

Automatic image annotation refinement using fuzzy inference algorithms

Marina Ivašić-Kos¹ Miran Pobar¹ Slobodan Ribarić²

¹Department of Informatics, University of Rijeka, 51000 Rijeka, Croatia

²Faculty of Electrical Engineering and Computing, University of Zagreb, 10000 Zagreb, Croatia

Abstract

Facilitating tasks such as image search is one of the goals of image annotation methods that automatically assign keywords to images. In order to achieve as accurate annotation on object level as possible, and to reduce negative influence of misclassified objects on the inference of scenes, a knowledge based refinement of object classification results is proposed. A fuzzy knowledge representation scheme called KRFPN_r is used to represent knowledge about objects and their relations. Fuzzy inheritance and intersection algorithms are used to refine the image annotation on object level by discarding or replacing object labels that do not fit the context.

Keywords: automatic image annotation, annotation refinement, fuzzy knowledge representation scheme, fuzzy inference algorithms

1. Introduction

Automatic image annotation methods deal with visual features that can be extracted from the raw image data, such as color, texture, structure, etc. and can automatically assign keywords from a predefined vocabulary to an unlabeled image. In other words, the goal is to bridge the so-called semantic gap [1] between the available features and the keywords that could be useful e.g. for text based image retrieval.

When searching for images, keywords corresponding to objects or scenes are typically used. Therefore, automatic image annotation methods use a vocabulary containing keywords that correspond to objects like *building*, *bear*, *trees*, *lion*, *train*, etc. or to the whole scene like *forest*, *underwater* or more general *nature*.

Scenes may be very complex and with many different objects present in the image, so many object labels could be assigned to an image. The recognition of an object depends on the extracted features and the efficiency of a classifier. If it is assumed that the composition of objects makes the scene, as in [2], then scene labels can be inferred from object labels. Some objects, e.g. background objects like *sky*, *grass*, etc., may belong to a number of different scenes and some objects are characteristic to only one scene, e.g. *train* for scene *railway*. If there is an object that is highly discriminant for some scene, it is crucial to detect that object. On the other hand, for scenes without a typical object (e.g. *Mountain*) even background objects could play an important role for image interpretation. Hence, image annotation on object level should be as precise as possible so that misclassi-

fied objects do not negatively influence the image interpretation. The results of automatic image annotation strongly depend on the results of object classification, and image annotation will be more accurate as objects in the image or scene are better classified. However, refinement of classification results based on knowledge can also improve the accuracy of annotation.

Most automatic image annotation approaches proposed so far belong to the field of machine learning. The annotation task is closely related to the classification problem or problem of representing correlations between images and keywords. Methods based on classification like [3] classify images or image segments into a number of predefined classes based on low-level image features. Probabilistic methods learn a relevance model to represent correlations between images and keywords e.g. based on translation model [4] or using latent semantic analysis [5]. A recent survey of research made in that field can be found in [6, 7].

Lately graph-based image analysis algorithms have been investigated for refinement of classification results. In several approaches proposed so far [8-14], correlation between annotated keywords has been used. In [14], WordNet based semantic similarity is used to detect and discard irrelevant keywords among candidate annotated keywords. A graphical model for fusing visual content, represented by a nearest spanning chain, and keyword correlation by WordNet is used in [11]. Image annotation refinement by re-ranking the annotations using the random walk with restarts algorithm is proposed in Wang et al. [9]. The same group of authors has in [10] proposed a content based image annotation refinement using Markov chains. In [8] conditional random fields (CRF) and normalized Google distances (NGD) [14] are used for image annotation refinement.

In this paper we propose a fuzzy knowledge based refinement of image annotation, Fig. 1. An image is assumed to be labeled with object labels that are a result of classification of image segments. The object labels are input into the annotation refinement, which uses inference algorithms and a knowledge base with facts about objects and their relations for consistency checking and finding candidate replacement labels. In the consistency checking phase a possible "intruder" label, i.e. label that doesn't match the context can be detected. In the replacement phase, the intruder label is either discarded or replaced with a label of an object that has similar properties but better fits the context. The refined labels are used as annotation and for inferring scene labels.

