J Math Imaging Vis manuscript No.
(will be inserted by the editor)

Binary Image 2D Shape Learning and Recognition Based on Lattice-Computing (LC) Techniques

Vassilis G. Kaburlasos · S. E. Papadakis · Angelos Amanatiadis

Received: date / Accepted: date

Abstract This work introduces a *Type-II fuzzy lattice reasoning (FLRtypeII)* scheme for learning/generalizing novel 2D shape representations based "in principle" on a fuzzy lattice *inclusion measure* function. A 2D shape is represented as an element – induced from populations of three different shape descriptors – in the product lattice $(\mathsf{F}^3, \preceq)$, where $(\mathsf{F}, \preceq)$ denotes the lattice of Type-I *intervals numbers* (*INs*). Learning is pursued by inducing Type-II INs, i.e. intervals in $(\mathsf{F}, \preceq)$. Our proposed techniques compare well with alternative classification methods from the literature in three benchmark classification problems. Competitive advantages include the accommodation of granular data as well as the visual representation of a class. We discuss potential extensions to gray/color images, etc.

Keywords Fuzzy lattice reasoning · Granular data · Inclusion measure · Intervals' number · Lattice computing · Learning · Pattern recognition · 2D shape representation · Sigmoid function

V. G. Kaburlasos
Dept. of Industrial Informatics, TEI of Kavala, 65404 Greece
Tel.: +30-2510-462320
Fax: +30-2510-462348
E-mail: vgkabs@teikav.edu.gr

S. E. Papadakis
E-mail: spap@teikav.edu.gr

A. Amanatiadis
E-mail: aaman@teikav.edu.gr

1 Introduction

Lattice theory (LT) [6] has been an instrument in applied mathematics including mathematical morphology (MM) [41], formal concept analysis (FCA) [14] and logic [49]. Lately, LT has been proposed as a sound framework for both mathematically rigorous modeling and knowledge-representation [19] with emphasis in information engineering applications [20] according to the following rationale.

A number of disparate data types, of practical interest, including matrices of real numbers, (cumulative) functions, sets/partitions, logic values, data structures, (strings of) symbols, etc. are lattice(partially)-ordered. Moreover, different authors have acknowledged that information *granules* [38,50] are lattice-ordered [19,43]. Hence, LT emerged as a sound framework for rigorous analysis and design involving, either separately or jointly in any combination, disparate types of data including numeric and/or nonnumeric ones. The interest of this work is in pattern recognition on digital (binary) images based on intervals, i.e. granules, of cumulative function representations.

LT in image processing is typically employed in the context of MM [39], where MM techniques depend on the *algebra-based* definition of a mathematical lattice including the operations of *meet* ($\wedge$) and *join* ($\vee$) as explained in [36]. It turns out that typical MM techniques, directly applicable on images, can perform well in noise filtering applications [44]; however, the aforementioned techniques do not satisfy any translation /rotation /scale -invariant pattern recognition requirements.

The unifying potential of LT for disparate data unification/fusion has been, partly, recognized in MM. For instance, a number of authors have established connections between mathematical morphology and fuzzy set theory [7,8,10,12, 33,42]. Other authors have applied lattice indepedent techniques on vectors of features extracted from images [18]. Moreover, largely in the context of MM, the term *Lattice-Computing* (LC) has been proposed to denote any computation in a mathematical lattice [15]. Graña and colleagues have demonstrated a number of LC techniques in signal/image processing applications [16–18]. Nevertheless, conventional MM fails to take any advantage of the *semantics* inherent in a lattice (partial) order relation.

This work proposes a complementary, to standard MM, technology towards the development of mathematically rigorous image processing techniques based on the partial order semantics in a lattice. From a technical point of view, we propose the manipulation of intervals in mathematical lattices, induced from real numbers, based on both positive valuations functions and dual isomorphic functions.

Of particular interest here is *fuzzy lattice reasoning* (*FLR*). The latter is a term originally proposed for denoting a specific rule-based reasoning scheme for clas-

sification in a general complete lattice $(\mathsf{L}, \preceq)$ data domain based on an *inclusion measure* function $\sigma : \mathsf{L} \times \mathsf{L} \to [0, 1]$ as explained in [19]. Recently, the scope of FLR was widened by including any decision-making based on an *inclusion measure* function [22]. Note that, on its introduction, the FLR was applied exclusively in lattice $(\mathsf{R}^N, \leq)$, that is the (conventional) Euclidean space [28]. Later work extended the FLR to the product lattice $(\mathsf{F}^N, \preceq)$, where $(\mathsf{F}, \preceq)$ in the lattice of Intervals' Numbers (INs), in pattern recognition /system modeling /reasoning applications [19,21–25,36]. This work further extends FLR to the lattice $(\tau_O(\mathsf{F}), \preceq)$ of intervals in $(\mathsf{F}, \preceq)$.

Advantages of FLR include the accommodation of granular data, introduction of tunable nonlinearities and induction of descriptive decision-making knowledge from the data. In its capacity, a novel FLR scheme is introduced here in the domain of image processing for learning/recognizing 2D shape patterns on binary images.

The study of 2D shapes on digital images has been popular in the literature [5,30,47] including, in particular, 2D shape recognition [32,46]. A popular representation of a 2D shape is by a *shape descriptors* vector $\overrightarrow{x_d} = (x_1, \ldots, x_{N_d})$, where d stands for an index denoting the corresponding descriptor extraction algorithm. What is typically required from shape descriptors is a capacity to accurately reconstruct the shape they represent, especially following translation/ scale /rotation transformations [3]. Note that shape descriptors can be characterized by either *quantitative* or *qualitative* indices of performance [4].

A number of shape descriptors, introduced in the literature, fall in one of two major groups, namely *contour-based* (shape descriptors) and *region-based* ones [52]. The former are typically extracted from the contour (boundary) of a shape, whereas the latter are typically extracted from the whole shape region. On the one hand, popular contour-based shape descriptors include the *Fourier Descriptors* (*FD*), which have been especially useful in character recognition applications; their advantages include simple derivation, robustness to noise, etc. [51]. On the other hand, popular region-based shape descriptors include the *Image Moments* (*IM*) derived by an affine transformation decomposition to six single-parameter transforms [13]. In particular, the *Angular Radial Transform* (*ART*) is a moment-based image descriptor adopted in MPEG-7 [9]; more specifically, ART is defined by a complex orthogonal unitary transform on the unit disk.

In a recent image retrieval application [1], we have represented a 2D shape by a *shape descriptors* vector $\overrightarrow{x_d} = (x_1, \ldots, x_{N_d})$, where $d \in \{FD, ART, IM\}$. We remark that the entries of vector $\overrightarrow{x_d}$ are indexed according to inherent semantics of the employed descriptor extraction algorithm. A significant practical advantage of any aforementioned *shape descriptors* vector is the translation /scale /rotation invariance of the corresponding 2D shape [52]. Furthermore, note that computa-

tional experiments have demonstrated that the larger the (integer) number N_d, the more accurate is the corresponding 2D shape reconstruction [31].

This paper is a significant extension of preliminary work in [27] including the following novelties. First, it presents a wider hierarchy, including Type-II intervals/INs, of complete lattices stemming from $(\mathsf{R}, \leq)$. Second, it details a novel 2D shape representation technique. Third, it introduces a FLR scheme to the product lattice $(\mathsf{F}^N, \preceq)$. Fourth, it presents, comparatively, only new experimental application results. Fifth, it cites a large number of new references including novel perspectives.

The work here is organized as follows. Section 2 details a wide hierarchy of complete lattices. Section 3 summarizes a fuzzy lattice reasoning (FLR) scheme for learning/generalization. Section 4 details a novel 2D shape representation. Section 5 presents, comparatively, experimental application results. Section 6 concludes by summarizing the contribution as well as discussing potential future work. The Appendix displays useful mathematical notions.

2 A Hierarchy of Lattices in Perspective

This section introduces constructively, in six steps, a hierarchy of complete lattices; in particular, each subsection presents a progressively enhanced (lattice) hierarchy level. Various interpretations are also presented in this section. Useful notation and tools regarding general lattice theory are summarized in the Appendix.

2.1 Real Numbers

The set R of real numbers is a totally-ordered, non-complete lattice denoted by $(\mathsf{R}, \leq)$. The latter (lattice) can be extended to a complete lattice by including both symbols "$-\infty$" and "$+\infty$". In conclusion, the complete lattice $(\overline{\mathsf{R}}, \leq)$ emerges, where $\overline{\mathsf{R}} = \mathsf{R} \cup \{-\infty, +\infty\}$.

On the one hand, any strictly increasing function $v : \overline{\mathsf{R}} \to \overline{\mathsf{R}}$ is a positive valuation in the complete lattice $(\overline{\mathsf{R}}, \leq)$. Motivated by the two constraints presented underneath Theorem 1, here we consider positive valuation functions $v : \overline{\mathsf{R}} \to \overline{\mathsf{R}}$ such that both $v(-\infty) = 0$ and $v(+\infty) = A < +\infty$. That is, here we consider "saturated" positive valuation functions $v : \overline{\mathsf{R}} \to [0, A]$, where $0 < A < +\infty$. On the other hand, any one-to-one, strictly decreasing function $\theta : \overline{\mathsf{R}} \to \overline{\mathsf{R}}$ is an eligible dual isomorphic function in lattice $(\overline{\mathsf{R}}, \leq)$. We will refer to functions $\theta(.)$ and $v(.)$ as *dual isomorphic* and *positive valuation*, respectively.

2.2 Type-I Intervals

Type-I (including conventional) intervals $[a, b]$ of real numbers are studied in this section. A more general interval type is defined, in the first place, next.

Definition 1 *Generalized Type-I interval* is an element of the product lattice $(\overline{\mathsf{R}}, \leq^{\partial}) \times (\overline{\mathsf{R}}, \leq)$.

Recall that $\leq^{\partial}$ in Definition 1 denotes the *dual* (i.e. converse) of order relation $\leq$ in lattice $(\overline{\mathsf{R}}, \leq)$, i.e. $\leq^{\partial} \equiv \geq$. Product lattice $(\overline{\mathsf{R}}, \leq^{\partial}) \times (\overline{\mathsf{R}}, \leq) \equiv (\overline{\mathsf{R}} \times \overline{\mathsf{R}}, \geq \times \leq)$ will be denoted, simply, by $(\mathit{\Delta}, \preceq)$.

A generalized Type-I interval will be denoted by $[x, y]$, where $x, y \in \overline{\mathsf{R}}$. It follows that the *meet* ($\curlywedge$) and *join* ($\curlyvee$) in lattice $(\mathit{\Delta}, \preceq)$ are given, respectively, by $[a, b] \curlywedge [c, d] = [a \vee c, b \wedge d]$ and $[a, b] \curlyvee [c, d] = [a \wedge c, b \vee d]$.

