Image Anal Stereol 2010;29:159-172 doi:10.5566/ias.v29.p159-172
Original Research Paper

SEMANTIC IMAGE ANALYSIS USING A SYMBOLIC NEURAL ARCHITECTURE

Ilianna Kollia, Nikolaos Simou, Andreas Stafylopatis and Stefanos Kollias

Image Video and Multimedia Systems Laboratory, Computer Science Division, School of Electrical and Computer Engineering, National Technical University of Athens, Heroon Polytechniou, 15780 Zografou, Greece
e-mail: stefanos@cs.ntua.gr

(Accepted July 30, 2010)

ABSTRACT

Image segmentation and classification are basic operations in image analysis and multimedia search which have gained great attention over the last few years due to the large increase of digital multimedia content. A recent trend in image analysis aims at incorporating symbolic knowledge representation systems and machine learning techniques. In this paper, we examine interweaving of neural network classifiers and fuzzy description logics for the adaptation of a knowledge base for semantic image analysis. The proposed approach includes a formal knowledge component, which, assisted by a reasoning engine, generates the a-priori knowledge for the image analysis problem. This knowledge is transferred to a kernel based connectionist system, which is then adapted to a specific application field through extraction and use of MPEG-7 image descriptors. Adaptation of the knowledge base can be achieved next. Combined segmentation and classification of images, or video frames, of summer holidays, is the field used to illustrate the good performance of the proposed approach.

Keywords: fuzzy description logics, kernel based connectionist systems, machine learning, semantic image analysis.

INTRODUCTION

Automatic image segmentation has been one of the major problems in the area of image processing and computer vision. For that reason a plethora of techniques has been proposed in the literature, including feature clustering (Comaniciu and Meer, 2002; Carson *et al.*, 2002), mathematical morphology (Meyer and Beucher, 1990) and graph-based techniques (Morris *et al.*, 1986; Felzenszwalb and Huttenlocher, 2004). Furthermore, in many cases, machine learning techniques are used to handle specific classification and adaptation issues (Papamarkos *et al.*, 2000; Naphade and Huang, 2001; Zhang *et al.*, 2001; Christel and Hauptmann, 2005; Spyrou *et al.*, 2009).

Research efforts have focused on incorporating certain knowledge about the domain, in which an image belongs to, providing semantically rich image segmentation. In this framework, Borenstein *et al.* (2004) proposed the combination of top-down (model-driven) and bottom-up segmentation, where information of the image level can solve ambiguities during the steps of a region-based segmentation process. In Luo and Savakis (2001) a Bayesian network was used to include low- and mid-level features for the classification of indoor or outdoor images; unsupervised fuzzy classification of regions was used for segmentation purposes in Lee and Crawford (2001). Spatial information about the regions of an image has been used to reduce the size of possible solutions, increasing the accuracy of segmentation and object recognition (Millet *et al.*, 2002). Lately, the usage of semantic analysis in multimedia applications has gained great attention (Stamou and Kollias, 2005) also reflected in recent European R&D activities (see for example IST FP6/FP7 projects Acemedia, Muscle, K-Space, X-Media, Mesh and Weknowit) [1].

Intelligent systems based on symbolic knowledge processing and artificial neural networks differ substantially. Nevertheless, they are both standard approaches to artificial intelligence and it is very desirable to combine the robustness of neural networks with the expressiveness of symbolic knowledge representation. This is the reason why the importance of efforts to bridge the gap between the connectionist and symbolic paradigms of artificial intelligence has been widely recognized. As the amount of hybrid data containing symbolic and statistical elements, as well as noise, increases, in diverse areas, such as bioinformatics, multimedia web mining, or multimodal application scenarios, neural-symbolic learning and reasoning becomes of particular practical importance. The merging of theory (background knowledge) and data learning (learning

[1] http://www.image.ece.ntua.gr/php/rd.php

from examples) has been indicated to provide learning systems that are more effective than purely symbolic and purely connectionist systems, especially when data are noisy. This has contributed to the growing interest in developing neural-symbolic systems (Pinkas, 1991; Garcez and Zaverucha, 1999; Garcez *et al.*, 2001; Hitzler *et al.*, 2004; Hammer and Hitzler, 2007).

This integration can be realized by an incremental workflow for knowledge adaptation. Symbolic knowledge bases can be embedded into a connectionist representation, where the knowledge can be adapted and enhanced from raw data. This knowledge may in turn be extracted into symbolic form, where it can be further used. This workflow is generally known as the neural-symbolic learning cycle (Hammer and Hitzler, 2007), as depicted in the following diagram (Fig. 1).

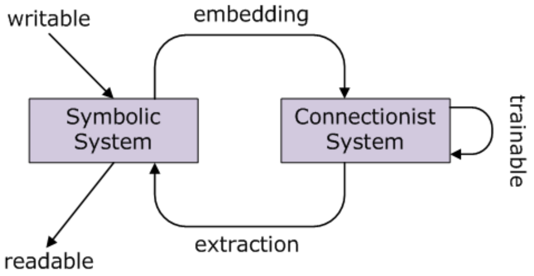


Fig. 1. *The neural-symbolic learning cycle.*

This paper focuses on developing a novel method for achieving connectionist adaptation of ontological knowledge represented by expressive fuzzy description logics. Moreover, it is shown that this method can be effectively used in real life multimedia applications, so as to improve the performance of image segmentation and classification methods.

In particular, a knowledge base generated using fuzzy description logics together with a reasoning engine comprise the symbolic part of the system. Recent research results that extract parameter kernel functions from Description Logics (DL) ontological representations are adopted for embedding the above knowledge to a kernel-based connectionist architecture. Adaptation of the connectionist system is performed as follows.

MPEG-7 image descriptors are first extracted from still images, or video image frames. A k-nearest neighbor algorithm is proposed, based on the MPEG-7 features and a correlation distance measure, to relate each new input data vector to one of the individuals included in the DL ontology, so that the connectionist system classifies it in a specific category. Whenever such a classification is not evaluated positively (*e.g.*, at the user environment), retraining of the connectionist system is performed, adapting its weights so as to provide good results in the specific application field. The new information, consisting of the new individual and its properties, is then transferred to the knowledge base, where they are evaluated and possibly used to update concepts and relations.

The resulting scheme can be implemented and used in real-life multimedia applications, in contrast to other, afore-mentioned schemes that have not shown such capability up to the present. Segmentation and classification of images of summer holidays is used as the application field illustrating the good performance of the proposed connectionist-symbolic analysis scheme and the obtained improvement over conventional machine learning methods.

The rest of the paper is organized as follows. The following section ("The proposed architecture") outlines the proposed architecture that mainly consists of the formal knowledge, the semantic interpretation layer and the knowledge adaptation components. These modules are described in detail in Sections "The formal knowledge component", "Semantic interpretation" and "The knowledge adaptation mechanism" respectively. Section "A multimedia analysis experimental study" presents a multimedia analysis experimental study illustrating the theoretical developments and also a comparison with state of the art approaches on the same problem. Conclusions and planned future activities are presented in Section "Conclusions".

THE PROPOSED ARCHITECTURE

The proposed system is based on a learning, evolving and adapting cognitive model. Starting with basic knowledge about the nature of the problem and by using powerful reasoning mechanisms, the proposed system gradually evolves its knowledge, by incorporating its observations along with its own or its user's evaluations.

