Pattern Recognition 52 (2016) 287–305

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

# Two-tier image annotation model based on a multi-label classifier and fuzzy-knowledge representation scheme

Marina Ivasic-Kos [a,*], Miran Pobar [a], Slobodan Ribaric [b]

[a] Department of Informatics, University of Rijeka, Rijeka, Croatia
[b] Faculty of Electrical Engineering and Computing, University of Zagreb, Zagreb, Croatia

ARTICLE INFO

*Article history:*
Received 31 October 2014
Received in revised form
31 August 2015
Accepted 22 October 2015
Available online 6 November 2015

*Keywords:*
Image annotation
Knowledge representation
Inference algorithms
Fuzzy Petri Net
Multi-label image classification

ABSTRACT

Automatic image annotation involves automatically assigning useful keywords to an unlabelled image. The major goal is to bridge the so-called semantic gap between the available image features and the keywords that people might use to annotate images. Although different people will most likely use different words to annotate the same image, most people can use object or scene labels when searching for images.

We propose a two-tier annotation model where the first tier corresponds to object-level and the second tier to scene-level annotation. In the first tier, images are annotated with labels of objects present in them, using multi-label classification methods on low-level features extracted from images. Scene-level annotation is performed in the second tier, using the originally developed inference-based algorithms for annotation refinement and for scene recognition. These algorithms use a fuzzy knowledge representation scheme based on Fuzzy Petri Net, KRFPNs, that is defined to enable reasoning with concepts useful for image annotation. To define the elements of the KRFPNs scheme, novel data-driven algorithms for acquisition of fuzzy knowledge are proposed.

The proposed image annotation model is evaluated separately on the first and on the second tier using a dataset of outdoor images. The results outperform the published results obtained on the same image collection, both on the object-level and on scene-level annotation. Different subsets of features composed of dominant colours, image moments, and GIST descriptors, as well as different classification methods (RAKEL, ML-kNN and Naïve Bayes), were tested in the first tier. The results of scene level annotation in the second tier are also compared with a common classification method (Naïve Bayes) and have shown superior performance. The proposed model enables the expanding of image annotation with new concepts regardless of their level of abstraction.



## 1. Introduction

Image retrieval, search and organisation have become a problem on account of the huge number of images produced daily. In order to simplify these tasks, different approaches for image retrieval have been proposed that can be roughly divided into those that compare visual content (content-based image retrieval) and those that use text descriptions of images (text-based image retrieval) [27,7].

Image retrieval based on text has appeared to be easier, more natural and more suitable for people in most everyday cases. This is because it is much simpler to write a keyword-based query than to provide image examples, and it is likely that the user does not have an example image of the query. Besides, images corresponding to the same keywords can be very diverse. For example, if a person has an image of a town, but wants a different view, content-based retrieval would not be the best choice because most of the results would be too similar to the image he already has. On the other hand, very diverse images can be retrieved with a keyword query, e.g. for the keyword *Dubrovnik,* different views of the town can be retrieved (Fig. 1.).

To be able to retrieve images using text, they must be labelled or described in the surrounding text. However, most images are not. Since manually providing image annotation is a tedious and expensive task, especially when dealing with a large number of images, automatic image annotation has appeared as a solution.

* Correspondence to: R. Matejčić 2, 51000 Rijeka, Croatia. Tel.: +385 51584710.
*E-mail address:* marinai@uniri.hr (M. Ivasic-Kos).

http://dx.doi.org/10.1016/j.patcog.2015.10.017
0031-3203/

**Fig. 1.** Part of the image search results for the keyword "Dubrovnik".

| | a | b |
|---|---|---|
| |  |  |
| Object labels | *tracks, train, cloud, sky, trees,* | *snow, polar bear* |
| Scene label | *SceneTrain, Transportation* | *ScenePolarbear, WildLife, Arctic* |

**Fig. 2.** Examples of images and their annotation at object and scene levels.

Automatic annotation methods deal with visual features that can be extracted from the raw image data, such as colour, texture and structure, and can automatically assign metadata in the form of keywords from a controlled vocabulary to an unlabelled image. The major goal is to bridge the so-called semantic gap [12] between the available features and the keywords that could be useful to people. In the paper [15] an image representation model is given to reflect the semantic levels of words used in image annotation.

This problem is challenging because different people will most likely annotate the same image with different words that reflect their knowledge of the context of the image, their experience and cultural background. However, most people when searching for images use two types of labels: object and scene labels. Object labels correspond to objects that can be recognised in an image, like *sky, trees, tracks* and *train* for the image in Fig. 2a. Scene labels represent the context of the whole image, like *SceneTrain* or more general *Transportation* for Fig. 2a. Scene labels can be either directly obtained as a result of the global classification of image features [23] or inferred from object labels, as proposed in our approach.