The set of *positive* (*negative*) generalized Type-I intervals $[a, b]$, characterized by $a \leq b$ ($a > b$), will be denoted by $\mathit{\Delta}_+$ ($\mathit{\Delta}_-$). It turns out that $(\mathit{\Delta}_+, \preceq)$ is a poset, which is *isomorphic* to the poset $(\tau(\overline{\mathsf{R}}), \preceq)$ of conventional intervals (sets) in $\overline{\mathsf{R}}$, i.e. $(\tau(\overline{\mathsf{R}}), \preceq) \cong (\mathit{\Delta}_+, \preceq)$. We augmented poset $(\tau(\overline{\mathsf{R}}), \preceq)$ by a unique *least* (empty) interval denoted by $O = [+\infty, -\infty]$ – We remark that a unique *greatest* interval $I = [-\infty, +\infty]$ already exists in $\tau(\overline{\mathsf{R}})$. Hence, the complete lattice $(\tau_O(\overline{\mathsf{R}}) = \tau(\overline{\mathsf{R}}) \cup \{O\}, \preceq) \cong (\mathit{\Delta}_+ \cup \{O\}, \preceq)$ of *Type-I intervals* emerged. In the sequel, we employ isomorphic lattices $(\mathit{\Delta}_+ \cup \{O\}, \preceq)$ and $(\tau_O(\overline{\mathsf{R}}), \preceq)$, interchangeably.

Consider a positive valuation function $v : \overline{\mathsf{R}} \to [0, A]$ as well as a dual isomorphic function $\theta : \overline{\mathsf{R}} \to \overline{\mathsf{R}}$. Then, based on Proposition 1, it follows that function $v_{\mathit{\Delta}} : \mathit{\Delta} \to [0, 2A]$ given by $v_{\mathit{\Delta}}([a, b]) = v(\theta(a)) + v(b)$ is a positive valuation in lattice $(\mathit{\Delta}, \preceq)$. Furthermore, based on both $v(-\infty) = 0$ and $v(+\infty) = A < +\infty$, there follow $v_{\mathit{\Delta}}(O = [+\infty, -\infty]) = 0$ as well as $v_{\mathit{\Delta}}(I = [-\infty, +\infty]) = 2A < +\infty$. In conclusion, the following two inclusion measures emerge in lattice $(\mathit{\Delta}, \preceq)$.

$\sigma_{\curlywedge}([a, b] \preceq [c, d]) = \frac{v([a \vee c, b \wedge d])}{v([a, b])} = \frac{v(\theta(a \vee c)) + v(b \wedge d)}{v(\theta(a)) + v(b)}$.

$\sigma_{\curlyvee}([a, b] \preceq [c, d]) = \frac{v([c, d])}{v([a \wedge c, b \vee d])} = \frac{v(\theta(c)) + v(d)}{v(\theta(a \wedge c)) + v(b \vee d)}$.

The aforementioned inclusion measures are, in particular, applicable in the complete lattice $(\tau_O(\overline{\mathsf{R}}), \preceq)$ of Type-I intervals as follows.

$$\sigma_{\curlywedge}([a, b] \preceq [c, d]) = \begin{cases} \frac{v(\theta(a \vee c)) + v(b \wedge d)}{v(\theta(a)) + v(b)}, & a \vee c \leq b \wedge d \\ 0, & a \vee c > b \wedge d \end{cases} \tag{1}$$

$$\sigma_{\curlyvee}([a, b] \preceq [c, d]) = \frac{v(\theta(c)) + v(d)}{v(\theta(a \wedge c)) + v(b \vee d)}. \tag{2}$$

In this work we employ, exclusively, inclusion measure $\sigma_{\curlyvee}(.,.)$ rather than $\sigma_{\curlywedge}(.,.)$ because only $\sigma_{\curlyvee}(.,.)$ is non-zero for non-overlapping Type-I intervals.

Therefore, only $\sigma_{\curlyvee}(.,.)$ is dependable for sensible decision-making in practical applications involving non-overlapping (Type-I) intervals.

Functions $\theta(.)$ and $v(.)$ can be selected in different ways. For instance, choosing $\theta(x) = -x$ and $v(.)$ such that $v(x) = -v(-x)$ it follows $v_{\Delta}([a,b]) = v(b) - v(a)$. In the context of this work, we select a pair of functions $v(x)$ and $\theta(x)$ so as to satisfy equality $v_{\Delta}([x,x]) = v(\theta(x)) + v(x) = Constant$ required for atoms by a standard FLR scheme [22,26,28]. For instance, eligible 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 $A, \lambda \in \mathsf{R}^{\geq 0}$, $\mu, x \in \mathsf{R}$.

A sigmoid positive valuation is preferable because a sigmoid is defined over the whole set R of real numbers. Moreover, empirical evidence strongly suggests that a sigmoid positive valuation can produce identical results as a linear positive valuation [22]. An additional advantage for a sigmoid positive valuation function is its capacity to deal with intervals of Type-I INs as explained below. Therefore, here we employ a parametric sigmoid positive valuation $v_s(x; A, \lambda, \mu)$ together with a parametric dual isomorphic function $\theta(x; \mu) = 2\mu - x$, exclusively; it follows $v_{\Delta}([x,x]) = A$ for an atom $[x,x]$.

2.3 Type-II Intervals

A Type-II interval is "an interval of (Type-I) intervals" as explained in this section. A more general interval type is defined, in the first place, next.

Definition 2 *Generalized Type-II interval* is an element of the product lattice $(\Delta, \preceq^{\partial}) \times (\Delta, \preceq) \equiv (\Delta \times \Delta, \succeq \times \preceq)$.

We remark that a generalized Type-II interval will be denoted by $[x,y]$, where $x, y \in (\Delta, \preceq)$. It follows

$[[a_1,a_2],[b_1,b_2]] \curlywedge [[c_1,c_2],[d_1,d_2]] = [[a_1,a_2] \curlyvee [c_1,c_2], [b_1,b_2] \curlywedge [d_1,d_2]]$, and

$[[a_1,a_2],[b_1,b_2]] \curlyvee [[c_1,c_2],[d_1,d_2]] = [[a_1,a_2] \curlywedge [c_1,c_2], [b_1,b_2] \curlyvee [d_1,d_2]]$.

Our interest here focuses on the complete lattice $(\tau_O(\tau_O(\overline{\mathsf{R}})), \preceq)$ of *Type-II intervals*. For the reader's interest, we point out that the least element O of the complete lattice $(\tau_O(\tau_O(\overline{\mathsf{R}})), \preceq)$ equals $O = [[-\infty, +\infty], [+\infty, -\infty]]$.

Recall that above we have assumed, first, a (strictly increasing) "saturated" *positive valuation* function $v : \overline{\mathsf{R}} \to [0, A]$ such that both $v(-\infty) = 0$ and $v(+\infty) = A < +\infty$ and, second, a (strictly decreasing) *dual isomorphic* function $\theta : \overline{\mathsf{R}} \to \overline{\mathsf{R}}$. Furthermore, we assumed a positive valuation function $v_{\Delta} : \Delta \to [0, 2A]$ given by $v_{\Delta}([a,b]) = v(\theta(a)) + v(b)$. Note that a dual isomorphic function $\theta_{\Delta} : \Delta \to \Delta$ is given by $\theta_{\Delta}([a,b]) = [b,a]$ as explained in the following: Inequality "$[a,b] \preceq [c,d]$" is equivalent to the disjunction "either $(c < a).AND.(b \leq d)$ or $(c \leq a).AND.(b <$

$d)$"; the latter is equivalent to the disjunction "either $(b < d).AND.(c \leq a)$ or $(b \leq d).AND.(c < a)$"; the latter is equivalent to inequality "$[d, c] \preceq [b, a]$"; the latter is equivalent to inequality "$\theta_{\Delta}([a, b]) \succeq \theta_{\Delta}([c, d])$".

Based on Proposition 1, it follows that function $v_{\Delta\Delta} : \Delta \times \Delta \to [0, 4A]$ given by $v_{\Delta\Delta}([[a_1, a_2], [b_1, b_2]]) = v(a_1) + v(\theta(a_2)) + v(\theta(b_1)) + v(b_2)$ is a positive valuation in lattice $(\Delta \times \Delta, \preceq)$. Furthermore, based on positive valuation function $v_s(x) = \frac{A}{1+e^{-\lambda(x-\mu)}}$ as well as dual isomorphic function $\theta(x) = 2\mu - x$, there follows $v_{\Delta\Delta}([[a_1, a_2], [a_1, a_2]]) = v(a_1) + v(\theta(a_2)) + v(\theta(a_1)) + v(a_2) = 2A$ for an *atom* Type-II interval $[[a_1, a_2], [a_1, a_2]]$. There follow both $v_{\Delta\Delta}(O) = 0$ and $v_{\Delta\Delta}(I) = 4v(+\infty) = 4A < +\infty$.

According to Theorem 1, two inclusion measures emerge in the complete lattice $(\Delta \times \Delta, \preceq)$ of generalized Type-II intervals as follows.

First, $\sigma_{\curlywedge}([[a_1, a_2], [b_1, b_2]] \preceq [[c_1, c_2], [d_1, d_2]]) =$
$= \frac{v_{\Delta\Delta}([[a_1 \wedge c_1, a_2 \vee c_2], [b_1 \vee d_1, b_2 \wedge d_2]])}{v_{\Delta\Delta}([[a_1, a_2], [b_1, b_2]])} =$
$= \frac{v(a_1 \wedge c_1) + v(\theta(a_2 \vee c_2)) + v(\theta(b_1 \vee d_1)) + v(b_2 \wedge d_2)}{v(a_1) + v(\theta(a_2)) + v(\theta(b_1)) + v(b_2)}.$

Second, $\sigma_{\curlyvee}([[a_1, a_2], [b_1, b_2]] \preceq [[c_1, c_2], [d_1, d_2]]) =$
$= \frac{v_{\Delta\Delta}([[c_1, c_2], [d_1, d_2]])}{v_{\Delta\Delta}([[a_1 \vee c_1, a_2 \wedge c_2], [b_1 \wedge d_1, b_2 \vee d_2]])} =$
$= \frac{v(c_1) + v(\theta(c_2)) + v(\theta(d_1)) + v(d_2)}{v(a_1 \vee c_1) + v(\theta(a_2 \wedge c_2)) + v(\theta(b_1 \wedge d_1)) + v(b_2 \vee d_2)}.$

The aforementioned inclusion measures are, in particular, applicable in the complete lattice $(\tau_O(\tau_O(\overline{\mathsf{R}})), \preceq)$ of Type-II intervals as follows.

$$\sigma_{\curlywedge}([[a_1, a_2], [b_1, b_2]] \preceq [[c_1, c_2], [d_1, d_2]]) = \begin{cases} \frac{v(a_1 \wedge c_1) + v(\theta(a_2 \vee c_2)) + v(\theta(b_1 \vee d_1)) + v(b_2 \wedge d_2)}{v(a_1) + v(\theta(a_2)) + v(\theta(b_1)) + v(b_2)}, & b_1 \vee d_1 \leq a_1 \wedge c_1 \leq a_2 \vee c_2 \leq b_2 \wedge d_2 \\ 0, & otherwise \end{cases} \tag{3}$$

$$\sigma_{\curlyvee}([[a_1, a_2], [b_1, b_2]] \preceq [[c_1, c_2], [d_1, d_2]]) = \begin{cases} \frac{v(c_1) + v(\theta(c_2)) + v(\theta(d_1)) + v(d_2)}{v(a_1 \vee c_1) + v(\theta(a_2 \wedge c_2)) + v(\theta(b_1 \wedge d_1)) + v(b_2 \vee d_2)}, & a_1 \vee c_1 \leq a_2 \wedge c_2 \\ \frac{v(c_1) + v(\theta(c_2)) + v(\theta(d_1)) + v(d_2)}{2A + v(\theta(b_1 \wedge d_1)) + v(b_2 \vee d_2)}, & a_1 \vee c_1 > a_2 \wedge c_2 \end{cases} \tag{4}$$

The *size* of a Type-II interval $[[a_1, a_2], [b_1, b_2]]$ in the poset $(\tau(\tau_O(\overline{\mathsf{R}})), \preceq)$ equals

$$Z_{\Delta\Delta}([[a_1, a_2], [b_1, b_2]]) = v(\theta(b_1)) + v(b_2) - v(\theta(a_1)) - v(a_2) \tag{5}$$

We remark that $Z_{\Delta\Delta}([O, [b_1, b_2]]) = v(\theta(b_1)) + v(b_2) = v_{\Delta}([b_1, b_2])$.