Fig. 2 summarizes the proposed system architecture, consisting of two main components: the *Formal Knowledge* and the *Knowledge Adaptation*. The Formal Knowledge stores the terminology and assertions, *i.e.*, the constraints that describe the problem under analysis in the appropriate knowledge representation formalism. More specifically, the *Ontologies module* formally represents the general knowledge about the problem. It is actually a formal ontological description representing the concepts and relationships of the field, providing formal definitions and axioms that hold in every similar environment. This forms the system's knowledge

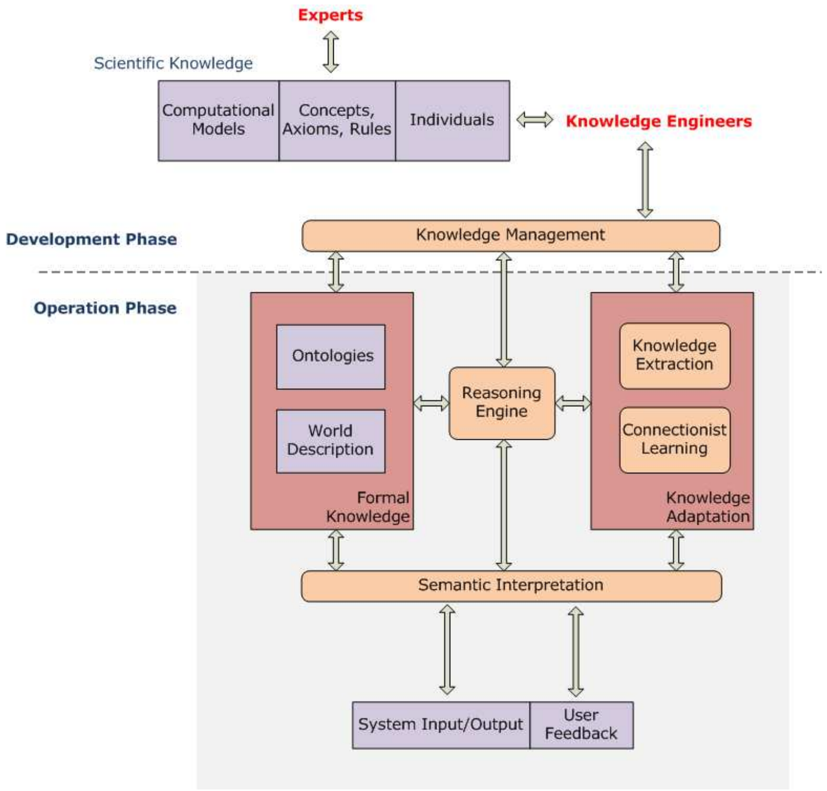


Fig. 2. *The semantic adaptation architecture.*

which is generally generated during the Development Phase by knowledge engineers and experts.

Moreover, the *Formal Knowledge* contains the *World Description* that is actually a representation of all objects and individuals of the world, as well as their properties and relationships in terms of the Ontology. It is evident that most of the above data involve different types of uncertain information and, thus, they can be represented as formal (fuzzy) description logic assertions connecting the objects and individuals of the world with the concepts and relationships of the Ontology. These assertions are provided automatically or semi-automatically by the *Semantic Interpretation* module.

In real environments however, this global knowledge representation is rather optimistic. Unfortunately, there may be lot of reasons that cause inconsistencies in the *Formal Knowledge*. For example, it is impossible to model all specific real environments and thus, in some cases, conflicting assertions can arise. As a more abstract example (and more difficult to handle), the personality and expressivity of a specific user makes some of the axioms and constraints of the Ontology non-applicable or even wrong, if applied in general to all user interactions. These inconsistencies make the formal use of knowledge that the *Reasoner* provides rather problematic. In such cases, the *Knowledge Adaptation* component of the system tries to resolve the inconsistency through a recursive learning process.

The knowledge adaptation improves the knowledge of the system by changing to some degree the axioms of the terminology of the system. The new information as represented in a connectionist model and, with the aid of learning algorithms, is adapted and then re-inserted in the knowledge base through the *Knowledge Extraction* module for adaptation purposes.

THE FORMAL KNOWLEDGE COMPONENT

FORMAL KNOWLEDGE AND CONNECTIONIST MODELS

The focus of the proposed system architecture in Fig. 2 is the adaptation of the knowledge base, so as to deal with contextual information and raw data peculiarities obtained from multimodal inputs. In this paper, we adopt recent results in formal knowledge representation and neural-symbolic integration. In particular, formal knowledge is transferred to a connectionist system and is adapted by means of machine learning algorithms. Knowledge extraction from trained networks is another important issue, which is included in the neural-symbolic loop, although not studied analytically in this paper.

KERNEL DEFINITION FOR DESCRIPTION LOGICS

In this section recent work that extracts parameter kernel functions for individuals within ontologies is presented (Bloehdorn and Sure, 2007; Fanizzi *et al.*, 2007; 2008). Exploitation of these kernels permits inducing classifiers for individuals in Semantic Web (OWL) ontologies. In this paper, extraction of kernel functions is the main outcome of the *Formal Knowledge* component – assisted by the reasoning engine – for feeding the connectionist-based *Knowledge Adaptation* module.

The basis for developing these functions in the framework of the formal knowledge is the encoding of similarity between individuals, as they are presented to the knowledge base of the system, by exploiting semantic aspects of the reference representations.

The family of kernel functions is defined as $k_p^F : \mathrm{Ind}(A) \times \mathrm{Ind}(A) \to [0,1]$, for a knowledge base $K = \langle T, A \rangle$ consisting of the TBox T (set of terminological axioms of concept descriptions-Ontology) and the ABox A (assertions on the world state-World Description); $\mathrm{Ind}(A)$ indicates the set of individuals appearing in A, and $F = \{F_1, F_2, \ldots, F_m\}$ is a set of concept descriptions. These functions are defined as the L_p mean of the, say m, simple concept kernel functions κ_i, $i = 1, \ldots, m$, where, for every two individuals a,b, and $p > 0$,

$$\kappa_i(a,b) = \begin{cases} 1 & (F_i(a) \in A \wedge F_i(b) \in A) \vee \\ & (\neg F_i(a) \in A \wedge \neg F_i(b) \in A); \\ 0 & (F_i(a) \in A \wedge \neg F_i(b) \in A) \vee \\ & (\neg F_i(a) \in A \wedge F_i(b) \in A); \\ \frac{1}{2} & \text{otherwise.} \end{cases} \tag{1}$$

$$\forall a, b \in \mathrm{Ind}(A) \quad k_p^F(a,b) := \left[\sum_{i=1}^{m} \left| \frac{\kappa_i(a,b)^p}{m} \right| \right]^{1/p}. \tag{2}$$

The rationale of these kernels is that similarity between individuals is determined by their similarity with respect to each concept F_i, *i.e.*, if they both are instances of the concept or of its negation. Because of the Open World Assumption for the underlying semantics, a possible uncertainty in concept membership is represented by an intermediate value of the kernel. A value of $p = 1$ has generally been used for implementing Eq. 2 in Fanizzi *et al.* (2008). In our case, we have used the mean value of the above kernel, which is computed through high level feature relations, and a normalized linear kernel which is computed through low level feature values, extracted from multimedia data as described in the knowledge adaptation mechanism section.

THE REASONING ENGINE

The reasoning engine, included in Fig. 2, is of major importance for the whole procedure, because it assists the operation of all knowledge related components. First, during the knowledge development phase, it is responsible for enriching manual generation of concepts and relations, so that computation of the kernels in Eqs. 1 and 2 includes the fewest ambiguities possible, and any inconsistencies are removed from the knowledge representation. In fact Eqs. 1 and 2 are computed, by relating every two individuals w. r. t. each concept in the knowledge base, by using the reasoning engine. In the operation phase, it interacts with the semantic interpretation layer and the connectionist system for achieving knowledge adaptation to real life environments. Both crisp and fuzzy reasoners can form this engine. In our case, we have been using the FIRE engine (Stoilos *et al.*, 2006).

The FIRE system is based on Description Logic f-SHIN (Stoilos *et al.*, 2007) that is a fuzzy extension of the DL SHIN (Horrocks *et al.*, 2000) and it similarly consists of an alphabet of distinct concept names (**C**), role names (**R**) and individual names (**I**). The domain of interest is represented by concepts and role descriptions using DLs constructors. The set of constructors specifies the name of the DL language (Baader *et al.*, 2002) and in the case of f-SHIN these are the ALC constructors (*i.e.*, negation $\neg$, conjunction $\sqcap$, disjunction $\sqcup$, full existential quantification $\exists$ and value restriction $\forall$) extended by transitive roles (S), roles hierarchy (H), inverse roles (I), and number restrictions (N). Hence, if R is a role then R^- is also a role, namely the inverse of R. f-SHIN concepts are inductively defined as follows:

1. If $C \in \mathbf{C}$, then C is a f-SHIN concept
2. If C and D are concepts, R is a role, S is a simple role and $n \in \mathbb{N}$, then $(\neg C)$, $(C \sqcup D)$, $(C \sqcap D)$, $(\forall R.C)$, $(\exists R.C)$, $(\geq nS)$ and $(\leq nS)$ are also f-SHIN concepts.

Differently to crisp DLs, the semantics of fuzzy DLs are given by a *fuzzy interpretation* (Straccia, 2001). A fuzzy interpretation is a pair $\mathscr{I} = \langle \Delta^{\mathscr{I}}, \cdot^{\mathscr{I}} \rangle$ where $\Delta^{\mathscr{I}}$ is a non-empty set of objects and $\cdot^{\mathscr{I}}$ is a fuzzy interpretation function, which maps an individual name a to elements of $\mathsf{a}^{\mathscr{I}} \in \Delta^{\mathscr{I}}$ and a concept name A (role name R) to a membership function $A^{\mathscr{I}} : \Delta^{\mathscr{I}} \to [0,1]$ ($R^{\mathscr{I}} : \Delta^{\mathscr{I}} \times \Delta^{\mathscr{I}} \to [0,1]$).

