y) = 0 \Leftrightarrow x = y$ (coincidence)

C2. $d(x,y) = d(y,x)$ (symmetry)

C3. $d(x,z) + d(z,y) \geq d(x,y)$ (triangle inequality)

A distance function $d_{\mathbf{F}} : \mathbf{F} \times \mathbf{F} \to \mathfrak{R}_0^+$ in the lattice F of INs is defined [13] as:

$$d_{\mathbf{F}}(F,G) = \int_0^1 d_{\mathbf{I}}(F_h, G_h)\, dh \tag{3}$$

where the function $d_{\mathbf{I}} : \mathbf{I} \times \mathbf{I} \to \mathfrak{R}_0^+$ is a distance given by

$$\begin{aligned} d_{\mathbf{I}}([a,b],[c,d]) &= [v(\theta(a \wedge c)) - v(\theta(a \vee c))] \\ &+ [v(b \vee d) - v(b \wedge d)]; \end{aligned} \tag{4}$$

where $v : \mathfrak{R} \to \mathfrak{R}_0^+$ and $\theta : \mathfrak{R} \to \mathfrak{R}$ are strictly increasing and decreasing functions, respectively.

In conclusion, a distance function $d:\mathbf{F}^N \times \mathbf{F}^N \rightarrow \Re_0^+$ is given by

$$d(\mathbf{F},\mathbf{G}) = d\left(\left(F_1,...,F_N\right),\left(G_1,...,G_N\right)\right) = \sum_{i=1}^{N} d_{\mathbf{F}}\left(F_i,G_i\right) \tag{5}$$

Note that the following functions $v:\Re \rightarrow \Re_0^+$ and $\theta:\Re \rightarrow \Re$ have been used in Eq.(4).

$$v(x) = \frac{A}{1+e^{-\lambda(x-\mu)}} \text{ and } \theta(x) = 2\mu - x \tag{6}$$

where $A,\lambda \in \Re^+$ and $\mu \in \Re$ are free parameters. These parameters increase flexibility of the distance function when comparing two INs at different resolutions.

B. Parameter Estimation

The three parameters (A,λ,μ) of Eq.(6) need to be tuned appropriately since they control the discrimination ability of the LC-based MDC classifier.

Due to the lack of analytical expression of the classification error as a function of these free parameters, a stochastic optimization algorithm seems to be a good choice. For this purpose, optimization techniques such as Genetic Algorithms (GAs) [20], Particle Swarm Optimization (PSO) [21], etc., could be applied. Herein, a simple GA is incorporated having the following settings: 100 chromosomes population, crossover and mutation probabilities 0.8 and 0.01 respectively, 1 elite chromosome and stochastic universal approximation selection method; moreover, the algorithm was executed for 100 generations.

It is worthwhile noting that genetic algorithm's chromosomes consist of the free parameters of Eq.(6). For the case of the *pFV* scheme the tuned parameters are three (A,λ,μ), while for *pF* are the three previous parameters per feature resulting in $3\times$ (NumberOfFeatures) parameters totally. The fitness function being optimized is the same for both strategies and corresponds to the classification error (WrongClassifiedPatterns /TotalPatterns) during the training stage.

IV. Experiments

A set of experiments were conducted in order to evaluate the two different INs construction schemes. To this end, appropriate software was developed in MATLAB 2012b environment, while all experiments were executed in an Intel i5 3.3GHz PC with 8GB RAM.

A. Datasets Description

Two typical pattern classification problems sharing common attributes and attracting the scientists' interest in the recent years, a human face and facial expression recognition were selected as test bench to study the classification performance of *pFV* and *pF* schemes. For the experiments two popular benchmark datasets of the above mentioned problems were used the YALE [22] and JAFFE [23]. The first one consists of 165 images of 15 individuals in 11 different instances. The latter consists of 213 images describing the seven basic human expressions (neutral, angry, disgust, fear, happy, sad, and surprise) of several Japanese female models.

Moreover, an additional pre-processing task has to be performed to the initial images, in order to remove irrelevant image content such as background/hair. For this reason the Viola-Jones face detector [24] is applied to separate the head region from the background, and then by masking the face with an ellipse the hair is removed. As a result, the final images, including as much facial information as possible are used for feature extraction.