2.4 Type-I INs

Based on the analysis above, this subsection presents *Type-I intervals' numbers* (Type-I INs). A more general number type is defined, in the first place, next.

Definition 3 *Type-I generalized intervals's number*, or *Type-I GIN* for short, is a function $G : (0, 1] \to \Delta$.

Let G denote the set of Type-I GINs. It follows complete lattice $(\mathsf{G}, \preceq)$, as the product of complete lattices $(\Delta, \preceq)$. Our interest here focuses on the *sublattice*[1] of *Type-I intervals' numbers* defined next.

Definition 4 A *Type-I Intervals' Number*, or *Type-I IN* for short, is a Type-I GIN F such that both $F(h) \in (\Delta_+ \cup \{O\})$ and $h_1 \leq h_2 \Rightarrow F(h_1) \succeq F(h_2)$.

Let F denote the set of Type-I INs. It turns out, as shown in [36], that $(\mathsf{F}, \preceq)$ is a complete lattice with least element $O = O(h) = [+\infty, -\infty]$, $h \in (0, 1]$ and greatest element $I = I(h) = [-\infty, +\infty]$, $h \in (0, 1]$. A Type-I IN will be denoted by a capital letter in italics, e.g. $F \in \mathsf{F}$.

Definition 4 implies that a Type-I IN F is a function from interval $(0, 1]$ to the set $\tau_O(\overline{\mathsf{R}})$ of Type-I intervals. Hence, a Type-I IN will also be denoted by $F(h) = [a_h, b_h]$, $h \in (0, 1]$, where both interval-ends a_h and b_h are functions of $h \in (0, 1]$.

Based on equations (1) and (2), respectively, the following two inclusion measures emerge in the complete lattice $(\mathsf{F},\preceq)$ of Type-I INs [19]:

$$\sigma_\curlywedge(E_1 \preceq E_2) = \int_0^1 \sigma_\curlywedge(E_1(h) \preceq E_2(h))dh. \tag{6}$$

$$\sigma_\curlyvee(E_1 \preceq E_2) = \int_0^1 \sigma_\curlyvee(E_1(h) \preceq E_2(h))dh. \tag{7}$$

The complete lattice $(\mathsf{F}, \preceq)$ of Type-I INs has been studied in a series of publications [21,22,25,36]. In short, it has been shown that a Type-I IN is a mathematical object, which may be interpreted as either a possibility- or a probability-distribution function. Moreover, the cardinality of the set F equals the cardinality $\aleph_1$ of the set R of real numbers; that is, there is a one-to-one correspondence between Type-I INs and real numbers. Nevertheless, the set R of real numbers is *totally*(*lattice*)-*ordered*, whereas the set F of Type-I INs is *partially*(*lattice*)-*ordered*.

In practice, a Type-I IN can be interpreted as an information *granule*.

[1] A *sublattice* of a lattice $(\mathsf{L}, \preceq)$ is another lattice $(\mathsf{S}, \preceq)$ such that $\mathsf{S} \subseteq \mathsf{L}$.

2.5 Type-II INs

Another information *granule* of interest here is an interval $[U, W]$ of Type-I INs, i.e. the set $\{X : X \in \mathsf{F} \text{ and } U \preceq X \preceq W\}$. A latter interval will be called Type-II IN. It follows the complete lattice $(\tau_O(\mathsf{F}), \preceq)$ of Type-II INs. We remark that the least (empty) interval O in $\tau_O(\mathsf{F})$ is $O = O(h) = [+\infty, -\infty]$, for $h \in [0, 1]$. A Type-II IN will be denoted by a double-lined capital letter, e.g. $\mathbb{E} \in \tau_O(\mathsf{F})$.

Based on equations (3) and (4), respectively, two inclusion measures emerge in lattice $(\tau_O(\mathsf{F}), \preceq)$ as follows.

$$\sigma_{\curlywedge}(\mathbb{E}_1 \preceq \mathbb{E}_2) = \int_0^1 \sigma_{\curlywedge}(\mathbb{E}_1(h) \preceq \mathbb{E}_2(h))dh. \tag{8}$$

$$\sigma_{\curlyvee}(\mathbb{E}_1 \preceq \mathbb{E}_2) = \int_0^1 \sigma_{\curlyvee}(\mathbb{E}_1(h) \preceq \mathbb{E}_2(h))dh. \tag{9}$$

The *size* of a Type-II IN $\mathbb{E} = [A, B]$ is computed as follows.

$$Z(\mathbb{E}) = Z([A, B]) = \int_0^1 Z_{\Delta\Delta}([A(h), B(h)])dh. \tag{10}$$

We remark that previous attempts [24,25] to deal with Type-II INs have failed mainly due to the employment of "non-saturated" (strictly increasing) positive valuation functions $v : \overline{\mathsf{R}} \to \overline{\mathsf{R}}$ as well as due to the employment of the poset $(\tau(\overline{\mathsf{R}}), \preceq)$. Whereas, Type-II INs are dealt with here in the complete lattice $(\tau_O(\overline{\mathsf{R}}), \preceq)$ based on "saturated" (sigmoid) positive valuations $v : \overline{\mathsf{R}} \to [0, A]$. Further details will be presented elsewhere because they are outside the scope of this paper.

2.6 Extensions to More Dimensions

A N-tuple IN of either Type-I or Type-II will be indicated by an "over right arrow". More specifically, the former will be denoted by $\overrightarrow{E} = (E_1, \ldots, E_N) \in (\mathsf{F}^N, \preceq)$, whereas the latter will be denoted by $\overrightarrow{\mathbb{E}} = (\mathbb{E}_1, \ldots, \mathbb{E}_N) \in ((\tau_O(\mathsf{F}))^N, \preceq)$. Note that N-tuples $\overrightarrow{E}$ of Type-I INs have already been shown to be the vectors of a *cone* in a linear space [19,23,36].

Lattice $((\tau_O(\mathsf{F}))^N, \preceq)$ is the "sixth level" in a hierarchy of complete lattices whose previous levels include, first, lattice $(\overline{\mathsf{R}}, \leq)$, second, lattice $(\tau_O(\overline{\mathsf{R}}), \preceq)$, third, lattice $(\tau_O(\tau_O(\overline{\mathsf{R}})), \preceq)$, fourth, lattice $(\mathsf{F}, \preceq)$ and, fifth, lattice $(\tau_O(\mathsf{F}), \preceq)$.

We remark that function $\psi : (\tau_O(\overline{\mathsf{R}}), \preceq) \to (\{O\} \times \tau_O(\overline{\mathsf{R}}), \preceq)$ given by $\psi([b_1, b_2]) = [O = [+\infty, -\infty], [b_1, b_2]]$ is *isomorphic*. In other words, lattices $(\tau_O(\overline{\mathsf{R}}), \preceq)$ and

$(\{O\} \times \tau_O(\overline{\mathsf{R}}), \preceq)$ are *isomorphic*; symbolically, $(\tau_O(\overline{\mathsf{R}}), \preceq) \cong (\{O\} \times \tau_O(\overline{\mathsf{R}}), \preceq)$. It follows $(\mathsf{F}, \preceq) \cong (\{O\} \times \mathsf{F}, \preceq)$.

The analysis above has shown how to define inclusion measure functions in lattice $(\tau_O(\mathsf{F}), \preceq)$. The aforementioned functions can be extended to the product lattice $((\tau_O(\mathsf{F}))^N, \preceq)$, based on Proposition A.11 in [25]. For the reader's convenience, the latter proposition's statement is repeated in the following: "Let function $\sigma_i : \mathsf{L}_i \times \mathsf{L}_i \to [0,1]$ be an inclusion measure in lattice $(\mathsf{L}_i, \preceq)$, $i = 1, \ldots, N$. Then, the *convex combination* $\sigma((u_1, \ldots, u_N) \preceq (w_1, \ldots, w_N)) = \lambda_1\sigma_1(u_1 \preceq w_1) + \cdots + \lambda_N\sigma_N(u_N \preceq w_N)$ is an inclusion measure in the product lattice $(\mathsf{L}, \preceq) = (\mathsf{L}_1, \preceq) \times \cdots \times (\mathsf{L}_N, \preceq) = (\mathsf{L}_1 \times \cdots \times \mathsf{L}_N, \preceq)$", where by "convex combination" we mean a set $\lambda_1, \ldots, \lambda_I$ of non-negative numbers such that $\lambda_1 + \cdots + \lambda_I = 1$.

The *size* of a Type-II N-tuple IN $\overrightarrow{\mathbb{E}} = (\mathbb{E}_1, \ldots, \mathbb{E}_N)$ is computed as follows.

$$Z(\overrightarrow{\mathbb{E}} = (\mathbb{E}_1, \ldots, \mathbb{E}_N)) = Z(\mathbb{E}_1) + \cdots + Z(\mathbb{E}_N). \tag{11}$$

3 A Type-II Fuzzy Lattice Reasoning (FLR) Scheme

Fuzzy Lattice Reasoning (FLR) was introduced for computing hyperboxes in lattice $(\mathsf{R}^N, \preceq)$ [19,28]. Later, a FLR extension was introduced in lattice $(\{O\} \times \mathsf{F}^N, \preceq)$ using a different mathematical notation [25]. In all cases, the instrument for fuzzy lattice reasoning (FLR) is an inclusion measure function $\sigma(.,.)$ in a complete lattice $(\mathsf{L}, \preceq)$, which (σ) supports at least two different modes of reasoning, namely *Generalized Modus Ponens* and *Reasoning by Analogy*. More specifically, on the one hand, *Generalized Modus Ponens* is supported as follows: Given both an implication (rule) "IF *variable* V_0 is E THEN *proposition* p" and a proposition "*variable* V_0 is E_p", where both E_p and E are lattice $(\mathsf{L}, \preceq)$ elements such that $E_p \preceq E$, it follows "*proposition* p". On the other hand, *Reasoning by Analogy* is supported as follows: Given both a set of implications (rules) "IF *variable* V_0 is E_k THEN *proposition* p_k", $k \in \{1, \ldots, K\}$ and a proposition "*variable* V_0 is E_p", where both E_p and E_k for $k \in \{1, \ldots, K\}$ are lattice $(\mathsf{L}, \preceq)$ elements such that such that $E_p \not\preceq E_k$, for $k \in \{1, \ldots, K\}$, it follows "*proposition* p_J", where $J \doteq arg \underset{k \in \{1,\ldots,K\}}{\mathtt{max}} \{\sigma(E_p \preceq E_k) < 1\}$. A novel FLR extension in the lattice $((\tau_O(\mathsf{F}))^N, \preceq)$ of N-tuples of Type-II INs is introduced in the following.