Fuzzy set theory is used in order to extend the interpretation function to give semantics to complex concepts, roles and axioms (Klir and Yuan, 1995). Therefore, fuzzy complement, indicated as c, is used to interpret negation constructor, s-norm (u) and t-norm (t) are used to interpret fuzzy union and fuzzy intersection constructors respectively, while implication is interpreted by fuzzy implication $\mathscr{J}$. In fuzzy set theory there are different functions for these operations that specify different fuzzy logics. In the DL f_{KD}-SHIN, Lukasiewicz complement $c_L(a) = 1-a$, Gödel t-norm $t_G(a,b) = \min(a,b)$, Gödel s-norm $u_G(a,b) = \max(a,b)$ and Kleene-Dienes implication $\mathscr{J}_{KD}(a,b) = \max(1-a,b)$ are used which form the Zadeh fuzzy logic (Klir and Yuan, 1995). The complete set of semantics is depicted in Table 1.

A f_{KD}-SHIN knowledge base Σ is a triple $\langle T,R,A \rangle$, where T is a fuzzy *TBox* (Terminological Box), R is a fuzzy *RBox* (Role Box) and A is a fuzzy *ABox* (Assertional Box). *TBox* is a finite set of fuzzy concept axioms which are of the form $C \equiv D$ called fuzzy concept equivalence axioms or $C \sqsubseteq D$ called fuzzy concept inclusion axioms saying that C is equivalent or C is a sub-concept of D, respectively. Similarly, *RBox* is a finite set of fuzzy role axioms of the form $\mathsf{Trans}(R)$ called fuzzy transitive role axioms and $R \sqsubseteq S$ called fuzzy role inclusion axioms saying that R is transitive and R is a sub-role of S, respectively. Finally, *ABox* is a finite set of fuzzy assertions of the form $\langle a : C \bowtie n \rangle$, $\langle (a,b) : R \bowtie n \rangle$, where $\bowtie$ stands for $\geq$, $>$, $\leq$, $<$ or $a \not\doteq b$, for $a,b \in \mathbf{I}$. Fuzzy representation enriches expressiveness, so a fuzzy assertion of the form $\langle a : C \geq n \rangle$ means that a participates in the concept C with a membership degree that is at least equal to n.

The main reasoning services supported by crisp reasoners are *entailment* and *subsumption*. These services are also available by FiRE together with greatest lower bound queries which incorporate the fuzzy element. Since a fuzzy *ABox* might contain many positive assertions for the same individual, without forming a contradiction, it is of interest to compute what is the best lower and upper truth-value bounds of a fuzzy assertion. The term of *greatest lower bound* (GLB) of a fuzzy assertion with respect to a knowledge base has been defined in (Straccia, 2001).

The reason why we use fuzzy reasoning is that fuzzy assertional component permits more detailed descriptions of a domain. Furthermore, the fact that we deal with image analysis algorithms that can provide rich though imperfect results makes fuzzy DLs the most appropriate formalism for such a representation. On the other hand this representation requires a more sophisticated way for the kernels evaluation. In order to compute Eqs. 1 and 2 the GLB reasoning service of FiRE is used, but the resulting greatest lower bound is treated using a threshold. In other words, if GLB for $F_i(a) > 0.5$, then $F_i(a) \in A$, while if GLB for $F_i(a) < 0.5$, then $\neg F_i(a) \in A$. In that way we incorporate the uncertainty of image analysis algorithms in the creation of the kernels. As a future extension, we intend to further work on the incorporation of the fuzzy element in the estimation of kernel functions using, as well, fuzzy operations like fuzzy aggregation and fuzzy weighted norms for the individual's evaluation.

SEMANTIC INTERPRETATION

The main operation of the semantic interpretation (SI) layer is to create the assertional component of the knowledge base, in other words to link the individuals with the concepts. In the experimental study of this paper, the SI layer includes a segmentation algorithm, an algorithm for extraction of low level MPEG-7 features from image segments, and an algorithm for matching image segments (individuals) based on correlation of their feature values. A semantic variation of the Recursive Shortest Spanning Tree (Morris *et al.*, 1986) constitutes the main segmentation method, which is also responsible for extracting spatial relations among image segments. These relations are inserted in the knowledge base and used by the reasoning engine. Moreover, it merges neighboring regions, based on their labeling, or obtained classification.

FEATURE EXTRACTION AND CLASSIFICATION

The given image, or video frame, is initially processed by a hierarchical segmentation algorithm (Adamek *et al.*, 2005) which partitions it in a number of regions. Standard MPEG-7 low level visual features, especially color and texture metrics, are extracted, based on the values of pixels belonging in each region. Extraction of these low-level descriptors is performed using the Visual Descriptor Extraction tool [2]. VDE has been developed based on the experimentation Model (XM) of MPEG-7 (Yamada *et al.*, 2001). It uses XM extraction algorithms, optimized in order to provide a faster performance, while remaining fully compatible to the XM in terms of the MPEG-7 descriptors. Since regions usually share the property of homogeneity of a certain feature (color and/or texture), it is possible

[2] http://image.ntua.gr/iva/tools/vde

Table 1. *The f_{KD}-SHIN syntax and semantics.*

Constructor	Syntax	Semantics
top	$\top$	$\top^{\mathcal{I}}(a) = 1$
bottom	$\bot$	$\bot^{\mathcal{I}}(a) = 0$
general negation	$\neg C$	$(\neg C)^{\mathcal{I}}(a) = c(C^{\mathcal{I}}(a))$
conjunction	$C \sqcap D$	$(C \sqcap D)^{\mathcal{I}}(a) = t(C^{\mathcal{I}}(a), D^{\mathcal{I}}(a))$
disjunction	$C \sqcup D$	$(C \sqcup D)^{\mathcal{I}}(a) = u(C^{\mathcal{I}}(a), D^{\mathcal{I}}(a))$
exists restriction	$\exists R.C$	$(\exists R.C)^{\mathcal{I}}(a) = \sup_{b \in \Delta^{\mathcal{I}}}\{t(R^{\mathcal{I}}(a,b), C^{\mathcal{I}}(b))\}$
value restriction	$\forall R.C$	$(\forall R.C)^{\mathcal{I}}(a) = \inf_{b \in \Delta^{\mathcal{I}}}\{\mathcal{J}(R^{\mathcal{I}}(a,b), C^{\mathcal{I}}(b))\}$
at-most	$\leq pR$	$\inf_{b_1,\ldots,b_{p+1} \in \Delta^{\mathcal{I}}} \mathcal{J}(t_{i=1}^{p+1} R^{\mathcal{I}}(a,b_i), u_{i<j}\{b_i = b_j\})$
at-least	$\geq pR$	$\sup_{b_1,\ldots,b_p \in \Delta^{\mathcal{I}}} t(t_{i=1}^{p} R^{\mathcal{I}}(a,b_i), t_{i<j}\{b_i \neq b_j\})$
inverse role	R^-	$(R^-)^{\mathcal{I}}(b,a) = R^{\mathcal{I}}(a,b)$
equivalence	$C \equiv D$	$\forall a \in \Delta^{\mathcal{I}}.C^{\mathcal{I}}(a) = D^{\mathcal{I}}(a)$
sub-concept	$C \sqsubseteq D$	$\forall a \in \Delta^{\mathcal{I}}.C^{\mathcal{I}}(a) \leq D^{\mathcal{I}}(a)$
transitive role	$\mathsf{Trans}(R)$	$\forall a,b \in \Delta^{\mathcal{I}}.R^{\mathcal{I}}(a,b) \geq \sup_{c \in \Delta^{\mathcal{I}}}\{t(R^{\mathcal{I}}(a,c), R^{\mathcal{I}}(c,b))\}$
sub-role	$R \sqsubseteq S$	$\forall a,b \in \Delta^{\mathcal{I}}.R^{\mathcal{I}}(a,b) \leq S^{\mathcal{I}}(a,b)$
concept assertions	$\langle a : C \bowtie n \rangle$	$C^{\mathcal{I}}(a^{\mathcal{I}}) \bowtie n$
role assertions	$\langle (a,b) : R \bowtie n \rangle$	$R^{\mathcal{I}}(a^{\mathcal{I}}, b^{\mathcal{I}}) \bowtie n$

to label them, using a simple classifier (Papadopoulos *et al.*, 2007). We may consider the classifier outputs, denoting the category that each region (*i.e.*, segment) belongs to, as fuzzy variables $\mu_a(C_k)$ showing the respective degree of confidence in the classification of segment α to each category C_k. This information feeds the semantic segmentation algorithm, which is presented next.