B. Features Extraction

From the face images of each dataset a set of feature vectors are extracted in order to reduce the carrying information about the faces by maintaining the discriminant information.

The method of orthogonal moments is selected as the feature extraction technique, due to its popularity and efficiency in representing the patterns. Orthogonal image moments have been applied in many disciplines of the engineering life, especially in pattern recognition and computer vision [25][26] with remarkable success.

Two popular moment families are used in this study the Zernike moments (ZMs) and Tchebichef moments (TMs). The first moment family (ZMs) belongs to the continuous radial moments, is rotation invariant and is probably the most used moment type in the literature. The latter moments (TMs) are defined to the discrete domain and thus is more accurate than the ZMs.

For the sake of the experiments the ZMs and TMs moment features are computed up to order 18 and 9 respectively. The computed moments are rearranged in the form of vectors by applying the zigzag scan scheme. The resulted feature vectors are of 100 sizes in both moment types.

C. Simulations

A "10-fold cross-validation" training scheme, according to which each dataset is partitioned in 10 equal parts (each class is represented fairly), is applied. Nine-tenths of the data are used for training, whereas the remaining one-tenth is used to test the trained classifier. The same procedure is applied until all tenths of the data are used for testing.

In order to better highlight the classification abilities of the two INs' construction schemes, the typical MDC model is used under two different configurations in comparison with the proposed LC-MDC models.

One important parameter that controls the operation of the MDC model is the selection of the classes' prototypes from which the samples' distances are measured. To this, end two tactics are commonly applied: the first sample (MDC-A) or the mean feature vector (MDC-B) of the training samples of each class is used as classes' prototypes.

Moreover, the well-known Euclidean distance is applied to measure the proximity of the samples.

According to the “10-fold cross-validation” the 90% training samples are used to construct the classes’ prototypes, while the rest 10% are used to test the generalization capability of the model. Furthermore, for the case of the LC-based models where a parameter estimation process is needed, the 10% of the samples are used to validate the constructed model, while the remaining 80% and 10% are used for training and testing purposes.

The following Tables I, II illustrate the corresponding experimental statistics (minimum (Min), maximum (Max), average (Ave) and standard deviation (Std)) in terms of classification rate (%), for the YALE and JAFFE datasets, respectively. In these tables, the LC^{pFV}-MDC and LC^{pF}-MDC, correspond to the LC-based MDC models applying the *pFV* and *pF* INs’ construction schemes, respectively.

TABLE I. CLASSIFICATION RATE (%) STATISTICS FOR YALE DATASET

Classifier	**Features**	**Classification Statistics**			
		Min	***Max***	***Ave***	***Std***
MDC-A	ZMs	20.00	73.33	42.00	15.41
	TMs	46.67	66.67	52.67	6.63
MDC-B	ZMs	26.67	100.00	78.67	29.28
	TMs	66.67	100.00	88.67	12.98
LC^{pFV}-MDC	ZMs	6.67	53.33	29.33	15.14
	TMs	13.33	66.66	36.66	15.47
LC^{pF}-MDC	ZMs	33.33	100.00	77.33	22.92
	TMs	66.67	100.00	85.33	13.98

TABLE II. CLASSIFICATION RATE (%) STATISTICS FOR JAFFE DATASET

Classifier	**Features**	**Classification Statistics**			
		Min	***Max***	***Ave***	***Std***
MDC-A	ZMs	9.52	38.09	20.57	8.40
	TMs	0.00	38.09	17.26	10.90
MDC-B	ZMs	20.00	42.86	29.14	8.11
	TMs	14.28	52.38	28.64	12.41
LC^{pFV}-MDC	ZMs	14.28	33.33	22.95	6.61
	TMs	9.52	28.57	17.24	6.46
LC^{pF}-MDC	ZMs	9.52	52.38	34.90	13.41
	TMs	14.29	57.14	30.14	13.65

From the above tables it is obvious that among the two different types of the traditional MDC, the MDC-B is more efficient (8.57%-35.67% better, on average), since the selection of the prototypes by averaging the samples’ feature vectors is more robust to classes’ variability.