3.1 FLRtypeII Scheme for Learning (Training)

Algorithm 1 introduces the FLRtypeII scheme for learning (training) in the product lattice $(\mathsf{F}^N, \preceq)$ by computing intervals; the latter (intervals) are elements of

lattice $((\tau_O(\mathsf{F}))^N, \preceq)$. Note that the training phase employs a series of data pairs $(\overrightarrow{\mathbb{E}}_i, \ell(\overrightarrow{\mathbb{E}}_i))$, $i = 1, \ldots, n$, where $\overrightarrow{\mathbb{E}}_i \in (\tau_O(\mathsf{F}))^N$, moreover $\ell(\overrightarrow{\mathbb{E}}_i)$ obtains values in the set $\{1, \ldots, L\}$ of class labels.

Algorithm 1 : FLRtypeII scheme for learning (training)

1: Let Z_0 be a user-defined threshold size.
2: The first training datum $(\overrightarrow{\mathbb{E}}_1, \ell(\overrightarrow{\mathbb{E}}_1))$ is memorized; $M(1) = 1$.
3: **for** $i = 2$ to $i = n$ **do**
4: Present the next training datum $(\overrightarrow{\mathbb{E}}_i, \ell(\overrightarrow{\mathbb{E}}_i))$ to the "set" classes $c_1, \ldots c_{M(i-1)}$.
5: **while** there exit "set" classes **do**
6: For each "set" class c_j calculate $\sigma_\Upsilon(\overrightarrow{\mathbb{E}}_i \preceq c_j)$.
7: Competition among the "set" classes c_j, where $j \in \{1, \ldots, M(i-1)\}$.
Winner is class c_J: $J = \underset{j\in\{1,\ldots,M(i-1)\}}{\mathtt{argmax}} \{\sigma_\Upsilon(\overrightarrow{\mathbb{E}}_i \preceq c_j)\}$.
Let $\overrightarrow{\mathbb{W}}_L$ be the interval of $c_J = \bigcup_k \{\overrightarrow{\mathbb{W}}_{J,k}\}$: $L = \underset{k}{\mathtt{argmax}}\{\sigma_\Upsilon(\overrightarrow{\mathbb{E}}_i \preceq \overrightarrow{\mathbb{W}}_{J,k})\}$
– Break (possible) ties by selecting the interval $\overrightarrow{\mathbb{W}}_L$ with the smallest size.
8: *Assimilation Condition*: Both $Z(\overrightarrow{\mathbb{E}}_i \curlyvee \overrightarrow{\mathbb{W}}_L) < Z_0$ and $\ell(\overrightarrow{\mathbb{E}}_i) = \ell(\overrightarrow{\mathbb{W}}_L)$.
9: If the *Assimilation Condition* is satisfied then replace $\overrightarrow{\mathbb{W}}_L$ by $\overrightarrow{\mathbb{E}}_i \curlyvee \overrightarrow{\mathbb{W}}_L$; otherwise, "reset" class c_J.
10: **end while**
11: If all the classes $c_1, \ldots c_{M(i-1)}$ have been "reset" then memorize the training datum $(\overrightarrow{\mathbb{E}}_i, \ell(\overrightarrow{\mathbb{E}}_i))$; moreover, $M(i) = M(i-1) + 1$.
12: **end for**

We remark that integer function $M(i-1)$, with (integer) argument "$i-1$", denotes the number of classes at (discrete) time "i". Moreover, a clas $c_j = \bigcup_k \{\overrightarrow{\mathbb{W}}_{j,k}\}$ is the (finite) set of intervals $\overrightarrow{\mathbb{W}}_{j,k} \in (\tau_O(\mathsf{F}))^N$ characterized by the same label $\ell(\overrightarrow{\mathbb{W}}_{j,k})$; the latter (label) denotes the label $\ell(c_j)$ of class c_j.

It is important to point out that function $\sigma_\Upsilon(\overrightarrow{\mathbb{E}}_i \preceq c_j)$, for class $c_j = \bigcup_k \{\overrightarrow{\mathbb{W}}_{j,k}\}$, is a well-defined inclusion measure computable based on Proposition 3.4 in [29]. For the reader's convenience, the latter proposition's statement is repeated next: "Let function $\sigma_V : \mathsf{L} \times \mathsf{L} \to [0,1]$ be an inclusion measure in a lattice $(\mathsf{L}, \preceq)$. Then function $\sigma : 2^{\mathsf{L}} \times 2^{\mathsf{L}} \to [0,1]$ given by the *convex combination* $\sigma(\{u_1, \ldots, u_I\} \preceq \{w_1, \ldots, w_J\}) = \lambda_1 \underset{j}{\mathtt{max}} \sigma_V(u_1 \preceq w_j) + \cdots + \lambda_I \underset{j}{\mathtt{max}} \sigma_V(u_I \preceq w_j)$ is an inclusion measure", where by "convex combination" we mean a set $\lambda_1, \ldots, \lambda_I$ of non-negative numbers such that $\lambda_1 + \cdots + \lambda_I = 1$ – We remark that 2^{L} denotes the power set of (set) L. Here we assume $\lambda_1 = 1$; moreover, $\lambda_2 = \cdots = \lambda_I = 0$. It follows inclusion measure $\sigma_\Upsilon(\overrightarrow{\mathbb{E}}_i \preceq c_j) = \sigma_\Upsilon(\{\overrightarrow{\mathbb{E}}_i\} \preceq \bigcup_k \{\overrightarrow{\mathbb{W}}_{j,k}\}) = \underset{k}{\mathtt{max}} \sigma_\Upsilon(\overrightarrow{\mathbb{E}}_i \preceq \overrightarrow{\mathbb{W}}_{j,k})$.

3.2 FLRtypeII Scheme for Generalization (Testing)

Algorithm 2 presents the corresponding FLRtypeII scheme for generalization (testing) in the product lattice $((\tau_O(\mathsf{F}))^N, \preceq)$.

Algorithm 2 : FLRtypeII scheme for generalization (testing)

1: Assume a set $\bigcup_{j\in\{1,\dots M\}} \{c_j\}$ of labeled classes c_j, where each class c_j is a (finite) collection of intervals in lattice $((\tau_O(\mathsf{F}))^N, \preceq)$; i.e. $c_j = \bigcup_k \{\overrightarrow{\mathbb{W}}_{j,k}\} \subseteq 2^{(\tau_O(\mathsf{F}))^N}$, moreover let $\ell(c_j) \in \{1, \dots, L\}$ denote the label of class c_j.
2: Consider an unlabeled datum $\overrightarrow{\mathbb{E}}_0 \in (\tau_O(\mathsf{F}))^N$ for testing (classification).
3: For each class c_j calculate inclusion measure $\sigma_\Upsilon(\overrightarrow{\mathbb{E}}_0 \preceq c_j)$.
4: Competition among classes c_j, where $j \in \{1, \dots, M\}$.
Winner is class c_J: $J = \underset{j\in\{1,\dots,M\}}{\texttt{argmax}} \{\sigma_\Upsilon(\overrightarrow{\mathbb{E}}_0 \preceq c_j)\}$.
5: The class label $\ell(\overrightarrow{\mathbb{E}}_0)$ of datum $\overrightarrow{\mathbb{E}}_0$ is defined to be $\ell(\overrightarrow{\mathbb{E}}_0) = \ell(c_J)$.

We remark that an abovementioned class c_j, $j \in \{1, \dots M\}$ consists of a number of lattice $((\tau_O(\mathsf{F}))^N, \preceq)$ intervals, where an interval is interpreted here as an *(information) granule* [38,50]. Likewise, a training/testing datum $\overrightarrow{\mathbb{E}} \in (\tau_O(\mathsf{F}))^N$ is a granule, including trivial granules $\overrightarrow{\mathbb{E}} = [E, E]$ as a special case.

4 Data Preprocessing and 2D Shape Representation

This section, first, demonstrates an induction of Type-II INs from vectors of numbers, second, it details a novel 2D shape representation based on Type-II INs.

4.1 Inducing Type-I INs

Consider a vector $\overrightarrow{x} = (x_1, \dots, x_n)$ with real number entries. Two entries x_i, x_j are called "successive" if and only if there no other entry x_k such that $x_i \wedge x_j < x_k < x_i \vee x_j$.

A strictly-increasing *cumulative* (real) function $c : \mathsf{R} \to \mathsf{R}$ is induced from the entries of vector $\overrightarrow{x}$, first, by defining $c(x_i) = \frac{1}{n} \sum_{x_j \le x_i} \left(\int_{-\infty}^{+\infty} \delta(t - x_j) dt \right)$, where $\delta(t)$ is the Dirac delta function, and, second, by straight-line connecting two points $(x_i, c(x_i))$ and $(x_j, c(x_j))$, where x_i, x_j are "successive" entries of vector $\overrightarrow{x}$ – We remark that cumulative function $c(.)$ obtains values from 0 to 1 included.

Fig.1(b) shows a (strictly increasing) cumulative function c induced from the real numbers in Fig.1(a). Note that both Fig.1 (a) and (b) indicate the median value 1.484 of the corresponding data. In particular, Fig.1(b) indicates that $c(1.484) = 0.5$, as expected. Fig.1(c) displays a Type-I IN E_1 induced from function $c(.)$ such that for values less than the median 1.484, the envelope function of E_1 equals $2c$; whereas, for values larger than the median 1.484, the envelope function of E_1 equals $2(1 - c)$.

Recall that there is a one-to-one correspondence between Type-I INs and cumulative functions [19,23,25]. In other words, a Type-I IN is an alternative representation for a cumulative function.

4.2 Inducing Type-II INs

Assume a number of Type-I INs $E_1, E_2, \ldots$ each one induced as described above. Type-I INs $E_1, E_2, \ldots$ can, equivalently, be represented by *trivial* Type-II INs, respectively, as follows $\mathbb{E}_1 = [E_1, E_1], \mathbb{E}_2 = [E_2, E_2], \ldots$. For instance, three trivial Type-II INs $\mathbb{E}_1 = [E_1, E_1]$, $\mathbb{E}_2 = [E_2, E_2]$, $\mathbb{E}_3 = [E_3, E_3]$, i.e. elements of the complete lattice $(\tau_O(\mathsf{F}), \preceq)$, are shown in Fig.2(a). The latter (Type-II) INs are employed, next, for demonstrating the lattice *join* operation in $(\tau_O(\mathsf{F}), \preceq)$.

On the one hand, Fig.2(b) demonstrates Type-II IN $\mathbb{E}_1 \curlyvee \mathbb{E}_2 = [E_1 \curlywedge E_2, E_1 \curlyvee E_2]$. Note that $E_1 \curlywedge E_2 \neq O = O(h) = [+\infty, -\infty]$, for $h \in [0, 1]$, because Type-I INs E_1 and E_2 overlap. More specifically, since the tip of Type-I IN $E_1 \curlywedge E_2$ is $h_0 = 0.6129$, for $h \in [0, 0.6129]$ it is $(E_1 \curlywedge E_2)(h) \neq O(h) = [+\infty, -\infty]$, whereas for $h \in (0.6129, 1]$ it is $(E_1 \curlywedge E_2)(h) = O(h) = [+\infty, -\infty]$. We note that a Type-I IN X is in the interval $[E_1 \curlywedge E_2, E_1 \curlyvee E_2]$ if and only if $E_1 \curlywedge E_2 \preceq X \preceq E_1 \curlyvee E_2$. In the latter case we say that X is *encoded* in $[E_1 \curlywedge E_2, E_1 \curlyvee E_2]$ – Apparently, both E_1 and E_2 are also *encoded* in $[E_1 \curlywedge E_2, E_1 \curlyvee E_2]$. On the other hand, Fig.2(c) demonstrates Type-II IN $\mathbb{E}_2 \curlyvee \mathbb{E}_3 = [E_2 \curlywedge E_3, E_2 \curlyvee E_2]$, where $E_2 \curlywedge E_3 = O = O(h) = [+\infty, -\infty]$ for $h \in [0, 1]$ because Type-I INs E_2 and E_3 do not overlap – Apparently, both E_2 and E_3 are *encoded* in $[O, E_2 \curlyvee E_3]$.