SEMANTIC SEGMENTATION

RSST is a bottom-up segmentation algorithm that starts its processing from the pixel level and iteratively merges similar neighboring regions until certain termination criteria are satisfied. It uses an Attributed Relation Graph (*ARG*) (Berretti *et al.*, 2001) that is an internal graph representation of image regions. In the beginning, all edges of the graph are sorted according to a criterion, such as the color dissimilarity of two connected regions, using the Euclidean distance of the color components. The edge with the least weight is found and the two regions connected by that edge, are then merged. After each step, the merged region's attributes (*e.g.*, region's mean color) are re-calculated. Additionally, RSST also re-calculates the weights of the related edges and re-sorts them. In that way, in every step the edge with the least weight is selected, the two neighboring regions by that edge are merged, and the process continues recursively until the algorithm meets the termination criteria. Such criteria may vary, but they are usually based on either the number of regions, or a threshold based on the dissimilarity distance.

This algorithm was modified in order to operate on the fuzzy sets, aiming at improving the usually over-segmentation results obtained by the former procedure, incorporating the acquired region labeling in the segmentation process (Athanasiadis *et al.*, 2007). The modification of the traditional algorithm to S-RSST lies on the definition of the two criteria:

1. The dissimilarity criterion between two adjacent regions a and b (vertices v_a and v_b in the graph), based on which the graph's edges are sorted, and
2. the termination criterion.

In order to calculate the similarity between two regions, a metric between two fuzzy sets that corresponds to the candidate concepts of the two regions is defined. This dissimilarity value is computed according to the following formula and is assigned as the weight of the respective graph's edge e_{ab}:

$$w(e_{ab}) = 1 - \sup_{c_k \in C} \left(t_G(\mu_a(c_k), \mu_b(c_k)) \right), \qquad (3)$$

where a and b are the two neighboring regions, $t_G(a,b) = \min(a,b)$, is Gödel t-norm and $\mu_a(c_k)$ is the degree of membership of the concept-region label $c_k \in C$ in the fuzzy set L_a.

Let us now examine one iteration of the S-RSST algorithm. Firstly, the edge e_{ab} with the least weight is selected, then regions a and b are merged. Vertex v_b

is removed completely from the ARG, whereas v_a is updated appropriately. This update procedure consists of the following two actions:

1. Re-evaluation of the membership degrees of the labels' fuzzy set in a weighted average (w. r. t. the regions' size) fashion.
2. Re-adjustment of the ARG edges by removing edge e_{ab} and re-evaluating the weight of the affected edges.

This procedure will continue until the edge with the least weight in the ARG e^*, is bigger than a threshold: $w(e^*) > T_w$. This threshold is calculated in the beginning of the algorithm, based on the histogram of all weights of the set of all edges.

THE KNOWLEDGE ADAPTATION MECHANISM

THE SYSTEM OPERATION PHASE

In the proposed architecture of Fig. 2, let us assume that the set of individuals (with their corresponding features and kernel functions), that have been used to generate the formal knowledge representation in the development phase, are provided, by the *Semantic Interpretation Layer*, to the *Knowledge Adaptation* component.

Support Vector Machines constitute a well known method which can be based on kernel functions to efficiently induce classifiers that work by mapping the instances into an embedding feature space, where they can be discriminated by means of a linear classifier. As such, they can be used for effectively exploiting the knowledge-driven kernel functions in Eqs. 1 and 2, and be trained to classify the available individuals in different concept categories included in the formal knowledge. In Fanizzi *et al.* (2008) it is shown that SVMs can exploit such kernels, so that they can classify the (same) individuals - used for extracting the kernels – accurately. A Kernel Perceptron is another connectionist method that can be trained using the set of individuals and applied to this linearly separable classification problem.

Let us assume that the system is in its – real life – operation phase. Then, the system deals with new individuals, with their corresponding – multimodal – input data and low level features being captured by the system and being provided through the semantic interpretation layer to the connectionist subsystem for classification to a specific concept. It is well known that due to local or user oriented characteristics, these data can be quite different from those of the individuals used in the training phase; thus they may be not well represented by the existing formal knowledge. In the following, we discuss the adaptation phase of the system to this local information, being realized through the adaptation of the connectionist architecture.

ADAPTATION OF THE CONNECTIONIST ARCHITECTURE

Whenever a new individual is presented to the system, it should be related, through the kernel function to each individual of the knowledge base w. r. t. a specific concept – category; the input data domain is, thus, transformed to another domain – taking into account the semantics that have been inserted to the kernel function.

There are some issues that should be solved in this procedure. The first is that the number of individuals can be quite large, so that transporting them in different user environments is quite difficult. A Principal Component Analysis (PCA), or a clustering procedure can reduce the number of individuals so as to be capable of effectively performing approximate reasoning. Consequently, it is assumed that through clustering, individuals become the centers of clusters, to which a new individual will be related through Eqs. 1 and 2.

The second issue is that the kernel function in Eqs. 1 and 2 is not continuous w. r. t. individuals. Consequently, the values of the kernel functions when relating a new individual to any existing one should be computed. To cope with this problem, it is assumed that the semantic relations, that are expressed through the above kernel functions, also hold for the syntactic relations of the individuals, as expressed by their corresponding low level features, estimated and presented at the system input. Under this assumption, a feature based matching criterion using a k-nearest neighbor, is used to relate the new individual to each one of the cluster centers w. r. t. the low level feature vector. Various techniques can be adopted for defining the value of the kernel functions at the resulting instances. A vector quantization type of approach, where each new individual is replaced by its closest neighbor, when computing the kernel value, is a straightforward choice. To extend the approach to a fuzzy framework, weighted averages and Gaussian functions around the cluster centers can be used to compute the new instances' kernel values.

In cases that classification of the new individual is evaluated as not correct by the user, the SVM or Kernel Perceptron are retrained - including the new individuals in the training data set, while getting the corresponding desired responses by the *User* or by the *Semantic Interpretation Layer* – thus, adapting its knowledge to the specific context and use.

The problem will, in parallel, be reported back to formal knowledge and reasoning mechanism, for updating system's knowledge for the specific context, and then (off-line) providing again the connectionist module of the user with a new, knowledge-updated, version of the system. This case is discussed in the following subsection.

ADAPTATION OF THE KNOWLEDGE BASE

Knowledge extraction from trained neural networks, *e.g.*, perceptrons, or neuro-fuzzy systems, has been a topic of extensive research (Kolman and Margaliot, 2005). Such methods can be used to transfer locally extracted knowledge to the central knowledge base. In our case, the – most characteristic – new individuals obtained in the local environment, together with the corresponding desired outputs – concepts of the knowledge base, can be transferred to the knowledge development module of the main system (Fig. 2), so that, with the assistance of the reasoning engine, the system's formal knowledge, *i.e.*, the TBox can be updated, w. r. t. the specific context or user.

More specifically, the new individuals obtained in the local environment form an ABox A'. In order to adapt a knowledge base (KB), according to the new world description, the concepts of interest, which are defined by axioms contained in the KB, must change appropriately. Since a concept of interest defined by an axiom is specified by some other concepts composed using DL constructors, adaptation of the KB can be achieved by the effective modification of the concepts and the constructors used.

However, both formal and connectionist part adaptation is based on the assumption that only small modifications of the a-priori knowledge are envisaged, caused by the specific context of the application, while the original knowledge and respective reasoning about the application field is generally valid. In this framework, axioms and concepts that are considered of major importance for the field are not adapted, thus restricting adaptation only to the remaining concepts. The concepts of an axiom are separated in these two categories from the knowledge engineer that defines the knowledge base.