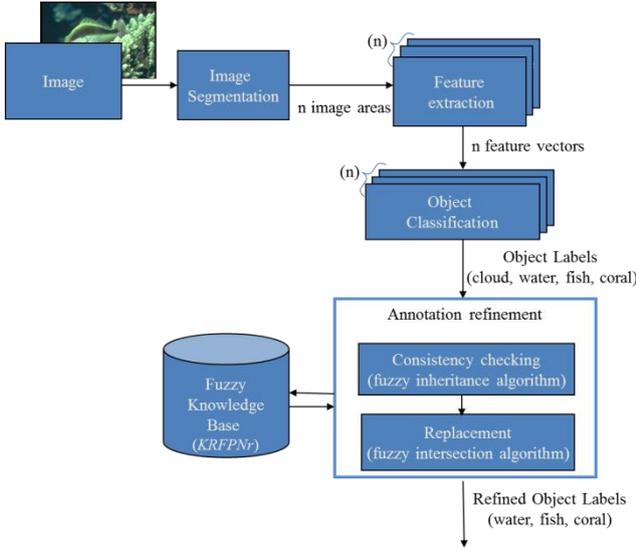


Figure 1. Diagram of image annotation with fuzzy knowledge based annotation refinement

The paper is organized as follows: Section 2 presents the fuzzy knowledge-representation formalism used to model the facts about objects and their relationships. Sections 3 and 4 detail the consistency checking and replacement phases with descriptions of the fuzzy inference algorithms. A conclusion is given in Section 5.

2. A Fuzzy Knowledge-representation Formalism used for Annotation Refinement

The assumption is that scene inference depends on objects that are obtained as classification results, so it is useful to detect and discard or replace those object labels that do not fit the context. Considering that automatic object classification is subject to errors and that knowledge about concepts is often incomplete, the ability to make conclusions from imprecise, fuzzy knowledge becomes necessary. A knowledge-representation scheme based on Fuzzy Petri net [15] is adapted for this purpose, referred to as *KRFPNr*. Fuzzy pseudo-spatial, spatial and attribute relations between objects are defined based on data set and included in knowledge base.

The *KRFPNr* scheme is defined as 12-tuple: $KRFPNr = (P, T, I, O, M, \Omega, f, c, \alpha, \beta, D, \Sigma)$, (1) elements of which are described below.

$P = \{p_1, p_2, \dots, p_n\}, n \in \mathbb{N}$ is a set of places and $T = \{t_1, t_2, \dots, t_m\}, m \in \mathbb{N}$ is a set of transitions.

The link between places and transitions is given with the input and output functions, $I: T \rightarrow \mathcal{P}(P) \setminus \emptyset$ and $O: T \rightarrow \mathcal{P}(P) \setminus \emptyset$, so that for each transition there are at least one input and output place.

A place may be marked with one or more tokens from the set $M = \{m_1, m_2, \dots, m_l\}, l \geq 0$. The tokens are used to define the execution of a Fuzzy Petri Net (FPN) in discrete steps. The execution of the *KRFPNr* is important for automatic reasoning using the fuzzy inference algorithms. In each execution step token distribution and token values are determined. Token distribution within places is given as $\Omega_w(p) \in \mathcal{P}(M)$ for an execu-

tion step $w=0, 1, \dots$, where $\mathcal{P}(M)$ is a power set of M . In our case, in the initial marking each place can have at most one token. A place p that contains one or more tokens in step w is called a marked place and $\Omega_w(p) \neq \emptyset$. When every input place $\forall p_i \in I(t_j)$ of a transition t_j is marked, transition t_j is enabled and can fire if is selected. By firing of the transition t_j , tokens simultaneously move from all the transition's input places $p_i \in I(t_j)$ to the corresponding output places $p_l \in O(t_j)$.

To each place from the set P , a concept d from the set D is assigned by the bijective function $\alpha: P \rightarrow D$. The concepts from the set D may be object and scene labels used for image annotation, e.g. *sky, whale, lion* or properties of objects like their color (*blue, orange, ...*), position (*middle, left, ...*), etc.