The illustrations presented above, in this section, are especially useful for interpreting our experimental results/practices below. A novel 2D shape representation, based on Type-II INs, is detailed next.

4.3 A Novel 2D Shape Representation

From each 2D shape on binary images we extracted, as described in [1], three *shape descriptors* vectors $\overrightarrow{x_d} = (x_1, \ldots, x_{N_d})$, where $d \in \{FD, ART, IM\}$. Recall

that *FD* stands for *Fourier descriptors*, *ART* stands for *angular radial transform* (descriptors) and *IM* stands for *image moments* (descriptors).

In this work we considered the three descriptors *FD*, *ART* and *IM* combined, for representing a 2D shape as follows. One data dimension was engaged per descriptor. Then, in each data dimension, a Type-I IN was computed from the cumulative function $c(.)$ induced from the entries of the corresponding vector $\overrightarrow{x_d} = (x_1, \ldots, x_{N_d})$. Hence, a 2D shape was represented by a 3-tuple of Type-I INs or, equivalently, by a 3-tuple of trivial Type-II INs. Moreover, in each of the three dimensions we considered both a parametric sigmoid positive valuation $v_s(x; A_i, \lambda_i, \mu_i) = \frac{A_i}{1+e^{-\lambda_i(x-\mu_i)}}$ and a parametric dual isomorphic function $\theta(x; \mu_i) = 2\mu_i - x$, where $A_i, \lambda_i \in \mathsf{R}^{\geq 0}$, $\mu_i, x \in \mathsf{R}$ and $i \in \{1, 2, 3\}$.

We point out that based on our proposed 2D shape representation, a (represented) 2D shape cannot be recovered. Nevertheless, an advantage of our proposed 2D shape representation is "information compression" because a Type-I IN can accurately represent any number of descriptors, i.e. any number of vector $\overrightarrow{x_d} = (x_1, \ldots, x_{N_d})$ entries. In other words, the distribution of any number of shape descriptors in a data dimension can be represented accurately by a Type-I IN.

Then, the FLRtypeII scheme for learning (training) was applied followed by the FLRtypeII scheme for generalization (testing) on three benchmark data sets as described in the following section.

5 Computational Experiments and Results

In this section we employ three benchmark data sets, namely Kimia-99, Kimia-216 [40] and Chicken-Pieces [2]. Note that precision-and-recall applications of the aforementioned data sets abound in the literature; nevertheless, classification results are scarcely reported.

In our classification experiments below we applied the FLRtypeII scheme; that is, both Algorithm 1 and Algorithm 2 were applied, with $N = 3$. A training/testing datum $\overrightarrow{\mathbb{E}}_i \in (\tau_O(\mathsf{F}))^3$ was a trivial Type-II IN, i.e. $\overrightarrow{\mathbb{E}}_i = [\overrightarrow{E_i}, \overrightarrow{E_i}]$. An inclusion measure ($\sigma$) and a size ($Z$) in the product lattice $((\tau_O(\mathsf{F}))^3, \preceq)$ were defined as in subsection 2.6; in particular, σ was calculated here using $\lambda_1 = \lambda_2 = \lambda_3 = 1/3$. We used *shape descriptors* vectors $\overrightarrow{x_d} = (x_1, \ldots, x_{N_d})$, where $d \in \{FD, ART, IM\}$, with $N_{FD} = 32$, $N_{ART} = 35$ and $N_{IM} = 6$. All descriptor values were normalized. Furthermore, a Type-I IN was represented with $L = 32$ intervals equally spaced from $h = 0$ to $h = 1$ included.

Parameter optimization was pursued by a genetic algorithm, where the phenotype of an "individual" consisted of specific values for three sigmoid function $v_s(x; A_i, \lambda_i, \mu_i)$ parameters A_i, λ_i and μ_i per data dimension $i \in \{1, 2, 3\}$; an addi-

tional parameter was the threshold size Z_0. In conclusion, there was a total number of $3 \times 3 + 1 = 10$ parameters, binary-encoded in the chromosome of an individual. The ranges of the aforementioned parameters were $A_i \in [0, 100]$, $\lambda_i \in [0, 10]$ and $\mu_i \in [-10, 10]$; moreover, based on the previously selected parameters A_1, A_2 and A_3, we assumed $Z_0 \in [0.01, A_1 + A_2 + A_3]$. We included 25 individuals per generation. The genetic algorithm was enhanced by the *microgenetic hill-climbing* operator; in addition, both *elitism* and *adaptive crossover/mutation* rates were implemented [36].

To avoid overtraining, the fitness value of a genetic algorithm individual was computed by partitioning, wherever possible, the data set in three subsets including (1) a training set (2) a validation set, and (3) a testing set. Let E_{trn} and E_{val} be the percentage error on the training and validation sets, respectively. The fitness (Q) of an individual was calculated as follows.

$$Q = b_e E_{trn} + (1 - b_e) E_{val} \tag{12}$$

where $b_e \in [0, 1]$ is a user-defined *balancing factor for error*. The genetic algorithm was left to evolve until no improvement was observed in the fitness (Q) of the best individual for 30 generations in a row. Then, the testing (data) set was applied once and the corresponding testing data percentage error E_{tst} was recorded.

5.1 Kimia-99 Data Set

Kimia-99 is a small subset of database MPEG-7 including labeled binary images. In particular, Kimia-99 includes 99 images partitioned in 9 classes with 11 images per class. All the 99 images in this data set are displayed in Fig.3 arranged in a 9×11 Table, where a row exclusively displays images in the same class. The name of a class is shown, respectively, at the end of a row in Fig.4. This data set was employed for illustrative purposes as detailed next.

We carried out Leave-1-Out classification experiments. That is, we carried out 99 different computational experiments leaving a single image out for testing, whereas all the remaining 98 images were employed for training. In turn, all the images were left out for testing. A validation set was not used here; hence $b_e = 1$.

We recorded no misclassifications. That is, the single datum left out for testing in all the 99 computational experiments was classified right. In each of the abovementioned experiments, a number of 3-tuple Type-II INs were induced for representing the classes. Fig.4 displays one 3-tuple Type-II IN per class as follows. The first three columns of the 9×4 Table in Fig.4 display Type-II INs

Table 1 Results of Leave-1-Out experiments on the Kimia-99 data set. A pair $(\overline{v}, \sigma_v)$ indicates the average $(\overline{v})$ and the corresponding standard deviation (σ_v) of a sigmoid positive valuation function $v_s(x; A_i, \lambda_i, \mu_i)$ parameter $v \in \{A_i, \lambda_i, \mu_i\}$ in a data dimension $i \in \{1, 2, 3\}$ for 99 computational experiments. The corresponding pair for the threshold size Z_0 was $(0.573, 0.262)$.

Data dimension i	$(\overline{A}_i, \sigma_{A_i})$	$(\overline{\lambda}_i, \sigma_{\lambda_i})$	$(\overline{\mu}_i, \sigma_{\mu_i})$
$i = 1$ (*FD* descriptors)	$(20.54, 11.02)$	$(16.71, 9.63)$	$(15.58, 12.05)$
$i = 2$ (*ART* descriptors)	$(1.825, 0.957)$	$(1.938, 0.800)$	$(1.496, 0.770)$
$i = 3$ (*IM* descriptors)	$(0.013, 0.008)$	$(0.018, 0.007)$	$(0.016, 0.008)$

corresponding, respectively, to *FD*, *ART* and *IM* descriptors; the fourth column displays the corresponding class name. Hence, the first row in Fig.4 displays an induced "Type-II 3-tuple IN" granule for class FISH, the second row displays a corresponding granule for class RABBIT, etc. We point out that the lower/upper bound $U/W \in \tau_O(\mathsf{F})$ of a Type-II IN, say $\mathbb{E} = [U, W]$, in Fig.4 is shown in bold (black) color; whereas, all the *encoded* Type-I INs are shown in light (red) color within a Type-II IN.

The set of Type-II INs representing a class (Fig.4) may be interpreted as the visual representation of the class in question. A closer study of our results revealed that, first, the Type-II INs shown in the first column (header: FD) of Fig.4 are, in general, similar to one another. Second, the Type-II INs in the second column (header: ART) of Fig.4 are quite dissimilar from one another. Third, the Type-II INs in the third column of Fig.4 (header: IM) are quite similar to one another. The latter similarity implicitly suggests that the *IM* descriptors are not significant in this pattern recognition problem.

The small significance of the *IM* descriptors was confirmed by the small parameter A_3 value of the corresponding sigmoid positive valuation function as shown in Table 1. The latter Table also suggests that the large (average) parameter A_1 value for *FD* descriptors is partly neutralized by the large (average) parameter $\overline{\mu}_1$ value; hence, the training/testing data appear in the saturation region of the corresponding (sigmoid) positive valuation function. Note that in a recently reported evaluation of individual descriptors including *FD*, *ART* and *IM* in an image retrieval application [1], the *IM* descriptors have resulted in the best performance. We believe that there is no contradiction with the results reported here because in this work we employ the descriptors *FD*, *ART* and *IM* "combined" instead of "individually". It is interesting to point out that, in a system modeling context, a combination of (system) inputs is already known to result in better performance than any (system) input alone [37].

Table 2 Classification results of Leave-1-Out experiments on the Kimia-216 data set.

Method	Classification Accuracy
Tree-Union	97.7 %
Class-Segment-Sets	97.2 %
FLRtypeII	95.4 %
Skeleton-Based	94.1 %

5.2 Kimia-216 Data Set

Kimia-216 is a larger, than Kimia-99, subset of database MPEG-7. In particular, Kimia-216 includes 216 labeled binary images partitioned in 18 classes with 12 images per class.

The Tree-Union classification method has reported 97.7% classification accuracy (Table 2) for Leaving-1-Out experiments [48]; more specifically, 5 images were reported misclassified. The aforementioned publication [48] has also reported classification results by another two methods shown in Table 2.

We carried out Leave-1-Out experiments using our proposed FLRtypeII classifier. A validation set was not used here; hence $b_e = 1$. We recorded 10 misclassifications in 216 experiments. Therefore, the accuracy of our proposed FLRtypeII classifier was 95.4% as shown in Table 2. Furthermore, the average threshold size Z_0 in 216 training experiments was $Z_0 = 0.417$ with standard deviation $\sigma_{Z_0} = 0.237$.

5.3 Chicken-Pieces Data Set

This data set consists of 446 shapes of five different chicken parts, namely wing, breast, leg, thigh and quarter (Fig.5). Following the literature we randomly divided this data set into three subsets of 149 shapes for training, 149 shapes for validation and 148 shapes for testing [11].

The Kernel-Edit Distance method in [11] has reported 87.16% classification accuracy on the testing data. The aforementioned publication also reports the classification results by the alternative methods shown in Table 3.

We applied our proposed FLRtypeII classifier using the validation set available in this problem with $b_e = 0.5$. Then, the testing data were applied. We recorded 20 misclassifications. Therefore, the classification accuracy of our proposed FLRtypeII classifier was 86.48% (Table 3). Our genetic optimization algorithm estimated an optimal threshold size $Z_0 = 0.114$.