Therefore, in order to adapt a knowledge base $K = \langle T, R, A \rangle$, for a defined concept F_i, using concepts of major importance, denoted as C, we check all the concepts for adaptation denoted as $R_1F_i \ldots R_nF_i$ under the specific context, *i.e.*, in A'. Let $|R_nF_i|$ denote the occurrences of $R_nF_i \in A'$, t denote a threshold defined according to the data size and $Axiom(F_i)$ denote the axiom defined for the concept F_i in the knowledge base (*i.e.*, $Axiom(F_i) \in T$). Furthermore, we write $R_nF_i \in Axiom(F_i)$ when the concept R_nF_i is used in $Axiom(F_i)$ and $R_nF_i \notin Axiom(F_i)$ when it is not used. Knowledge adaptation is made according to the following criteria:

$$|R_nF_i| = \begin{cases} [0,t/4) & \text{If } R_nF_i \in Axiom(F_i) \rightarrow \text{Remove } R_nF_i \text{ from } Axiom(F_i); \\ [t/4,t] & \text{No adaptation in } K; \\ > t & \text{If } R_nF_i \notin Axiom(F_i) \rightarrow Axiom(F_i) \cup R_nF_i. \end{cases} \tag{4}$$

Eq. 4 implies that the related concepts with the most occurrences in A' are selected for the adaptation of the terminology, while those that are not significant are removed. At this point we must note that the DL constructor that will be used for the incorporation of a concept, in order to adapt the knowledge base, is specified by the domain expert. Future work includes a semi-automatic selection of constructors, that will be based on the inconsistencies formed by the use of specific DL constructors for updating the knowledge base.

A MULTIMEDIA ANALYSIS EXPERIMENTAL STUDY

CONSTRUCTION OF THE FORMAL KNOWLEDGE – ADAPTATION OF THE CONNECTIONIST ARCHITECTURE

The proposed architecture has been evaluated in an image analysis application, involving classification of segments in images of summer holidays. Such images typically include persons swimming or playing sports in the beach or visiting places with buildings or trees; in this framework the selected concepts of interest in our experiments are the following: *Sea*, *Sand*, *Sky*, *Tree* and *Building*.

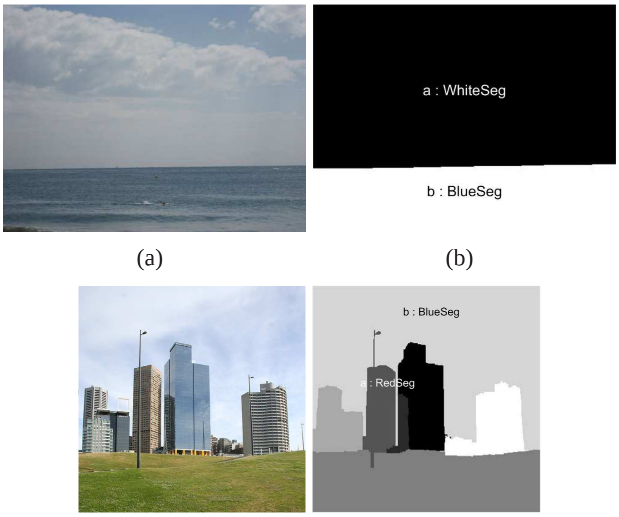
Following the described region-based segmentation procedure, we let each individual in our knowledge base correspond to an image segment. 45 low level features are used to characterize each individual, derived from the MPEG-7 Color Structure, Scalable Color and Homogeneous Texture Descriptors together with the Dominant Color of each segment. To illustrate the performance of the proposed neuro-symbolic architecture, we created a segment base composed of 3000 segments extracted from about 500 images of the Acemedia and K-Space datasets. This extraction was made through the described semantic R-SST segmentation algorithm, which also produced the spatial relations between each segment and its neighboring ones. Therefore, the resulting alphabet of our KB is composed of the set of concepts **C** = $\{$*WhiteSeg*, *BlueSeg*, *GreenSeg*, *RedSeg*, *YellowSeg*, *BrownSeg*, *GreySeg*, *BlackSeg*$\}$, the set of roles **R** = $\{$*aboveOf*, *belowOf*, *leftOf*, *rightOf*$\}$, while the set of individuals **I** consists of the segments of the images. The Terminological Box (TBox – Formal Knowledge) was then created based on the above mentioned concepts; this is described in Table 2 while the assertional component (ABox) of the KB is of the form:

$$\langle image1\_seg01 : BlueSeg \geq 0.65\rangle,$$
$$\langle image1\_seg01 : GraySeg \geq 0.35\rangle,$$
$$\langle image1\_seg02 : GreenSeg \geq 0.75\rangle,$$
$$\ldots$$
$$\langle (image1\_seg01, image1\_seg02) : above\text{-}of \geq 1\rangle,$$
$$\langle (image1\_seg01, image1\_seg03) : left\text{-}of \geq 1\rangle,$$
$$\ldots$$

In addition to the above assertions that are extracted by the semantic R-SST algorithm, each segment was also annotated with respect to the concepts of interest by the user, permitting the evaluation of classification.

After that, we selected 1500 image segments, which the Fuzzy Reasoning Engine FiRE classifies correctly according to the axioms defined in the initial TBox. This process forms new assertions in the assertional component of the KB (ABox) while the remaining 1500 segments, where errors occur, form the testing data. The following step has been to transfer the above knowledge to the connectionist – kernel based - architecture using this ABox. To accomplish this, we compute the kernel function κ_i, for every two segments combination w. r. t. (see Eqs. 1,2) the concepts of interest. Then, we compute the k_p^F for every two-segment combination, thus defining a kernel matrix (1500 × 1500), each row of which indicates the similarity of segment with all other ones. Some of the images and segments used for this process are illustrated in Fig 3. As we can observe the dominant colour and the colour of the neighboring segments in these examples are as defined in our knowledge base. For example in Fig. 3c the building in the middle has colour red (*i.e.*, *RedSeg*) and it is below of *BlueSeg*, therefore it is classified by the KB as *BuildingSeg*.

Segment classification was then accomplished through the use of a Support Vector Machine. At first we verified that the SVM was able to classify correctly the segments (based on the above kernel matrix). Having verified this, we assume that the SVM has been delivered to a user who wants to test its performance to the rest 1500 segments, which the knowledge base has not classified correctly. Since this SVM is the only tool that the user possesses (no knowledge base or reasoning engine), and one has to apply it to the new image segments, which are characterized by their low – level MPEG-7 features, it is assumed that the SVM package includes – apart from the trained SVM and its kernel matrix – a file with the low-level MPEG-7 features of each one of the 1500 training segments.



(a) (b)

(c) (d)

Fig. 3. *Some of the images with their segments that were classified correctly by FiRE according to the initial TBox and used for the kernel evaluation process.*

For each test image segment, obtained at the users environment, the semantic interpretation layer of the approach automatically extracts the low level MPEG-7 features. It then uses a matching algorithm to select the training data segment (individual) whose corresponding low-level MPEG-7 features are closest to those of the testing one. This is done by comparing the Euclidean correlation distance measure between the testing feature set and each feature set of the training segments. Assuming that if two image segments are the most similar w. r. t. their low level features, then this will hold for their high level characteristics as well, the layer uses the characteristics of the selected training segment (*i.e.*, the concept it is an instance of and the relationships it participates) for the testing image segment as well. The role of the semantic interpretation layer stops here. The characteristics of the test image segment are then inserted in the kernel Eqs. 1-2, which is Positive Definite (Fanizzi *et al.*, 2008), so that the SVM

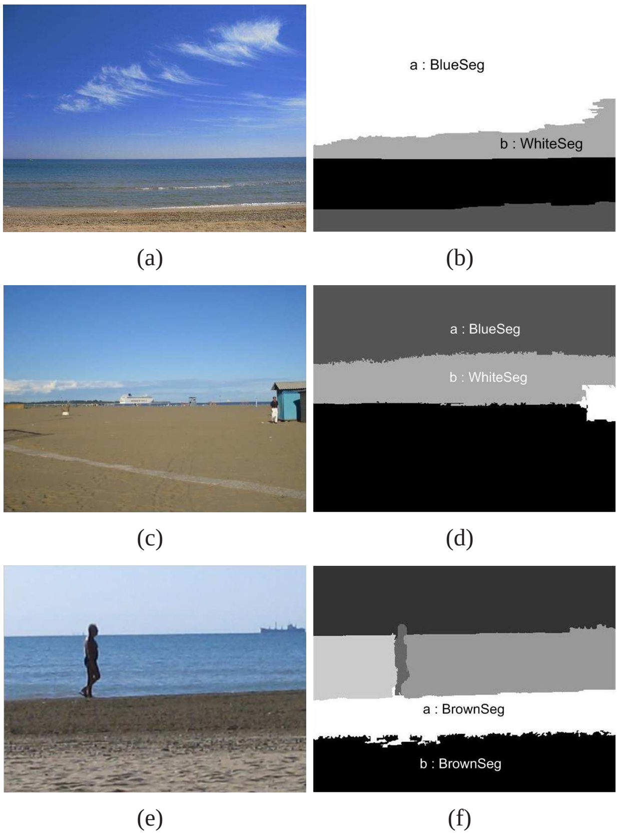


Fig. 4. *Some of the images and their segments that were classified correctly by the SVM while not by the initial KB.*