Concerning the LC-based MDC models the *pF* INs’ construction scheme outperforms the *pFV* in both datasets significantly. The INs meta-representation of the feature vectors seems not to be highly disciminative, since the reduction applied along to the features (from 100 to 64) causes the rejection of some useful information. Therefore, the dimensionality reduction implied by the *pFV* strategy is useful with very long feature vectors, where there is significant information redundancy such as biomedical data [18]. For feature vectors of low dimensions, this scheme discards useful information by making the resulted INs less discriminative.

On the contrary, the *pF* scheme seems to be able to describe the tolerance of each feature along to its class and thus to provide a more robust meta-representation. This tolerance is occurred very frequently due to illumination, noise and environmental variations existing during the image patterns acquisition.

For the case of the YALE dataset, the LC^{pF}-MDC classifier performs competitively to MDC-B, with the latter being slightly more accurate on average. However, this dataset is quite easy and it seems that the mean value of the feature vectors is adequate to distinguish the classes. However, when more difficult JAFFE dataset is considered, all order statistics enclosed by the INs meta-representation of the LC^{pF}-MDC are somewhat necessary to accurate classifying the patterns. Furthermore, the existence of the adjustable distance metric used to distinguish the patterns, give more degrees of freedom to the overall classification process and enables the adaptation of the LC^{pF}-MDC to different applications.

Summarizing, the *pF* INs construction scheme has shown superior classification accuracy compared to *pFV* by establishing this strategy the appropriate choice when dealing with classification in the LC paradigm. Although, the *pF* scheme was applied on the MDC model, giving promising results, it can be potentially incorporated into any lattice-based classifier.

V. CONCLUSION

A detailed analysis of the INs meta-representation in the LC context was presented in the previous sections. To this end, two different alternatives for INs construction were analyzed and studied by incorporating them to the MDC model. Appropriate experiments shown the superiority of the *pF* scheme compared to the *pFV*, due to its robustness to the classes’ separability.

Moreover, all order statistics enclosed in the INs constructed by the *pF* scheme make the corresponding MDC model competitive to the traditional classifiers. The *pF* INs meta-representation can be used to any lattice based classifier in order to improve the discrimination ability of the initial feature vectors and the generalization performance of the classification model.

REFERENCES

[1] W. Niu, J. Long, D. Han, Y.-F. Wang, "Human activity detection and recognition for video surveillance," *IEEE International Conference on Multimedia and Expo (ICME '04)*, vol.1, pp.719-722, 30 June, Taipei - Taiwan, 2004.

[2] W.W. Boles, “A security system based on human iris identification using wavelet transform,” *Engineering Applications of Artificial Intelligence*, vol. 11, no. 1, pp. 77–85, 1998.

[3] C. Zhan, W. Li, P. Ogunbona, and F. Safaei, “A real-time facial expression recognition system for online games,” *Int. J. Comput. Games Technol.*, vol. 2008, pp. 542918-1–542918-7, Jan. 2008.

[4] A.J. Wright, G. Fellows, T.J. Byrnes, K.S. Opstad, D.J.O. McIntyre, J.R. Griffiths, B.A. Bell, C.A. Clark, T. R. Barrick, F.A. Howe “Pattern recognition of MRSI data shows regions of glioma growth that agree with DTI markers of brain tumor infiltration,” *Magnetic Resonance in Medicine*, vol. 62, no. 6, pp. 1646-1651, 2009.

[5] D. Shen, Horace H.S. Ip, “Discriminative wavelet shape descriptors for recognition of 2-D patterns,” *Pattern Recognition*, vol. 32, no. 2, pp. 151–165, 1999.

[6] G.A. Papakostas, Y.S. Boutalis, D.A. Karras, B.G. Mertzios, “Pattern classification by using improved wavelet compressed Zernike moments,” *Applied Mathematics and Computation*, vol. 212, no. 1, pp. 162-176, 2009.

[7] G.A. Papakostas, Y.S. Boutalis, D.A. Karras, B.G. Mertzios, “Efficient computation of Zernike and Pseudo-Zernike moments for pattern classification applications,” *Pattern Recognition and Image Analysis*, vol. 20, no. 1, pp. 56-64, 2010.