Since object and scene labels are most commonly used, in this paper we focus on automatic image annotation at scene and object levels.

We propose a two-tier annotation model for automatic image annotation, where the first tier corresponds to object annotation and the second tier to scene-level annotation. An overview of the proposed model is given after the sections with the related work, in Section 3. The first assumption is that there can be many objects in any image, but an image can be classified into one scene. The second assumption is that there are typical objects of which scenes are composed. Since many object labels can be assigned to an image, the object-level annotation was treated as a multi-label problem and appropriate multi-label classification methods RAKEL and ML-kNN were used. The details of object level annotation are given in Section 4. The scene-level annotation relies on originally developed inference based algorithm defined for fuzzy knowledge representation scheme based on Fuzzy Petri Net, KRFPNs. The KRFPNs scheme and data-driven algorithms for knowledge acquisition about domain images are defined in Section 5. The inference approach for annotation refinement at the object level is given in Section 6. The conclusions about scenes are drawn using inference-based algorithms on facts about image domain and object labels obtained in the first tier, as detailed in Section 7. The major contributions of this paper can be summarised as follows:

1. The adaptive two-tier annotation model in which each tier can be independently used and modified.
2. The first tier uses multi-label classification to suit the image annotation at object level.
3. The second tier uses inference-based algorithms to handle the recognition of scenes and higher-level concepts that do not have specific low-level features.
4. The definition of the knowledge representation scheme based on Fuzzy Petri Nets, to represent knowledge about domain images that is often incomplete, uncertain and ambiguous.
5. Novel data-driven algorithms for acquisition of fuzzy knowledge about relations between objects and about scenes.
6. Novel inference based algorithm for annotation refinement to reduce the error propagation through the hierarchical structure of concepts during the inference of scenes.
7. Novel inference based algorithm for automatic scene recognition.
8. A comparison of inference based scene classification with an ordinary classification approach.

The performance of the proposed two-tier automatic annotation system was evaluated on outdoor images with regard to different feature subsets (dominant colours, moments, GIST descriptors [23] and compared to the published results obtained on the same image database as detailed in Section 8. The inference based scene classification is also compared with an ordinary classification approach and has shown better performance. The paper ends with a conclusion and directions for future work, Section 8.

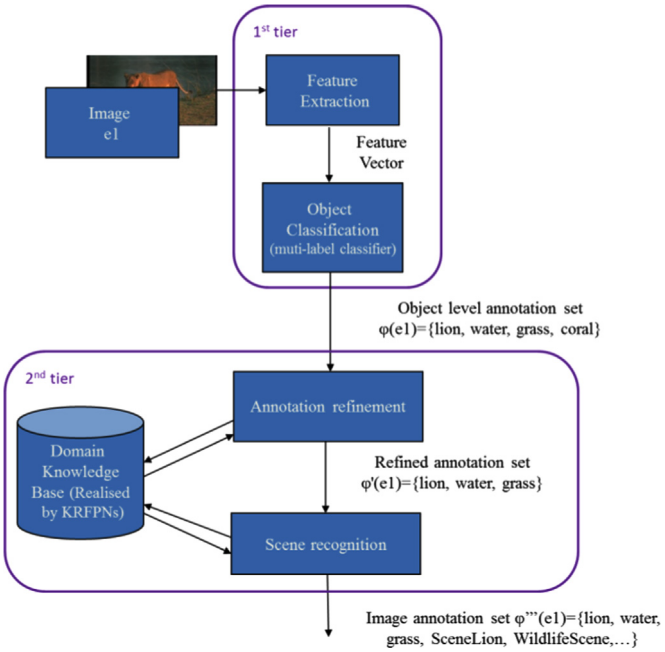


**Fig. 3.** Framework of a two-tier automatic image annotation system.

## 2. Related work

Automatic image annotation (AIA) has been an active research topic in recent years due to its potential impact on image retrieval, search, image interpretation and description. The AIA approaches proposed so far can be divided in various ways, e.g. according to the theory they are most related to [9] or according to the semantic level of concepts that are used for annotation [30]. AIA approaches are most closely related to statistical theory and machine learning or logical reasoning and artificial intelligence, while semantic levels can be organised into flat or structured vocabularies.