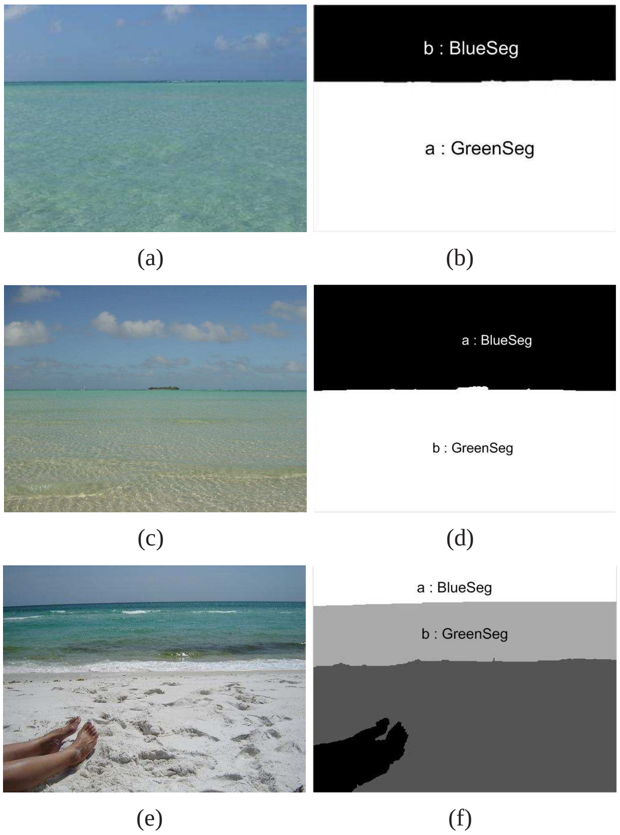


Fig. 5. *An image and its segments, annotated by the user and used to adapt the SVM (a,b); Similar images, correctly classified by the adapted SVM (c–f).*

provides the respective classification. Whether this has been a successful choice or not, will be evaluated by the user, and, if it is not, it may be used to adapt/retrain the classifier based on the possible users feedback.

Based on the above, we computed the correlation distance between each new input sample and the 1500 training segments and classified each input segment by the SVM. This resulted in correct classification of 510 out of the 1500 segments, which the original knowledge base and reasoning had not classified correctly. The following figure (Fig. 4) illustrates some segments that had not been initially classified correctly. For example, in Fig. 4b,d, sky segments include white colour and therefore were classified incorrectly by the KB because of their neighbors (see Table 2). Similarly in Fig. 4f, a sand segment with brown colour is below of another sand segment with the same colour, fact that causes mistaken classification by the KB. All these misclassified segments by the KB were corrected by the SVM.

Let us now assume that the user informs the system that some segment (of the rest 990) has been wrongly classified, providing also its correct category. After that, adaptation of the SVM kernel matrix can be performed by enhancing correlations of the segment with training segments of the correct category, while reducing (by half) its correlation value with segments of the misleading categories. With this procedure the SVM was able to classify correctly the specific segment and all other segments that were similar to them. Such an example is illustrated in Fig. 5. The user corrected the wrong classification of the KB and the SVM in Fig. 5b in which the sea segment was wrongly classified because of its green colour. This correction was used for the adaptation of the SVM and after that SVM succeeded in the classification of sea segments Fig. 5c,e that are also green. It should, however, be mentioned that the resulting adaptation is valid for specific contexts – image segments in our application – and should be tagged as such, if it is to be preserved in the system knowledge base.

ADAPTATION OF THE KNOWLEDGE BASE

The next step, after the enrichment and the possible adaptation of the SVM according to user corrected data, was to transfer the acquired knowledge

Table 2. *The initial knowledge for the specific context.*

$$
\begin{aligned}
\mathit{SandSeg} &\sqsubseteq (\mathit{BrownSeg} \sqcup \mathit{GraySeg}) \sqcap (\exists \mathit{belowOf}.(\mathit{BlueSeg} \sqcup \mathit{BrownSeg}) \\
&\quad \sqcup \exists \mathit{leftOf}.(\mathit{BlueSeg} \sqcup \mathit{BrownSeg}) \sqcup \exists \mathit{rightOf}.(\mathit{BlueSeg} \sqcup \mathit{BrownSeg})) \\
\mathit{BuildingSeg} &\sqsubseteq (\mathit{RedSeg} \sqcup \mathit{YelowSeg} \sqcup \mathit{BrownSeg} \sqcup \mathit{GraySeg}) \sqcap (\exists \mathit{belowOf}.\mathit{BlueSeg} \sqcup \exists \mathit{leftOf}.\mathit{BlueSeg} \\
&\quad \sqcup \exists \mathit{rightOf}.\mathit{BlueSeg}) \\
\mathit{SeaSeg} &\sqsubseteq (\mathit{BlueSeg} \sqcup \mathit{WhiteSeg}) \sqcap (\exists \mathit{belowOf}.\mathit{BlueSeg} \sqcup \exists \mathit{leftOf}.(\mathit{BlueSeg} \sqcup \mathit{BrownSeg}) \\
&\quad \sqcup \exists \mathit{rightOf}.(\mathit{BlueSeg} \sqcup \mathit{BrownSeg}) \sqcup \exists \mathit{aboveOf}.\mathit{BrownSeg} \\
\mathit{SkySeg} &\sqsubseteq (\mathit{BlueSeg} \sqcup \mathit{WhiteSeg} \sqcup \mathit{GraySeg}) \sqcap (\exists \mathit{aboveOf}.(\mathit{BlueSeg} \sqcup \mathit{GraySeg} \sqcup \mathit{RedSeg} \sqcup \mathit{YellowSeg} \\
&\quad \sqcup \mathit{GreenSeg}) \sqcup \exists \mathit{leftOf}.(\mathit{BlueSeg} \sqcup \mathit{GraySeg} \sqcup \mathit{RedSeg} \sqcup \mathit{YellowSeg} \sqcup \mathit{GreenSeg}) \\
&\quad \sqcup \exists \mathit{rightOf}.(\mathit{BlueSeg} \sqcup \mathit{GraySeg} \sqcup \mathit{RedSeg} \sqcup \mathit{YellowSeg} \sqcup \mathit{GreenSeg})) \\
\mathit{TreeSeg} &\sqsubseteq \mathit{GreenSeg} \sqcap (\exists \mathit{belowOf}.\mathit{BlueSeg} \sqcup \exists \mathit{leftOf}.\mathit{BlueSeg} \sqcup \exists \mathit{rightOf}.\mathit{BlueSeg})
\end{aligned}
$$

to the Formal Knowledge – TBox. The adapted SVM produces a new ABox consisting of the initial assertions extracted by the semantic interpretation layer (*i.e.*, colours and spatial relations) together with the classification of each segment to the concepts of interest (*i.e.*, *SeaSeg*, *SkySeg* etc), as enriched by the SVM.

In the original knowledge base every concept of interest was defined according to a disjunction of some colour concepts, considered as concept of major importance, together with concepts that specify the colours of neighboring regions, considered as the concepts for adaptation. For example, the concept *SeaSeg* was defined by the axiom $\mathit{SeaSeg} \sqsubseteq (\mathit{BlueSeg} \sqcup \mathit{WhiteSeg}) \sqcap (\exists \mathit{belowOf}.\mathit{BlueSeg} \sqcup \exists \mathit{leftOf}.\mathit{BlueSeg} \sqcup \exists \mathit{rightOf}.\mathit{BlueSeg})$. Assuming *SeaSeg* as F_1 (see Eq. 4), then the concept of major importance that will remain unchanged from the adaptation process is concept $\mathit{BlueSeg} \sqcup \mathit{WhiteSeg}$ while the concept formed by the disjunctions of the neighboring criteria colours *i.e.*, $\exists \mathit{belowOf}.\mathit{BlueSeg} \sqcup \ldots \sqcup \exists \mathit{rightOf}.\mathit{BlueSeg}$, may be adapted. Therefore, in our approach, the set of concepts that are examined for adaptation consists of the concepts formed using the full existential operator with each one of the 4 roles, having, as role-filler, each of the colour concepts (*i.e.*, $\exists \mathit{belowOf}.\mathit{BlueSeg}$, $\exists \mathit{belowOf}.\mathit{WhiteSeg}$, $\ldots$, $\exists \mathit{rightOf}.\mathit{BlueSeg}$, $\exists \mathit{rightOf}.\mathit{WhiteSeg}$, $\ldots$).