To each transition from the set T , a relationship r from the set Σ is assigned by the function $\beta: T \rightarrow \Sigma$. The set Σ contains all the relationships defined between concepts in the scheme, e.g. *occurs_with, is_near, consists_of*.

The uncertainty and confidence related to the concepts and the relationships between them are expressed by the values of the association functions $f(t_i), t_i \in T$, and $c(m_i), m_i \in M$. The degree of truth of the relationship mapped to a transition is given by the transition value $f: T \rightarrow [0, 1]$. The degree of truth of the concept mapped to the marked place is represented by the token value $c: M \rightarrow [0, 1]$. The transition values and the initial token values are defined according to the used training dataset.

2.1. Modelling the reliability of relations

The transition value depends on the reliability of relationship that is associated to the transition. There are three types of relationships defined in the *KRFPNr* scheme: the pseudo-spatial, spatial and attribute relationships.

A relationship *occurs_with* $\in \Sigma$ belongs to pseudo-spatial relationships and describes a mutual occurrence of two objects in the same scene. The reliability of relation *occurs_with* between objects $d_i, d_j \in D$, denoted as d_j *occurs_with* d_i , means the reliability that the object d_i appears on the scene along with the object d_j . The reliability f_r of the relationship *occurs_with*, between objects d_j and d_i is computed using data in the training set as:

$$f_r(d_i \text{ occurs_with } d_j) = P(d_i | d_j) = \frac{P(d_i \cap d_j)}{P(d_j)} \cong \frac{f_{ij} + mp_j}{f_j + m}, \quad i \neq j. \quad (2)$$

The probabilities $P(d_j \cap d_i)$ and $P(d_i)$ are estimated as the empirical relative frequency f_{ij} of mutual occurrence of both objects d_j and d_i on the scenes in the training set and empirical relative frequency f_j of occurrence of object d_j in the training set. To be able to obtain an estimate of reliability in case when there are no examples of mutual occurrence in the training data set, an m - estimate is implemented [16]. The parameter m is the size of a virtual set of samples and is experimentally set to 2, and p_j is the estimated probability that a sample is the object d_j .

The reliability of appearance of object d_j when object d_i exists on the scene is not equal to the reliability of appearance of object d_i when object d_j is on the scene. For example, the reliability of appearance of object *sky* is higher when object *airplane* exists on the scene than the appearance of object *airplane* if object *sky* is on the scene. That is because the *airplane* in most cases is in the *sky* but *sky* co-occurs with a large number of different objects like *trees*, *lion*, *train*, ...

The *occurs_with* relationship is used to check whether the results of segment classification are consistent. The value of transition assigned to relation *occurs_with* is set to the reliability of that relation:

$$\begin{aligned} f(t_j) &= f_r(d_j \text{ occurs_with } d_i), t_j : \\ \beta(t_j) &= \text{occurs_with}, d_i \in I(t_j), d_j \in O(t_j) \end{aligned} \quad (3)$$

Spatial relationships in our scheme specify a position of an object on the scene such as *at_the_top*, *at_the_bottom* or where an object is located on the scene with respect to a reference object, like *above*, *next_to*. Spatial relationships can be useful for consistency checking as well as for finding objects that are most similar to the reference object according to their spatial properties.

Other types of spatial relationships, such as topological, distance and internal relations [17], have not been used in this experiment but can easily be added to the scheme.

The reliability of a spatial relation is computed using empirical relative frequencies of objects and their spatial positions with m-estimate, similarly as reliability of *occurs_with* relations. The value of transition assigned to a spatial relation is set to the reliability of that relation.

An attribute relation associates an object with specific values of its attribute. These kinds of relationships are used to determine objects that are most similar to the reference object according to their perceptual properties.

3. Consistency-checking phase

It is to be expected that some of the object labels obtained after segment classification do not fit the context due to classification errors. The consistency of obtained labels is checked using the facts included in the knowledge base and the fuzzy-inheritance inference algorithm [15]. The fuzzy-inheritance algorithm generates the fuzzy-inheritance tree where the leaf nodes correspond to the properties of the analyzed object or to objects with which that object can occur on the same scene.