Table 3 Classification results on the Chicken-Pieces data set.

Method	Classification Accuracy
Kernel-Edit Distance	87.16 %
FLRtypeII	86.48 %
Symbolic	84.45 %
Kernel-PCA	81.75 %
Structural	81.10 %
Kernel-LDA	80.40 %

6 Conclusion

This technical work has demonstrated novel lattice-computing (LC) techniques for both 2D shape representation and recognition.

Based on a hierarchy of lattices stemming from the lattice $(\mathsf{R}, \leq)$ of real numbers, the FLRtypeII scheme was introduced for learning and/or recognizing Type-II-IN-based representations of binary image 2D shapes; where, a Type-II IN was induced from a population of image descriptors in a data preprocessing step.

Experimental results have demonstrated, comparatively, the effectiveness of our proposed techniques. In addition, the FLRtypeII scheme induced Type-II-IN-based class representations/conceptualizations.

In a future work we plan to extend our techniques to gray/color image applications by considering additional and/or alternative populations of image descriptors. We also plan applications of larger scale. Furthermore, our proposed techniques might be employed by interval-valued morphological operators [34].

Acknowledgements This work has been supported, in part, by a project Archimedes-III contract.

Appendix

This Appendix summarizes useful notation and tools regarding general lattice theory [6, 19].

I. Mathematical Background Given a set P, a binary relation $(\preceq)$ in P is called *partial order* if and only if it satisfies the following conditions: $x \preceq x$ (*reflexivity*), $x \preceq y$ and $y \preceq x \Rightarrow x = y$ (*antisymmetry*), and $x \preceq y$ and $y \preceq z \Rightarrow x \preceq z$ (*transitivity*) – We remark that the *antisymmetry* condition may be replaced by the following equivalent condition: $x \preceq y$ and $x \neq y \Rightarrow y \not\preceq x$. If both $x \preceq y$ and $x \neq y$ then we write $x \prec y$. A *partially ordered set*, or *poset* for short, is a pair $(P, \preceq)$, where P is a set and $\preceq$ is a partial order relation in P. Note that this work

employs, first, "curly" symbols $\preceq$, $\prec$, $\curlyvee$, $\curlywedge$ for general poset elements and, second, "straight" symbols $\leq$, $<$, $\vee$, $\wedge$ for real numbers.

A *lattice* is a poset $(\mathsf{L}, \preceq)$ any two of whose elements $x, y \in \mathsf{L}$ have both a *greatest lower bound*, or *meet* for short, and a *least upper bound*, or *join* for short, denoted by $x \curlywedge y$ and $x \curlyvee y$, respectively. Two elements $x, y \in \mathsf{L}$ in a lattice $(\mathsf{L}, \preceq)$ are called *comparable*, symbolically $x \sim y$, if and only if it is either $x \preceq y$ or $x \succ y$. A lattice $(\mathsf{L}, \preceq)$ is called *totally-ordered* if and only if $x \sim y$ for $x, y \in \mathsf{L}$. For example, a totally-ordered lattice is the poset $(\mathsf{R}, \leq)$ of real numbers. If $x \nsim y$ holds for x, y in a lattice $(\mathsf{L}, \preceq)$ then x and y are called *incomparable* or, equivalently, *parallel*, symbolically $x||y$.

Given a lattice $(\mathsf{L}, \preceq)$ it is known that $(\mathsf{L}, \preceq^{\partial}) \equiv (\mathsf{L}, \succeq)$ is also a lattice, namely *dual* (lattice), where $\preceq^{\partial}$ denotes the *dual* (i.e. converse) of order relation $\preceq$. Furthermore, it is known that the product $(\mathsf{L}_1, \preceq) \times (\mathsf{L}_2, \preceq)$, of two lattices $(\mathsf{L}_1, \preceq)$ and $(\mathsf{L}_2, \preceq)$, may define another lattice with order $(x_1, x_2) \preceq (y_1, y_2) \Leftrightarrow x_1 \preceq y_1$ and $x_2 \preceq y_2$. In the latter (product) lattice it holds both $(x_1, x_2) \curlywedge (y_1, y_2) = (x_1 \curlywedge y_1, x_2 \curlywedge y_2)$ and $(x_1, x_2) \curlyvee (y_1, y_2) = (x_1 \curlyvee y_1, x_2 \curlyvee y_2)$. It follows that the product $(\mathsf{L}, \succeq) \times (\mathsf{L}, \preceq) \equiv (\mathsf{L} \times \mathsf{L}, \succeq \times \preceq)$ is another lattice with order $(x_1, x_2) \preceq (y_1, y_2) \Leftrightarrow x_1 \succeq y_1$ and $x_2 \preceq y_2$; moreover, $(x_1, x_2) \curlywedge (y_1, y_2) = (x_1 \curlyvee y_1, x_2 \curlywedge y_2)$ and $(x_1, x_2) \curlyvee (y_1, y_2) = (x_1 \curlywedge y_1, x_2 \curlyvee y_2)$. An element of lattice $(\mathsf{L} \times \mathsf{L}, \succeq \times \preceq)$ here will be denoted by a pair of L elements within square brackets, e.g. $[a, b]$.

Our interest, in the context of this work, is in *complete* lattices – We point out that complete lattices are significant also in mathematical morphology [35]. Recall that a lattice $(\mathsf{L}, \preceq)$ is called *complete* when each of its subsets X has both a greatest lower bound and a least upper bound in L; hence, for $X = \mathsf{L}$ it follows that a complete lattice has both a *least* and a *greatest* element. Unless otherwise indicated, this work employs the same symbols O and I to denote the least and the greatest element, respectively, in any complete lattice. Likewise, this work employs the same symbol $\preceq$ to denote a partial order relation.

Especially interesting, in the context of this work, is the complete lattice $(\tau_O(\mathsf{L}), \preceq)$ of intervals; the latter (lattice) is a sublattice of $(\mathsf{L} \times \mathsf{L}, \succeq \times \preceq)$ such that the corresponding *join* and *meet* are, respectively, given by $[a, b] \curlyvee [c, d] = [a \curlywedge c, b \curlyvee d]$ and either $[a, b] \curlywedge [c, d] = [a \curlyvee c, b \curlywedge d]$ if $a \curlyvee c \preceq b \curlywedge d$ or $[a, b] \curlywedge [c, d] = [I, O]$ otherwise. Note that $(\tau(\mathsf{L}), \preceq)$ denotes the poset of conventional lattice intervals $[a, b]$ with $a \preceq b$. We point out that lattice $(\tau_O(\mathsf{L}), \preceq)$ differs from poset $(\tau(\mathsf{L}), \preceq)$ in a single element, that is the least element $[I, O]$ of lattice $(\mathsf{L} \times \mathsf{L}, \succeq \times \preceq)$ which (least element) belongs to $(\tau_O(\mathsf{L}), \preceq)$ but not to $(\tau(\mathsf{L}), \preceq)$ as detailed in [19]. Note that the least element $[I, O]$ in lattice $(\tau_O(\mathsf{L}), \preceq)$ represents the "empty" interval.

Consider the following definition.

Definition 5 Let $(\mathsf{L}, \preceq)$ be a complete lattice with least and greatest elements O and I, respectively. An *inclusion measure* in $(\mathsf{L}, \preceq)$ is a real function $\sigma : \mathsf{L} \times \mathsf{L} \rightarrow [0, 1]$, which satisfies the following conditions

I0. $\sigma(x, O) = 0, \forall x \neq O$.
I1. $\sigma(x, x) = 1, \forall x$.
I2. $x \curlywedge y \prec x \Rightarrow \sigma(x, y) < 1$.
I3. $u \preceq w \Rightarrow \sigma(x, u) \leq \sigma(x, w)$.

We remark that an inclusion measure $\sigma(x, y)$ can be interpreted as a (fuzzy) degree to which x is less than or equal to y; therefore notation $\sigma(x \preceq y)$ may be used instead of $\sigma(x, y)$. We also point out that our proposed inclusion measure σ might be considered in future mathematical morphology studies [45].

II. Useful Mathematical Tools Two different inclusion measures are presented next, based on a *positive valuation*[2] function.

Theorem 1 *Let function $v : \mathsf{L} \rightarrow \mathsf{R}$ be a positive valuation in a complete lattice $(\mathsf{L}, \preceq)$. Then, both functions* sigma-meet $\sigma_{\curlywedge}(x, y) = \frac{v(x \curlywedge y)}{v(x)}$ *and* sigma-join $\sigma_{\curlyvee}(x, y) = \frac{v(y)}{v(x \curlyvee y)}$ *are inclusion measures.*

We introduce two constraints on positive valuation functions as explained next. First constraint: "$v(O) = 0$" (in order to satisfy condition I0 of Definition 5). Second constraint: "$v(I) < +\infty$" (because a positive valuation function $v : \mathsf{L} \rightarrow \mathsf{R}$ implies a metric (distance) function $d : \mathsf{L} \times \mathsf{L} \rightarrow \mathsf{R}^{\geq 0}$ given by $d(a, b) = v(a \curlyvee b) - v(a \curlywedge b)$, moreover infinite distances between lattice elements are not desired).

A bijective (i.e. one-to-one) *dual isomorphic*[3] function $\theta : \mathsf{L} \rightarrow \mathsf{L}$ such that $x \prec y \Leftrightarrow \theta(x) \succ \theta(y)$, in a lattice $(\mathsf{L}, \preceq)$, will be used in this work for extending an inclusion measure from a complete lattice $(\mathsf{L}, \preceq)$ to the corresponding lattice $(\tau_O(\mathsf{L}), \preceq)$ of intervals. In the first place note that given a dual isomorphic function $\theta : \mathsf{L} \rightarrow \mathsf{L}$ there follow, by definition, both $\theta(x \curlywedge y) = \theta(x) \curlyvee \theta(y)$ and $\theta(x \curlyvee y) = \theta(x) \curlywedge \theta(y)$. The latter equalities are handy in proving the following Proposition [22].

Proposition 1 *Let real function $v : L \rightarrow R$ be a* positive valuation *in a lattice $(\mathsf{L}, \preceq)$; moreover, let bijective function $\theta : L \rightarrow L$ be* dual isomorphic *in $(\mathsf{L}, \preceq)$ such that $x \prec y \Leftrightarrow \theta(x) \succ \theta(y)$. Then, function $v_{\Delta} : \mathsf{L} \times \mathsf{L} \rightarrow R$ given by $v_{\Delta}(a, b) = v(\theta(a)) + v(b)$ is a positive valuation in lattice $(\mathsf{L} \times \mathsf{L}, \succeq \times \preceq)$.*

[2] *Positive valuation* in a lattice $(\mathsf{L}, \preceq)$ is a real function $v : \mathsf{L} \rightarrow \mathsf{R}$ that satisfies both $v(x) + v(y) = v(x \curlywedge y) + v(x \curlyvee y)$ and $x \prec y \Rightarrow v(x) < v(y)$.

[3] A function $\psi : (P, \preceq) \rightarrow (Q, \preceq)$, between posets $(P, \preceq)$ and $(Q, \preceq)$, is called *(order) isomorphic* iff both "$x \preceq y \Leftrightarrow \psi(x) \preceq \psi(y)$" and "$\psi$ is onto Q". Then, posets $(P, \preceq)$ and $(Q, \preceq)$ are called *isomorphic*, symbolically $(P, \preceq) \cong (Q, \preceq)$.