All the segments that are classified by the adapted SVM result in an ABox also consisting of the initial assertions and the assertions added after classification. After the classification of an individual as *SeaSeg*, the assertions in *ABox* are examined. All concepts that can be adapted are examined and indexed for each individual. If the occurrence of one of these concepts exceeds a threshold, that is defined according to the total number of regions, and this concept is not used in the axiom, then it is proposed to the knowledge engineer for incorporation in the relevant axiom (*i.e.*, *SeaSeg*). The incorporation of the concept is made with the use of the appropriate DL constructor selected by the knowledge engineer. On the other hand, if one of these concepts has very few occurrences and this concept is used in the relevant axiom, then it is suggested for removal.

To understand the reason why selection of representative segments is necessary for the adaptation of a knowledge base let us examine the following example illustrated in Fig. 6. The images on the left are the original images, while on the right the segmented images are illustrated together with the desired targets as assigned by the annotator. If we first examine the images of first row (*i.e.*, images a,b) we observe that the color of the segments agrees with the concepts defined in our knowledge base (see Table 2). In other words the colour of segment a is Gray, White, so the knowledge base correctly classifies it as *SkySeg* while the segment b that is blue is classified as *SeaSeg*. On the other hand examining the colours of the segments in the original image of the second row (c), according to which the concepts of interest are defined, we can observe that the colour of segment c is brown. This means that the defined axiom in the knowledge base for concept *SeaSeg* (Table 2) will mistakenly classify the specific segment as *SandSeg*. On the other hand, clearly this segment is not characteristic of concept *SeaSeg* and therefore adaptation of the knowledge base according to such a special case would detune its performance.

Using this technique, the relative concepts that play a significant role for a concept of interest, according to the new ABox formed by the connectionist model, are included in its definition.

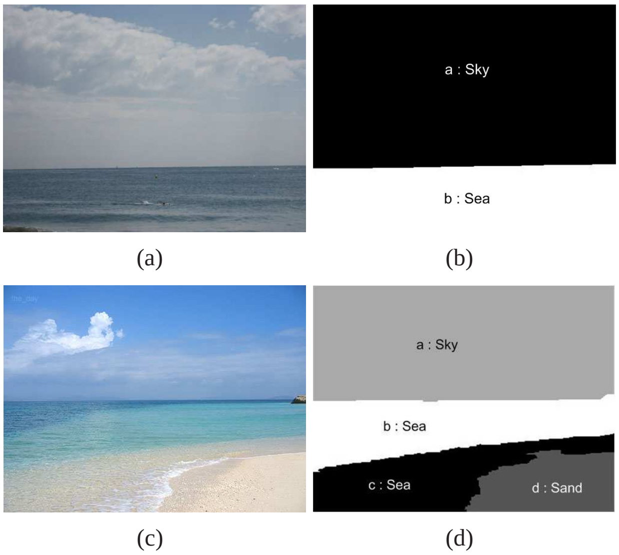


Fig. 6. *KB adaptation for specific contexts or circumstances.*

Table 3 illustrates the adapted knowledge base. The axioms were corrected by the adaptation process and in some cases differ form the original axioms defined. For example *TreeSeg* axiom was changed and in its new description the neighboring segments are of colour green while the possibility of brown neighbors in *SeaSeg* was omitted. Additionally, the white and brown neighbors were added as possible neighbors of *SkySeg* and *SandSeg* respectively as described in the previous section.

RESULTS

The performance of the proposed approach has been compared with standard approaches for classification of image regions (Athanasiadis *et al.*, 2009). These approaches have similar characteristics to our region-based method, since their purpose is to classify a region produced by a bottom-up segmentation algorithm to a predefined set of categories. The first approach introduces an individual Support Vector Machine (SVM) for every defined concept, to detect the corresponding instances. Every SVM is trained under the "one-against-all" approach using the same training set (composed of the same input features) which was also used for training in our approach.

More specifically, an individual SVM is introduced for every defined concept, to detect the corresponding instances. Every SVM is trained under the "*one-against-all*" approach. The region feature vector, consisting of the MPEG-7 visual descriptors, constitutes the input to each SVM, which returns for every image segment a numerical value in the range $[0,1]$. This value denotes the degree of confidence, to which the corresponding region is assigned to the concept associated with the particular SVM. For each region, the maximum of the K calculated degrees of confidence, $argmax(\mu_a(C_k))$, indicates its concept assignment, whereas the pairs of all supported concepts and their respective degree of confidence μ computed for segment α comprise the region's concept hypothesis.

The second approach is based on Particle Swarm Optimization (PSO) (Chandramouli and Izquierdo, 2006), which is based on optimization of the results of a Self Organizing Map classifier. Again in this case the same training set was used and an individual SOM network trained using the "*one-against-all*" approach is employed for each category. In the basic training algorithm the prototype vectors are trained according to $w_d(t+1) = w_d(t) + g_d(t)[x - w_d(t)]$ where w_d denotes the weight of the neurons in the SOM network, $g_d(t)$ is the neighborhood function and d is the dimension of the input feature vector.

Table 4 illustrates the performance of the above two algorithms and of the proposed in this paper approach, when classifying the same 1320 image segments in five concepts (sand, sea, sky, tree, building). Two columns of results are provided in the case of the proposed approach. The third column shows the performance of the SVM when its kernel is designed through Eqs. 1-2 by transferring the initial high level knowledge. The fourth column shows the resulting SVM performance after adaptation using the presented method.

As we can notice, in Table 4, the proposed methods outperform the other two by a large margin, using the standard precision and recall measures for the evaluation. This has been achieved, because the concepts can be represented in much detail (slightly varying from concept to concept) by an axiom; regions also have, in most cases, specific surroundings, thus, defining respective spatial relations. In this way, the interweaving of the formal knowledge and the connectionist system, enriches the latter with more information, that varies with the concept examined, and consequently manages to provide much better results, when compared to conventional machine learning techniques.

CONCLUSION

In this paper we presented a novel architecture for image analysis applications based on connectionist adaptation of ontological knowledge. The proposed system architecture, consists of two main components:

Table 3. *The adapted knowledge for the specific domain.*

$$\begin{aligned}
\mathit{SandSeg} &\sqsubseteq (\mathit{BrownSeg} \sqcup \mathit{GraySeg}) \sqcap (\exists \mathit{belowOf}.(\mathit{BlueSeg} \sqcup \mathit{BrownSeg}) \sqcup \exists \mathit{leftOf}.(\mathit{BlueSeg} \sqcup \mathit{BrownSeg}) \\
&\quad \sqcup \exists \mathit{rightOf}.(\mathit{BlueSeg} \sqcup \mathit{BrownSeg}) \\
\mathit{BuildingSeg} &\sqsubseteq (\mathit{RedSeg} \sqcup \mathit{YelowSeg} \sqcup \mathit{BrownSeg} \sqcup \mathit{GraySeg}) \\
\mathit{SeaSeg} &\sqsubseteq (\mathit{BlueSeg} \sqcup \mathit{WhiteSeg}) \sqcap (\exists \mathit{belowOf}.\mathit{BlueSeg} \sqcup \exists \mathit{leftOf}.\mathit{BlueSeg} \sqcup \exists \mathit{rightOf}.\mathit{BlueSeg}) \\
\mathit{SkySeg} &\sqsubseteq (\mathit{BlueSeg} \sqcup \mathit{WhiteSeg} \sqcup \mathit{GraySeg}) \sqcap (\exists \mathit{aboveOf}.(\mathit{BlueSeg} \sqcup \mathit{GreenSeg} \sqcup \mathit{RedSeg} \sqcup \mathit{GraySeg} \\
&\quad \sqcup \mathit{GreenSeg} \sqcup \mathit{WhiteSeg}) \sqcup \exists \mathit{leftOf}.(\mathit{WhiteSeg} \sqcup \mathit{GreenSeg} \sqcup \mathit{BlackSeg}) \\
&\quad \sqcup \exists \mathit{rightOf}.(\mathit{WhiteSeg} \sqcup \mathit{GreenSeg} \sqcup \mathit{BlackSeg})) \\
\mathit{TreeSeg} &\sqsubseteq \mathit{GreenSeg} \sqcap (\exists \mathit{belowOf}.\mathit{GreenSeg} \sqcup \exists \mathit{leftOf}.\mathit{GreenSeg} \sqcup \exists \mathit{rightOf}.\mathit{GreenSeg})
\end{aligned}$$

Table 4. *Comparison of Connectionist System (CS) performance (before and after adaptation) with two competitive methods: SVM classifier and Particle Swarm Optimization.*

Label	SVM		PSO		Initial CS/KB		Adapted CS/KB	
	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
Sand	50.62 %	35.48%	30.45%	63.19%	75.12%	51.72%	83.17%	72.13%
Building	47.43 %	44.65%	35.41%	48.59%	52.82%	31.43%	58.73%	37.61%
Sea	46.22 %	62.22%	18.81%	65.78%	68.15%	75.94%	88.74%	79.24%
Sky	66.83 %	50.7%	64.87%	64.66%	64.74%	50.1%	75.32%	64.92%
Tree	44.94 %	40%	12.07%	57 %	58.34%	51.65%	65.84%	60.23%
Total	51.21 %	46.61%	32.32%	59.84 %	63.83%	52.17%	74.36%	62.83%

the *Formal Knowledge* and the *Knowledge Adaptation*. The *Formal Knowledge* stores, the terminology and assertions that describe the problem under analysis. On the other hand, the *Knowledge Adaptation* converts the formal knowledge to a connectionist system that adapts and reconverts to formal knowledge, offering in that way an improved adapted knowledge.