It is assumed that the class with the highest reliability defines the likely context, so those classes that have a lower confidence value are analyzed first. For the chosen concept $d_i \in A(e) \subset D$, the corresponding place $p_k \in P$ is found using the function $\alpha^{-1}: \alpha^{-1}(d_i) = p_k$. The initial distribution of tokens $\Omega_0(p_k) = \{m_1\}, \Omega_0(p_j) = \emptyset, j \neq k$ is defined and the token value $c(m_1)$ is set to 1. The root node π_0 of the inheritance tree is formed according to the initial distribution and the corresponding token values. The nodes of inheritance trees have the form $\pi(p_j, c(m_1))$ $j = 1, 2, \dots, p$,

$l = 1, 2, \dots, r, 0 \leq r \leq |M|$, where the first component specifies the place p_j where the token m_1 is located and the second one its value, $c(m_1)$. The steps of the fuzzy inheritance algorithm are:

Algorithm 1. Fuzzy inheritance algorithm adjusted for consistency checking.

Input:
 Root node π_0 of the k-level inheritance tree. Depth of inheritance k .
 Facts in the knowledge base represented with KRFPN

Output:
 k-level inheritance tree with leaf nodes denoted as terminal, k-terminal and frozen.

Steps:

1. Let $\pi_x = \pi_0$.
2. If there are no enabled transitions for π_x , π_x is a terminal node (T); continue from step 7.
3. If π_x lies at the k-th level of the inheritance tree, π_x is a k-terminal node (k-T); continue from step 7.
4. For all transitions t_j that are enabled for node π_x create a node π_y . The first component of the new node is the output place p_y of the transition t_j , $p_y \in O(t_j)$ and the corresponding token value is calculated as $c_j f(t_j)$ where c_j is the second component of π_x and $f(t_j)$ is the value of the transition t_j . Nodes π_x and π_y are linked by a directed arc t_j .
5. If π_y corresponds to properties of concept d_i , then it is declared as a frozen node (F).
6. If the node π_y is not classified as a frozen node, then it becomes a frontier node.
7. If there are frontier nodes, let $\pi_x = \text{next frontier node}$; continue from step 2.

If the analyzed object d_i does not have the *occurs_with* relation defined for the other objects d_j in the annotation set, $d_j \in A(e) \setminus \{d_i\}$, it is assumed that it does not correspond to the probable context and that it is the result of misclassification. The measure of truth of *occurs_with* relation between the root node and all other objects is determined by the token values in leaf nodes (terminal, k-terminal or frozen nodes).

Leaf nodes of the inheritance tree include all objects from the knowledge base that make the codomain of the relation *occurs_with* for a given object $d_i \in A(e)$.

If the intersection of the codomain of the relation *occurs_with* for a given object $d_i \in A(e)$ and the set $A(e)$ is also the set $A(e)$, it can be concluded that object d_i can appear together with all objects from $A(e)$. The more elements are in the intersection set, the greater is the reliability that the object d_i fits the context. In case when the intersection is empty or the intersection contains no objects with confidence value $f_r(d_j \text{ occurs_with } d_i) > 0.05, i \neq j, d_j \in A(e)$, it is implied that object $d_i \in A(e)$ most likely does not match the context of the image.

Objects that have passed the consistency checking become parts of the final annotation set $A'(e) = A(e) \setminus \{d_{i1}, \dots, d_{il}\}$ for image e and objects $\{d_{i1}, \dots, d_{il}\}$ that do not fit the context are considered intruders and are discarded or, if possible, replaced with concepts that have similar properties according to the facts in the knowledge base.

3.1. Example of consistency checking

An example of consistency checking is presented using the unlabeled image e in Fig. 2. The image is segmented and each segment is represented with a feature vector. Each segment is then classified into one of predefined object labels $d_i \in D$. It is likely that one object label will appear more than once because of segmentation or actual multiple appearance on the scene. From obtained object labels the annotation set is formed: $A(e) = \{water, fish, ground, coral\}$. The corresponding confidence values are 0.56, 0.3, 0.11, 0.25. Note that the label $ground$ is a result of misclassification because it is not present in the image.