The following definition is handy in this work.

Definition 6 Given a positive valuation $v : \mathsf{L} \to \mathsf{R}$ in a lattice $(\mathsf{L}, \preceq)$ the *size* of an interval $[a, b]$, in poset $(\tau(\mathsf{L}), \preceq)$, is a non-negative (real) function $Z : \mathsf{L} \to \mathsf{R}^{\geq 0}$ given by $Z([a, b]) = v(b) - v(a)$.

We remark that the size of a trivial interval $[x, x]$ equals $Z([x, x]) = 0$. Furthermore, we point out that a trivial interval $[x, x] = a$ is an *atom* in the complete lattice $(\tau_O(\mathsf{L}), \preceq)$, where an atom a by definition satisfies both $O \prec a$ and there is no interval $t \in (\tau_O(\mathsf{L}), \preceq)$ such that $O \prec t \prec a$.

References

1. Amanatiadis, A., Kaburlasos, V.G., Gasteratos, A., Papadakis S.E.: Evaluation of shape descriptors for shape-based image retrieval. IET Image Processing (in press)
2. Andreu, G., Crespo, A., Valiente, J.M.: Selecting the toroidal self-organizing feature maps (TSOFM) best organized to object recognition. In: Proceedings of the International Conference on Neural Networks 1997, vol. 2, pp. 1341–1346
3. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. IEEE Transactions on Pattern Analysis and Machine Intelligence **24**(4), 509–522 (2002)
4. Berretti, S., Del Bimbo, A., Pala, P.: Retrieval by shape similarity with perceptual distance and effective indexing. IEEE Transactions on Multimedia **2**(4), 225–239 (2000)
5. Biasotti, S., Cerri, A., Frosini, P., Giorgi, D., Landi, C.: Multidimensional size functions for shape comparison. Journal of Mathematical Imaging and Vision **32**(2), 161–179 (2008)
6. Birkhoff, G.: Lattice Theory, Colloquium Publications 25. American Mathematical Society, Providence, RI (1967)
7. Bloch, I.: Spatial reasoning under imprecision using fuzzy set theory, formal logics and mathematical morphology. International Journal of Approximate Reasoning **41**(2), 77–95 (2006)
8. Bloch, I., Maitre, H.: Fuzzy mathematical morphologies: a comparative study. Pattern Recognition **28**(9), 1341–1387 (1995)
9. Bober, M.: MPEG-7 visual shape descriptors. IEEE Transactions on Circuits and Systems for Video Technology **11**(6), 716–719 (2001)
10. Braga-Neto, U., Goutsias, J.: A theoretical tour of connectivity in image processing and analysis. Journal of Mathematical Imaging and Vision **19**(1), 5–31 (2003)
11. Daliri, M.R., Torre, V.: Shape recognition based on kernel-edit distance. Computer Vision and Image Understanding **114**(10), 1097–1103 (2010)
12. Deng, T.-Q., Heijmans, H. J.A.M.: Grey-scale morphology based on fuzzy logic. Journal of Mathematical Imaging and Vision **16**(2), 155–171 (2002)
13. Flusser, J., Suk, T.: Pattern recognition by affine moment invariants. Pattern Recognition **26**(1), 167–174 (1993)
14. Ganter, B., Wille, R.: Formal Concept Analysis. Springer, Heidelberg, Germany (1999)
15. Graña, M.: Lattice Computing and Natural Computing – Guest Editorial. Neurocomputing **72**(10-12), 2065–2066 (2009)

16. Graña, M., Villaverde, I., Maldonado, J.O., Hernandez, C.: Two lattice computing approaches for the unsupervised segmentation of hyperspectral images. Neurocomputing **72**(10-12), 2111–2120 (2009)
17. Graña, M., Savio, A.M., García-Sebastián, M., Fernandez, E.: A lattice computing approach for on-line fMRI analysis. Image and Vision Computing **28**(7), 1155–1161 (2010)
18. Graña, M., Chyzhyk, D., García-Sebastián, M., Hernández, C.: Lattice independent component analysis for functional magnetic resonance imaging. Information Sciences (in press)
19. Kaburlasos, V.G.: Towards a Unified Modeling and Knowledge-Representation Based on Lattice Theory, Studies in Computational Intelligence 27. Springer, Heidelberg, Germany (2006)
20. Kaburlasos, V.G.: Information Engineering Applications Based on Lattices – Guest Editorial. Information Sciences (in press)
21. Kaburlasos, V.G., Kehagias, A.: Novel fuzzy inference system (FIS) analysis and design based on lattice theory. IEEE Transactions on Fuzzy Systems **15**(2), 243–260 (2007)
22. Kaburlasos, V.G., Pachidis, T.: A lattice-computing ensemble for reasoning based on formal fusion of disparate data types, and an industrial dispensing application. (submitted)
23. Kaburlasos, V.G., Papadakis, S.E.: Granular self-organizing map (grSOM) for structure identification. Neural Networks **19**(5), 623–643 (2006)
24. Kaburlasos, V.G., Papadakis, S.E.: Fuzzy lattice reasoning (FLR) implies a granular enhancement of the fuzzy-ARTMAP classifier. In: Proceedings of JCIS, Salt Lake City, Utah, 2007, pp. 1610-1616
25. Kaburlasos, V.G., Papadakis, S.E.: A granular extension of the fuzzy-ARTMAP (FAM) neural classifier based on fuzzy lattice reasoning (FLR). Neurocomputing **72**(10-12), 2067–2078 (2009)
26. Kaburlasos, V.G., Petridis, V.: Fuzzy lattice neurocomputing (FLN) models. Neural Networks **13**(10), 1145–1169 (2000)
27. Kaburlasos, V.G., Amanatiadis, A., Papadakis, S.E.: 2-D shape representation and recognition by lattice computing techniques. In: Corchado, E., Graña, M., Savio, A.M. (eds.) Proc. Int. Conf. HAIS, San Sebastián, Spain, 2010. LNAI 6077, pp. 391–398. Springer (2010)
28. Kaburlasos, V.G., Athanasiadis, I.N., Mitkas, P.A.: Fuzzy lattice reasoning (FLR) classifier and its application for ambient ozone estimation. International Journal of Approximate Reasoning **45**(1), 152–188 (2007)
29. Kaburlasos, V.G., Moussiades, L., Vakali, A.: Fuzzy lattice reasoning (FLR) type neural computation for weighted graph partitioning. Neurocomputing **72**(10-12), 2121-2133 (2009)
30. Kim, J.-G., Noble, J.A., Brady, J.M.: Probabilistic models for shapes as continuous cuves. Journal of Mathematical Imaging and Vision **33**(1), 39–65 (2009)
31. Liao, S.X., Pawlak, M.: On the accuracy of Zernike moments for image analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence **20**(12), 1358–1364 (1998)
32. Ling, H., Jacobs, D.W.: Shape classification using the inner-distance. IEEE Transactions on Pattern Analysis and Machine Intelligence **29**(2), 286–299 (2007)
33. Maragos, P.: Lattice image processing: a unification of morphological and fuzzy algebraic systems. Journal of Mathematical Imaging and Vision **22**(2-3), 333–353 (2005)
34. Mélange, T., Nachtegael, M., Sussner, P., Kerre, E.E.: On the decomposition of interval-valued fuzzy morphological operators. Journal of Mathematical Imaging and Vision **36**(3), 270–290 (2010)
35. Nachtegael, M., Sussner, P., Mélange, T., Kerre, E.E.: On the role of complete lattices in mathematical morphology: from tool to uncertainty model. Information Sciences (in press)
36. Papadakis, S.E., Kaburlasos, V.G.: Piecewise-linear approximation of nonlinear models based on probabilistically/possibilistically interpreted intervals' numbers (INs). Information Sciences **180**(24), 5060–5076 (2010)

37. Papadakis, S.E., Tzionas, P., Kaburlasos, V.G., Theocharis, J.B.: A genetic based approach to the Type I structure identification problem. Informatica **16**(3), 365–382 (2005)
38. Pedrycz, W., Skowron, A., Kreinovich, V. (eds.): Handbook of Granular Computing. John Wiley & Sons, Chichester, England (2008)
39. Ritter, G.X., Wilson, J.N.: Handbook of Computer Vision Algorithms in Image Algebra, 2nd ed. CRC Press, Boca Raton, FL (2000)
40. Sebastian, T.B., Klein, P.N., Kimia, B.B.: Recognition of shapes by editing their shock graphs. IEEE Transactions on Pattern Analysis and Machine Intelligence **26**(5), 550–571 (2004)
41. Serra, J.: Image Analysis and Mathematical Morphology. Academic Press, London, UK (1982)
42. Sussner, P.: Generalizing operations of binary autoassociative morphological memories using fuzzy set theory. Journal of Mathematical Imaging and Vision **19**(2), 81–93 (2003)
43. Sussner, P., Esmi, E.L.: Morphological perceptrons with competitive learning: lattice-theoretical framework and constructive learning algorithm. Information Sciences (in press)
44. Sussner, P., Ritter, G.X.: Rank-based decompositions of morphological templates. IEEE Transactions on Image Processing **9**(8), 1420–1430 (2000)
45. Sussner, P., Valle, M.E.: Classification of fuzzy mathematical morphologies based on concepts of inclusion measure and duality. Journal of Mathematical Imaging and Vision **32**(2), 139–159 (2008)
46. Thakoor, N., Gao, J., Jung, S.: Hidden markov model-based weighted likelihood discriminant for 2-D shape classification. IEEE Transactions on Image Processing **16**(11), 2707–2719 (2007)
47. Tzafestas, C.S., Maragos, P.: Shape connectivity: multiscale analysis and application to generalized granulometries. Journal of Mathematical Imaging and Vision **17**(2), 109–129 (2002)
48. Wang, B., Shen, W., Liu, W.-Y., You, X.-G., Bai, X.: Shape classification using tree -unions. In: Proceedings of the IEEE 2010 20th International Conference on Pattern Recognition (ICPR), pp. 983–986
49. Xu, Y., Ruan, D., Qin, K., Liu, J.: Lattice-Valued Logic, Studies in Fuzziness and Soft Computing 132. Springer, Heidelberg, Germany (2003)
50. Zadeh, L.A.: From computing with numbers to computing with words – from manipulation of measurements to manipulation of perceptions. IEEE Transactions on Circuits and Systems – I: Fundamental Theory and Applications **45**(1), 105–119 (1999)
51. Zhang, D., Lu, G.: Shape-based image retrieval using generic Fourier descriptor. Signal Processing: Image Communication **17**(10), 825–848 (2002)
52. Zhang, D., Lu, G.: Review of shape representation and description techniques. Pattern Recognition **37**(1), 1–19 (2004)

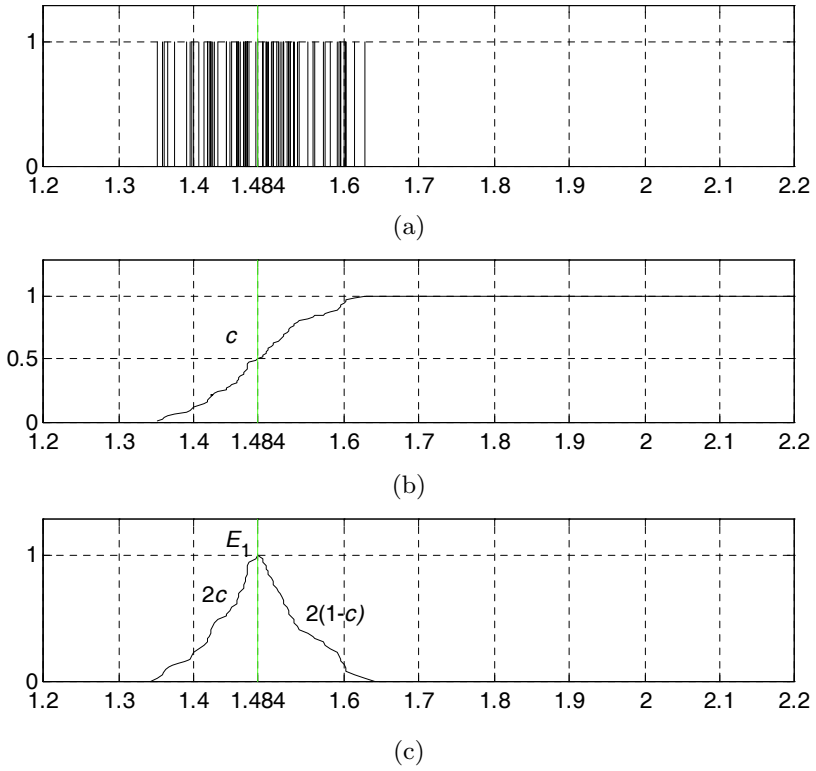

Fig. 1 Induction of a Type-I IN from a series of data samples whose median value equals 1.484. (a) The series of data samples. (b) The corresponding cumulative function c. (c) Computation of Type-I IN E_1 from function c above.