The proposed architecture was applied to an image classification and segmentation problem, presenting very promising results. A comparison with other state of the art approaches on the same problem validated the very good performance of our proposal. Future work, includes the incorporation of fuzzy set theory in the kernel evaluation. Additionally, we intend to further examine the adaptation of a knowledge base using the connectionist architecture, mainly focusing on an improved semi-automatic selection of the appropriate DL constructors.

REFERENCES

Adamek T, O'Connor N, Murphy N (2005). Region-based segmentation of images using syntactic visual features. In: Proc 6th Int Worksh Image Anal Multimedia Interact Serv (WIAMIS). Montreux, Switzerland.

Athanasiadis T, Mylonas P, Avrithis Y, Kollias S (2007). Semantic image segmentation and object labeling. IEEE T Circ Syst Vid 17:298–312.

Athanasiadis T, Simou N, Papadopoulos G, *et al.* (2009). Integrating image segmentation and classification for fuzzy knowledge-based multimedia indexing. In: Huet B, Smeaton AF, Mayer-Patel K, Avrithis Y, eds., Proc 15th Int Multimedia Model Conf (MMM). Sophia-Antipolis, France.

Baader F, McGuinness D, Nardi D, Patel-Schneider P (2002). The Description Logic Handbook: Theory, implementation and applications. Cambridge: Cambridge University Press.

Berretti S, Bimbo AD, Vicario E (2001). Efficient matching and indexing of graph models in content-based retrieval. IEEE T Circ Syst Vid 11:1089–105.

Bloehdorn S, Sure Y (2007). Kernel methods for mining instance data in ontologies. In: Gil Y, Motta E, Benjamins V, Musen MA, eds., Proc 6th Int Semantic Web Conf (ISWC). Galway, Ireland.

Borenstein E, Sharon E, Ullman S (2004). Combining top-down and bottom-up segmentation. In: Dougherty E, ed., Proceedings of Computer Vision and Pattern Recognition Workshop (CVPRW). Washington DC, USA.

Carson C, Belongie S, Greenspan H, Malik J (2002). Blobworld: Image segmentation using expectation-maximization and its application to image querying. IEEE T Pattern Anal 24:1026–38.

Chandramouli K, Izquierdo E (2006). Image classification using self organising feature maps and particle swarm optimisation. In: Proc 7th Worksh Image Anal Multimedia Interact Serv (WIAMIS). Seoul, Korea.

Christel MG, Hauptmann AG (2005). The use and utility of high-level semantic features in video retrieval. In: Leow WK, Lew MS, Chua T, Ma W, Chaisorn L, Bakker EM, eds., Proc 4th Int Conf Image Video Retrieval (CIVR). Singapore, Singapore.

Comaniciu D, Meer P (2002). Mean shift: A robust approach toward feature space analysis. IEEE T Pattern Anal 24:603–19.

Fanizzi N, d Amato C, Esposito F (2007). Randomised metric induction and evolutionary conceptual clustering for semantic knowledge bases. In: Proc 16th Conf Inform Knowledge Managem (CIKM). Lisboa, Portugal: ACM.

Fanizzi N, d Amato C, Esposito F (2008). Statistical learning for inductive query answering on owl ontologies. In: Sheth A, Staab S, Dean M, Paolucci M, Maynard D, Finin T, Thirunarayan K, eds., Proc 12th Int Semantic Web Conf (ISWC). Karlsruhe, Germany.

Felzenszwalb PF, Huttenlocher DP (2004). Efficient graph-based image segmentation. Int J Comput Vision 59:167–81.

Garcez ASA, Broda K, Gabbay DM (2001). Symbolic knowledge extraction from trained neural networks: A sound approach. Artif Intell 125:155–207.

Garcez ASA, Zaverucha G (1999). The connectionist inductive learning and logic programming system. Appl Intell 11:59–77.

Hammer B, Hitzler P (2007). Perspectives of Neural-Symbolic Integration. Studies in Computational Intelligence, vol. 77. Springer.

Hitzler P, Holldobler S, Seda A (2004). Logic programs and connectionist networks. J Appl Log 2:245–72.

Horrocks I, Sattler U, Tobies S (2000). Reasoning with individuals for the description logic $\mathscr{SHIQ}$. In: MacAllester D, ed., Proc 17th Conf Autom Deduct (CADE), no. 1831 in LNAI. Springer-Verlag.

Klir GJ, Yuan B (1995). Fuzzy Sets and Fuzzy Logic: Theory and Applications. Prentice-Hall.

Kolman E, Margaliot M (2005). Are artificial neural networks white boxes? IEEE T Neural Networ 16:844–52.

Lee S, Crawford M (2001). Unsupervised classification using spatial region growing segmentation and fuzzy training. In: Proc Int Conf Image Proc (ICIP), vol. 1. Thessaloniki, Greece.

Luo J, Savakis A (2001). Indoor vs outdoor classification of consumer photographs usinglow-level and semantic features. In: Proc Int Conf Image Proc (ICIP), vol. 2. Thessaloniki, Greece.

Meyer F, Beucher S (1990). Morphological segmentation. J Vis Commun Image R 1:21–46.

Millet C, Bloch I, Hede P, Moellic P (2002). Using relative spatial relationships to improve individual region recognition. In: Proc 2nd Eur Worksh Integr Knowledge Semantic Digit Media Technol.

Morris OJ, Lee MJ, Constantinides AG (1986). Graph theory for image analysis: An approach based on the shortest spanning tree. IEE Proc F 133:146–52.

Naphade M, Huang TS (2001). A probabilistic framework for semantic video indexing, filtering and retrieval. IEEE Multimedia 3:144–51.

Papadopoulos GT, Mezaris V, Kompatsiaris I, Strintzis MG (2007). Combining global and local information for knowledge-assisted image analysis and classification. J Adv Signal Procces 2007:18–33.

Papamarkos N, Strouthopoulos C, Andreadis I (2000). Multithresholding of color and gray-level images through a neural network technique. Image Vision Comput 18:213–22.

Pinkas G (1991). Propositional non-monotonic reasoning and inconsistency in symmetric neural networks. In: Proc 12th Int Joint Conf Artif Intel (IJCAI). San Francisco, CA, USA: Morgan Kaufmann.

Spyrou E, Tolias G, Avrithis PMY (2009). Concept detection and keyframe extraction using a visual thesaurus. Multimed Tools Appl 41:337–73.

Stamou G, Kollias S (2005). Multimedia Content and the Semantic Web: Standards, Methods and Tools. John Wiley & Sons.

Stoilos G, Simou N, Stamou G, Kollias S (2006). Uncertainty and the semantic web. IEEE Intell Syst 21:84–7.

Stoilos G, Stamou G, Pan JZ, Tzouvaras V, Horrocks I (2007). Reasoning with very expressive fuzzy description logics. J Artif Intell Res 30:273–320.

Straccia U (2001). Reasoning within fuzzy description logics. J Artif Intell Res 14:137–66.

Yamada A, Pickering M, Jeannin S, Cieplinski L, Ohm J, Kim M (2001). MPEG-7 Visual part of eXperimentation Model version 9.0. Tech. rep.

Zhang L, Lin F, Zhang B (2001). Support vector machine learning for image retrieval. In: Proc Int Conf Image Proc (ICIP), vol. 2. Thessaloniki , Greece.