Figure 2. Example of an original unlabeled image and its segmented image

Every obtained object label $d_i \in A(e) = \{water, fish, ground, coral\}$ is checked for inconsistency in order to verify whether there is an *occurs_with* relationship defined between that object and other obtained object labels. Objects are selected in descending order of their confidence value. In this case, the object $d_i = ground$ is selected first and checked with the rest of the labels in $A(e) \setminus \{ground\}$.

For the object $ground$, the appropriate place in the knowledge-representation scheme is determined by the function $\alpha^{-1}(ground) = p_{13}$, $ground \in D$, and a token m_1 is placed in the place p_{13} . According to the initially marked place, the initial token distribution Ω_0 is created, $\Omega_0 = (\emptyset, \emptyset, \dots, \{m_1\}, \dots, \emptyset)$. As explained, the corresponding root node of the inheritance tree is $\pi_0(p_{13}, \{1.0\})$. Figure 3 shows the inheritance tree formed by applying the fuzzy-inheritance algorithm for the object $ground$. The arcs of the inheritance tree are labeled with a transitions $t_j \in T$ and values $f(t_j)$. For example, the arc t_{168} corresponds to the *occurs_with* relation between the nodes π_0 and π_{11} . The node π_0 corresponds to the object $ground$ at the place p_{13} and the node π_{11} to the object sky at the place p_1 . The confidence of this relation is determined by the token value in a leaf node and is 0.35.

The inheritance tree has stopped on output nodes of the transitions, marked as frozen leaf nodes (F). Because the depth of the inheritance tree was 1, the leaf nodes are also marked as 1-terminal (1-T). Leaf nodes of inheritance tree include all objects from the knowledge base that are in relation *occurs_with* with object $ground$. The leaf nodes include places that function α maps to objects such as $\alpha(p_1) = sky, \alpha(p_4) = lion, \alpha(p_{10}) = grass, \dots, \alpha(p_{18}) = rock, \alpha(p_{25}) = water$ that make the codomain of relation *occurs_with* for a given object $ground$. The intersection of the codomain and the set $A(e)$ contains only object $\{water\}$ with low confidence value,

$f_r(water \text{ occurs_with } ground) = 0.04$ and for the remaining objects of set $A(e)$ the relation *occurs_with* is not defined. That is, the object $ground$ does not appears with objects coral and fish on the scene, so it can be concluded that the concept $ground$ most likely does not match the context of the image depicted in Fig. 2 and should be discarded or, if possible, replaced with a concept that has similar properties. Therefore, now the annotation set is: $A'(e) = A(e) \setminus \{ground\}$.

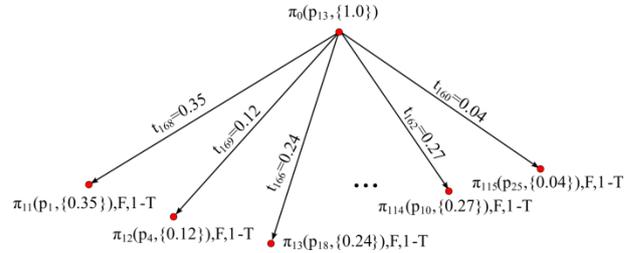


Figure 3. Inheritance tree for the object $ground$ detected as possible intruder in automatic annotation of image e

The inheritance tree is formed for the next object according to the confidence value in the set $A'(e) = \{water, fish, coral\}$, in this case for $coral$. For example, in Fig. 4 is the inheritance tree with object $coral$ as the root node and the leaf nodes that correspond to all objects from the knowledge base that are in relation *occurs_with* with object $coral$. After applying the function α on places at the leaf nodes,

$\alpha(p_{25}) = water, \alpha(p_{26}) = fish, \alpha(p_{17}) = sand, \dots, \alpha(p_{18}) = rock$ the codomain of relation *occurs_with* for a given object $coral$ is formed. The intersection of the codomain and the set $A'(e)$ is the set $A'(e)$, so it can be concluded that object $coral$ can appear together with objects from $A'(e)$ in the image. Since the intersection of of the codomain and the set $A'(e)$ is equal to the set $A'(e)$, the consistency checking for the rest of the objects in $A'(e)$ is not necessary and is not performed.