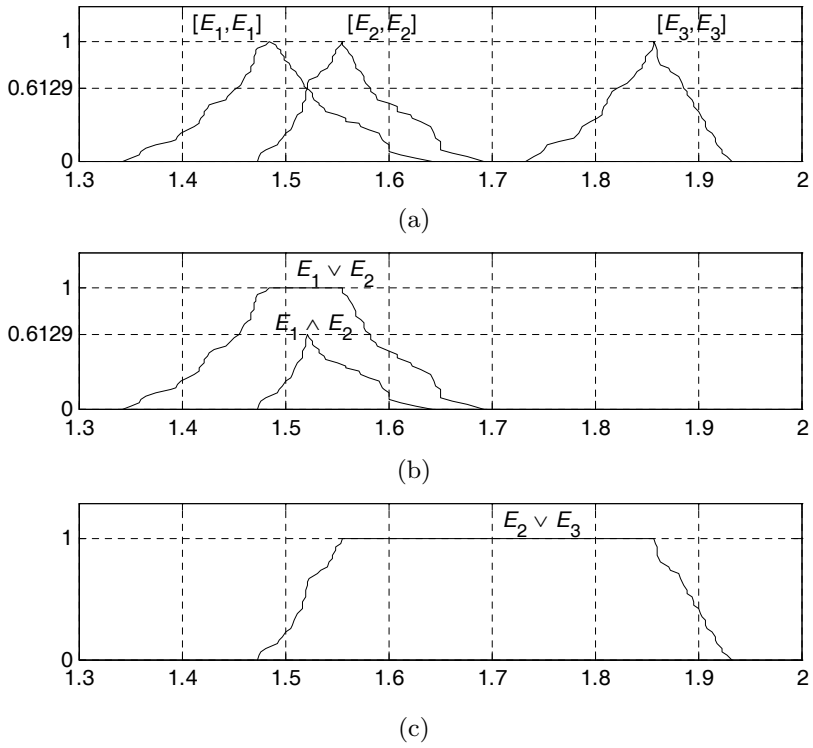

Fig. 2 Demonstrating the lattice *join* (Υ) operation between trivial Type-II INs. (a) Trivial Type-II INs $[E_1, E_1] = \mathbb{E}_1$, $[E_2, E_2] = \mathbb{E}_2$ and $[E_3, E_3] = \mathbb{E}_3$. (b) Type-II IN $\mathbb{E}_1 \Upsilon \mathbb{E}_2 = [E_1 \curlywedge E_2, E_1 \Upsilon E_2]$. Note that, since Type-I INs E_1 and E_2 overlap, the Type-I IN $E_1 \curlywedge E_2$ is not empty. More specifically, it is $(E_1 \curlywedge E_2)(h) \neq O = [+\infty, -\infty]$, for $h \in [0, 0.6129]$; nevertheless, for $h \in (0.6129, 1]$ it is $(E_1 \curlywedge E_2)(h) = O = [+\infty, -\infty]$. (c) Type-II IN $\mathbb{E}_2 \Upsilon \mathbb{E}_3 = [E_2 \curlywedge E_3, E_2 \Upsilon E_3] = [O, E_2 \Upsilon E_3]$. Note that, since Type-I INs E_2 and E_3 do not overlap, the Type-I IN $E_2 \curlywedge E_3$ is empty, that is $(E_2 \curlywedge E_3)(h) = O = [+\infty, -\infty]$, for $h \in [0, 1]$.

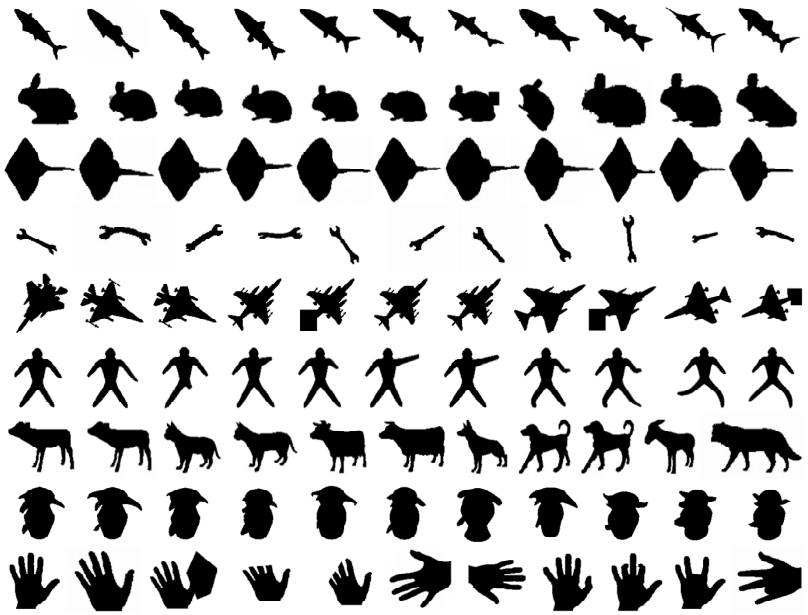

Fig. 3 The $9 \times 11 = 99$ Table of binary images above displays the complete Kimia-99 data set. More specifically, one row displays all the (11) images in a class for all the 9 different classes, namely FISH, RABBIT, KK, TOOL, FIGHTER, DUDE, FOURLEGGED, GENOME and HAND.

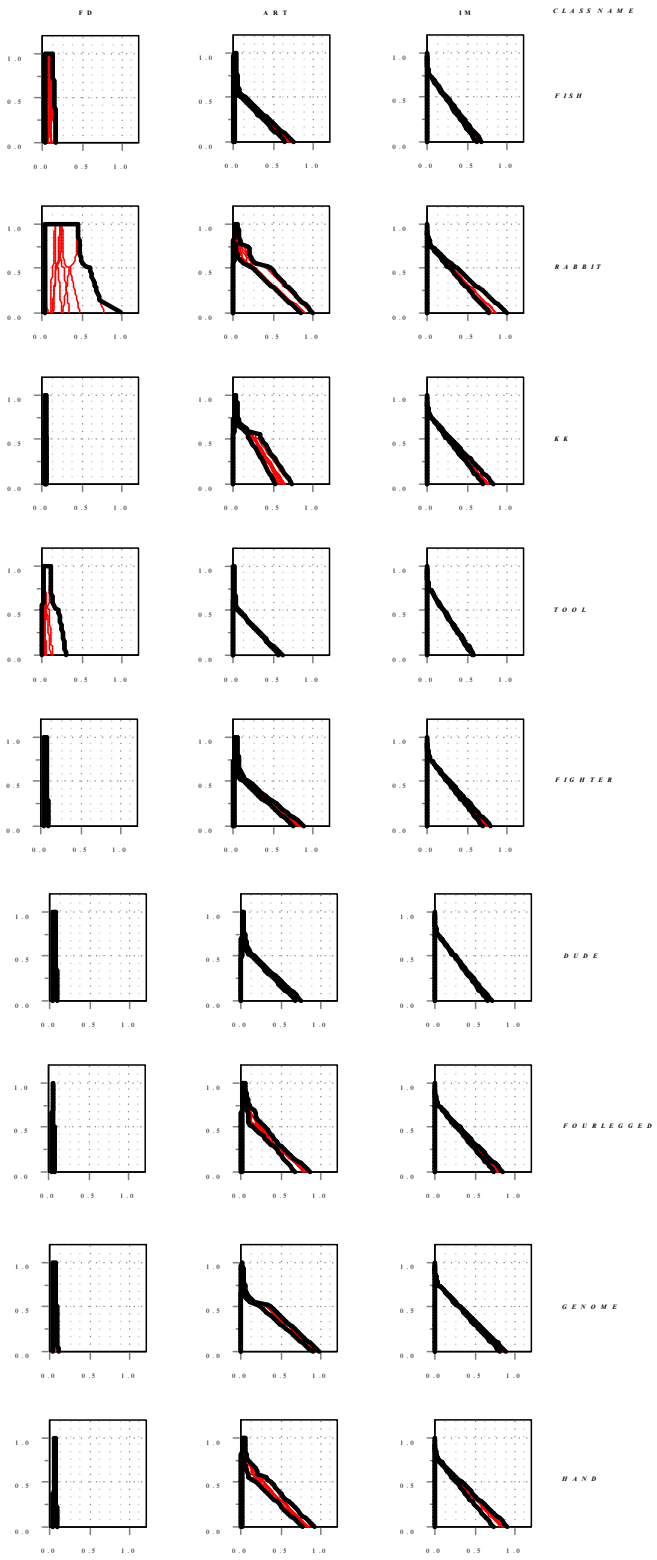

Fig. 4 The 9×4 Table above displays Type-II-IN granules for each of the 9 classes of the Kimia-99 data set. More specifically, one row of the Table above displays three Type-II INs corresponding to *FD*, *ART* and *IM* descriptors, respectively, as detailed in the text. At the end of a row, the corresponding class name is displayed.

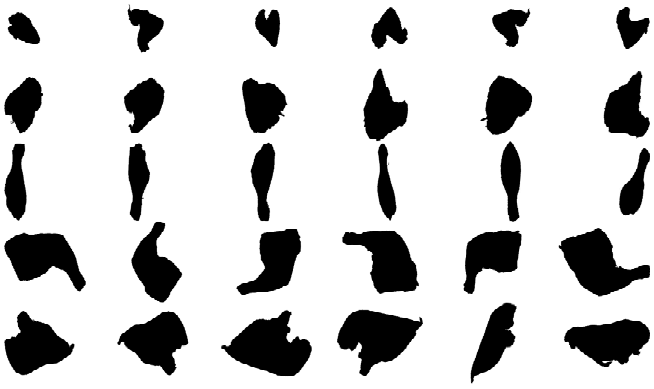

Fig. 5 The 5×6 Table of binary images above displays samples of the Chicken-Pieces data set. More specifically, a row of the Table above displays 6 pieces of chicken from 5 different chicken parts, namely wing, breast, leg, thigh and quarter.