[8] T. Ahonen, A. Hadid, M. Pietikainen, "Face description with local binary patterns: application to face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.28, no.12, pp.2037-2041, 2006.

[9] X. Wang, T. X. Han, S. Yan, “An HOG-LBP human detector with partial occlusion handling”, *IEEE International Conference on Computer Vision (ICCV'2009)*, pp. 32-39, Kyoto, Japan, October, 2009.

[10] V. G. Kaburlasos, *Towards a Unified Modeling and Knowledge-Representation Based on Lattice Theory* (Studies in Computational Intelligence). Heidelberg, Germany: Springer-Verlag, 2006.

[11] 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, nos. 10-12, pp. 2067–2078, 2009.

[12] V.G. Kaburlasos, S.E. Papadakis and G.A. Papakostas, “Lattice computing extension of the FAM neural classifier for human facial expression recognition,” *IEEE Transactions on Neural Networks & Learning Systems*, vol. 24, no. 10, pp. 1526 - 1538, 2013.

[13] G.A. Papakostas, V.G. Kaburlasos and Th. Pachidis, “Thermal infrared face recognition based on Lattice Computing (LC) techniques,” *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2013)*, pp. 1-6, 7-10 July, Hyderabad - India, 2013.

[14] V.G. Kaburlasos and L. Moussiades, “Induction of formal concepts by lattice computing techniques toward tunable classification,” *Journal of Engineering Science and Technology Review*, vol. 7, no. 1, pp. 1-8, 2014.

[15] V. G. Kaburlasos and A. Kehagias, “Fuzzy inference system (FIS) extensions based on lattice theory,” *IEEE Transactions on Fuzzy Systems*, in press, doi: 10.1109/TFUZZ.2013.2263807.

[16] R.A. Fisher, “The use of multiple measurements in taxonomic problems,” *Annals of Eugenics*, vol. 7, no. 2, pp. 179-188, 1936.

[17] D. Chyzhyk, M. Grana, A. Savio, J. Maiora, “Hybrid dendritic computing with kernel-LICA applied to Alzheimer's disease detection in MRI,” *Neurocomputing*, vol. 75, no. 1, pp. 72-77, 2012.

[18] G.A. Papakostas, A. Savio, M. Grana and V.G. Kaburlasos, “A lattice computing approach to Alzheimer’s disease computer assisted diagnosis based on MRI data,” *Neurocomputing*, accepted.

[19] G.A. Papakostas, A.G. Hatzimichailidis and V.G. Kaburlasos, “Distance and similarity measures between intuitionistic fuzzy sets: A comparative analysis from a pattern recognition point of view,” *Pattern Recognition Letters*, vol. 34, no. 14, pp. 1609-1622, 2013.

[20] G.A. Papakostas, E.D. Tsougenis, D.E. Koulouriotis and V.D. Tourassis, “Moment-based local image watermarking via genetic optimization”, *Applied Mathematics and Computation*, vol. 227, pp. 222-236, 2014.

[21] S.E. Papadakis, V.G. Kaburlasos and G.A. Papakostas, “Two Fuzzy Lattice Reasoning (FLR) classifiers and their application for human facial expression recognition,” *Journal of Multi-Valued Logic and Soft Computing*, vol. 22, pp. 561-579, 2014.

[22] Yale face database, http://vision.ucsd.edu/content/yale-face-database (access date: 04/01/2013).

[23] M.J. Lyons, S. Akamatsu, M. Kamachi, J. Gyoba, “Coding facial expressions with Gabor wavelets,” *3rd IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 200-205, 1998.

[24] P. Viola and M. J. Jones, “Robust real-time face detection,” *Int. J. Comput. Vis.*, vol. 57, no. 2, pp. 137–154, 2004.

[25] J. Flusser, T. Suk, B. Zitova, *Moments and moment invariants in pattern recognition*, John Wiley and Sons, 2009.

[26] G.A. Papakostas, E.G. Karakasis, D.E. Koulouriotis, “Orthogonal image moment invariants: highly discriminative features for pattern recognition applications”, *Cross-Disciplinary Applications of Artificial Intelligence and Pattern Recognition: Advancing Technologies*, Vijay Kumar Mago and Nitin Bhatia (Eds.), IGI Global, pp. 34-52, 2012.