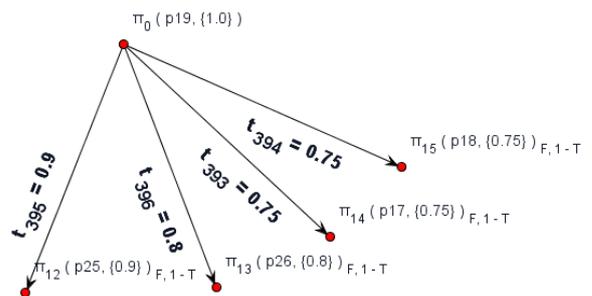


Figure 4. Inheritance tree for the object $coral$ that matches the likely context of the image e

The resulting annotation set after the consistency checking for image e is $A'(e) = \{water, fish, coral\}$.

If there is only one intruder class among the object labels in the set $A(e)$, the average precision of the image annotation can be increased, Fig. 5a), but if the majority of labels for an instance are wrong, the consistency checking can discard the true labels and the precision falls, Fig. 5b).

	a)	b)
Image example e		
$A(e)$	<i>coral, sky, wolf, trees, grass</i>	<i>shuttle, train, building</i>
$A'(e)$	<i>sky, wolf, trees, grass</i>	<i>train, building</i>

Figure 5. a) Positive and b) negative example of consistency checking

4. Replacement phase

We use the fuzzy intersection algorithm to find objects similar to the detected intruder that better fit the context of an image, based on their properties. The algorithm can find relationships between the facts stored in a knowledge base by generating inheritance trees for two nodes and finding their intersection. The nodes that appear in both inheritance trees are called intersection nodes.

If an intruder object d_i is detected in the consistency checking procedure it is compared with all other objects $d_j \in D \setminus \{d_i\}$ in the knowledge base using the fuzzy intersection algorithm to find possible replacement. For each pair d_i, d_j the algorithm finds the intersection nodes of their corresponding inheritance trees. The places of intersection nodes correspond to spatial or perceptual properties of objects. The object that has the biggest intersection is considered as the most similar to the intruder. It is tested for consistency with other concepts in the set $A'(e)$. If it is consistent, it is used as replacement; otherwise, the label is discarded.

The steps of the fuzzy intersection algorithm are as follows:

Algorithm 2. Fuzzy intersection algorithm adjusted for label replacement.

Input:

Concepts $d_i, d_j \in D$, depth of inheritance k .

Facts in the knowledge base represented with KRFPN

Output:

Intersection nodes of concepts d_i and d_j .

Steps:

1. For the given concepts d_i and d_j , find the corresponding places $p_k \in P$ and $p_l \in P$ using the function α^{-1} : $\alpha^{-1}(d_i) = p_k, \alpha^{-1}(d_j) = p_l$.
2. Define the initial distributions of tokens Ω_0^I and Ω_0^{II} , such that $\Omega_0^I(p_i) = \emptyset, i \neq k; \Omega_0^I(p_k) = \{m_1^I\}$, and $\Omega_0^{II}(p_i) = \emptyset, i \neq l; \Omega_0^{II}(p_l) = \{m_1^{II}\}$,
3. Define the root nodes of the trees $\pi_0^I(p_k, c\{m_1^I\})$ and $\pi_0^{II}(p_l, c\{m_1^{II}\})$ and set $c\{m_1^I\} = c\{m_1^{II}\} = 1$.
4. For the root nodes π_0^I construct k levels of the inheritance tree $InhTree^I$ using the Algorithm 1.
5. For the root nodes π_0^{II} construct k levels of the inheritance tree $InhTree^{II}$ using the Algorithm 1.
6. Find the nodes $\pi_{pr}^I, p, r = 0, 1, 2, \dots$ in $InhTree^I$ and π_{st}^{II} , $s, t = 0, 1, 2, \dots$ in $InhTree^{II}$ that match one another. Two nodes in different trees are intersection nodes if they have the same place, regardless of the number of tokens and their values.

4.1. Example of the intersection algorithm for replacement

The steps of the label replacement are shown using an example of automatic image annotation for Fig. 2. In the annotation set $A(e) = \{\textit{ground}, \textit{water}, \textit{fish}, \textit{coral}\}$ the label *ground* is detected as an intruder in the consistency checking phase. This label is compared with all other objects in the knowledge base; here only comparison with *trees* will be shown as demonstration.

For the concepts of interest, *ground* and *trees*, the root nodes of trees $InhTree^I$ and $InhTree^{II}$ are $\pi_0^I(p_{13}, \{1.0\})$ and $\pi_0^{II}(p_{25}, \{1.0\})$, respectively. The algorithm finds the intersection nodes π_{17}^I and π_{19}^{II} with the common place p_{125} and π_{18}^I and π_{14}^{II} with the common place p_{113} , as shown on Fig. 6. The place p_{125} corresponds to spatial property $\alpha(p_{125}) = \textit{at_the_top}$ and the place p_{113} corresponds to the perceptual property $\alpha(p_{113}) = \textit{brown_3}$.

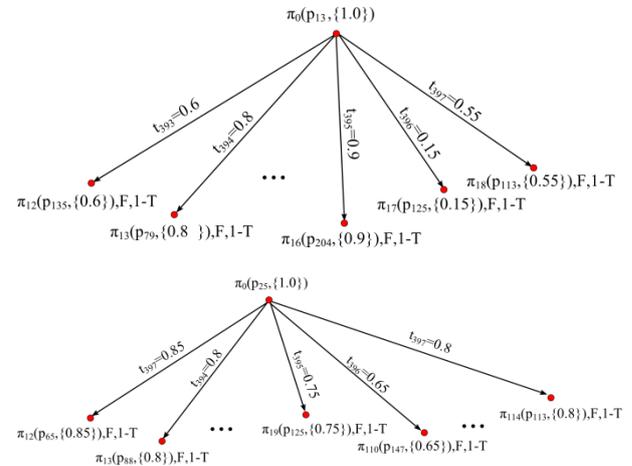


Figure 6. Example of inheritance trees with intersection nodes containing places p_{125} and p_{113}

After testing all objects in the knowledge base, the concept *rock* was determined as the candidate for replacement, as it has the biggest intersection with *ground*. After the consistency checking has shown *rock* is consistent with objects in $A'(e)$, it is used as replacement.

Fig. 8 shows additional examples of annotation when the detected intruder label is replaced with the concept chosen by the described procedure. In Fig 7a) the detected intruder (*coral*) was replaced with the concept that had the biggest intersection and was consistent with the remaining labels (*flowers*).

Fig 7b) shows an example when there are two wrong labels before consistency checking (*cheetah* and *coral*). The consistency checking algorithm detected the label *coral* as the intruder, and the replacement algorithm chose the replacement label *sand*. The replacement label fits the context, however the correct replacement should be *grass*. The label *cheetah* is not detected as an intruder during consistency checking, as it is consistent with other labels. Thus, it is not considered for replacement.

	a)	b)
Image example e		
$A(e)$	<i>coral, sky, trees, cloud, grass</i>	<i>grass, fox, cheetah, coral</i>
$A'(e)$	<i>flowers, sky, trees, cloud, grass</i>	<i>grass, fox, cheetah, sand</i>

Figure 7. Example of a) correct b) incorrect label replacement.

5. Conclusion

Automatic image annotation methods often use vocabularies that contain labels that correspond to objects (building, bear, train, etc.) or to scenes (forest, underwater, nature, etc.) because these types of words are typically used when searching for images. If it is assumed that scenes are composed of objects, then scene labels can be inferred from object labels. In this case, image annotation on object level should be as precise as possible to avoid the case where a misclassified object leads to wrong conclusion about the whole scene.

The approach proposed in this paper is to incorporate a fuzzy knowledge representation scheme and fuzzy inference algorithms into annotation process for annotation refinement. To represent knowledge about objects and their relationships, a fuzzy knowledge representation formalism, dubbed *KRFPN_r*, is used. The fuzzy inheritance algorithm is used with co-occurrence relations objects in the consistency-checking phase. The aim is to check whether each object label obtained with classification matches the most likely image context. The objects that do not match the context are considered intruders.

The fuzzy intersection algorithm is used with perceptual and spatial properties of objects in the replacement phase to detect possible replacement labels. The decision whether to discard or replace a detected intruder label is based on a confidence score that was experimentally set. The implementation of algorithms is demonstrated on examples of images representing outdoor scenes. Images are segmented and object classification results of segments are obtained.

Future work will include more precise determination of the replacement threshold value and testing of different strategies for deciding whether to discard or replace a label. The type of object (foreground or background) could also be used to guide or restrict the replacement of labels.

References

[1] Hare JS, Lewis PH, Enser PGB Sandom CJ. 2006 January 17-19. Mind the Gap: Another look at the problem of the semantic gap in image retrieval. Multimedia Content Analysis, Management and Retrieval, San Jose, California, USA.

[2] Ivašić-Kos, M.; Ribarić, S.; Ipšić, I. Low- and High-level Image Annotation Using Fuzzy Petri Net Knowledge Representation Scheme. International

Journal of Computer Information Systems and Industrial Management (IJCISIM). 4 (2012)

[3] J. Li and J. Z. Wang, "Real-Time Computerized Annotation of Pictures," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 30, 2008, pp. 985-1002.

[4] Duygulu, P., Barnard, K., Freitas, J.F.G. de, Forsyth, D. A., 2002. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary, ECCV 2002, UK, 2002, pp. 97-112.

[5] Monay F. and Gatica-Perez D., "On image auto-annotation with Latent Space Models", Proc. ACM Multimedia, Berkeley, CA, 2003, pp. 275-278.

[6] Datta, R., Joshi, D., Li, J. 2008. "Image Retrieval: Ideas, Influences, and Trends of the New Age", ACM Transactions on Computing Surveys, vol. 20, pp. 1-60, April 2008.

[7] Zhang, D., Islam, M. M., and Lu, G. (2012). A review on automatic image annotation techniques. Pattern Recognition, 45(1), 346-362.

[8] Wang, Y., Gong, S. (2007). Refining image annotation using contextual relations between words. ACM CIVR 07', July 9-11, Netherlands, 425-432.

[9] Wang, C, Jing, F., Zhang, L., & Zhang, H.-J. (2006). Image annotation refinement using random walk with restarts. ACM MM 06', October 23-27, Santa Barbara, California, USA. 647-650

[10] Wang, C., Jing, F., Zhang, L., Zhang, H.-J. Content based image annotation refinement. CVPR'07.

[11] Liu, J., Li, M., Ma, W.-Y., Liu, Q., & Lu, H. (2006). An adaptive graph model for automatic image annotation. ACM MIR 06', Santa Barbara, California, USA, October 26-27, 61-70.

[12] Zhou, X., Wang, M., Zhang, Q., Zhang, J., Shi, B. (2007). Automatic image annotation by an iterative approach: incorporating keyword correlations and region matching. ACM CIVR 07', July 9-11, Amsterdam, Netherlands, 425-432.

[13] Yohan, J., Khan, L., Wang, L., & Awad, M. (2005) Image annotations by combining multiple evidence and WordNet. ACM conference on Multimedia (MM05'), Singapore, 706-715.

[14] Cilibrasi, R. L., & Vitanyi, P. M. (2007). The google similarity distance. Knowledge and Data Engineering, IEEE Transactions on, 19(3), 370-383.

[15] Ribarić, S., Pavešić, N., 2009. "Inference Procedures for Fuzzy Knowledge Representation Scheme", Applied Artificial Intelligence, vol. 23, January 2009, pp. 16-43

[16] Džeroski, S., Cestnik, B. and Petrovski, I., 1993. Using the m-estimate in rule induction. J. Comput. Inf. Technol, 1: 37-46

[17] Isabelle Bloch, Fuzzy spatial relationships for image processing and interpretation: a review, Image and Vision Computing, vol. 23(2), 2005, pp 89-110.