Classical AIA approaches belonging to the field of machine learning look for mapping between image features and concepts at object or scene levels. Classification and probabilistic modelling have been extensively used for this purpose. Methods based on classification, such as the one described in [17], treat each of the semantic keywords or concepts as an independent class and assign each of them to one classifier. Methods based on the translation model [10] and methods, which use latent semantic analysis [22] fall into the category of probabilistic methods that aim to learn a relevance model to represent correlations between images and keywords. A recent survey of research in this field can be found in [35,7].

Lately, graph-based methods have been intensively investigated to apply logical reasoning to images, and many graph-based image analysis algorithms have proven to be successful. Conceptual graphs which encode the interpretation of the image are used for image annotation. Pan et al. [24] have proposed a graph-based method for image annotation in which images, annotations and regions are considered as three types of nodes of a mixed media graph. In [19], automatic image annotation is performed using two graph-based learning processes. In the first graph, the nodes are images and the edges are relations between the images, and in the second graph the nodes are words and the edges are relations between the words. In [8], the authors aim to illustrate each of the concepts from the WordNet ontology with 500-1000 images in order to create public image ontology called ImageNet.

Within the project aceMedia, in [21] ontology is combined with fuzzy logic to generate concepts from the beach domain. In [1], the same group of authors has used a combination of different classifiers for learning concepts and fuzzy spatial relationships.

In [13], a framework based on Fuzzy Petri Nets is proposed for image annotation at object level. In the fuzzy-knowledge base, nodes are features and objects. Co-occurrence relations are defined between objects, and attribute relations are defined between objects and features.

The idea of introducing a fuzzy knowledge representation scheme into image annotation system proposed in our previous research [14,16], is further developed in this paper. New inference-based algorithms for scene classification and annotation refinement as well as algorithms for data-driven knowledge acquisition are formally defined and enhanced. As opposed to the previous work where object level classification was performed using classification of segmented images, object level annotation is treated here as a multi-label classification problem on the whole image.

Binder et al. [3] have proposed a method called Output Kernel Multi-Task Learning (MTL) to improve ranking performance. Zhang et al. [36] have defined a graph-based representation for loosely annotated images where each node is defined as a collection of discriminative image patches annotated with object category labels. The edge linking two nodes models the co-occurrence relationship among different objects in the same image.

## 3. Overview of the two-tier image annotation model

Images of outdoor scenes commonly contain one or more objects of interest, for instance *person, boat, dog, bridge* and different kinds of background objects such as *sky, grass, water*, etc. However, people often think about these images as a whole, interpreting them as scenes,

for example *tennis match* instead of *person, court, racket, net*, and *ball*. To make the image annotation more useful for organising and for the retrieval of images, it should contain both object and scene labels. Object labels correspond to classes whose instances can be recognised in an image. Scene labels are used to represent the context or semantics of the whole image, according to common sense and expert knowledge. To deal with different levels of annotation with object and scene labels, a two-tier automatic image annotation model using a multi-label classifier and a fuzzy-knowledge representation scheme is proposed. In the first tier, multi-label classification is proposed since it can provide reliable recognition of specific objects without the need to include knowledge. However, higher-level concepts usually do not correspond to specific low-level features, and inference based on knowledge about the domain is therefore more appropriate. The relationships between objects in a scene can be easily identified and are generally unchanging, so they can be used to infer scenes and more abstract concepts. Still, knowledge about objects and scenes is usually imprecise, somewhat ambiguous and incomplete. Therefore, fuzzy-knowledge representation is used in the second tier. The architecture of the proposed system is depicted in Fig. 3. The input to the system is an unlabelled image and the results of automatic annotation are object and scene labels. First, from each image, low-level features are extracted which represent the geometric and photometric properties of the image. Each image is then represented by the $m$-component feature vector $\boldsymbol{x}=(x_1,x_2,\ldots,x_m)^T$. The obtained features are used for object classification. The assumption is that more than one object can be relevant for image annotation, so multi-label classification methods are used. The result of the image classification in the first tier is object labels from the set $C$ of all object labels. In the second tier, the object labels are used for scene-level classification. Scene-level classification is supported by the image-domain knowledge base, where each scene is defined as a composition of typical object classes. Other concepts at different levels of abstraction can be inferred using the defined hierarchical relations among concepts. Relations among objects and scenes can also be used to check the consistency of object labels and to discard those that do not fit the likely context. In this way, the propagation of errors is reduced through the hierarchical structure of concepts during the inference of scenes or more abstract concepts.

To represent fuzzy knowledge about the domain of interest, a knowledge-representation scheme based on the Fuzzy Petri Net formalism [26] is defined. The proposed scheme has the ability to cope with uncertain, imprecise, and fuzzy knowledge about concepts and relations, and enables reasoning about them. The knowledge-representation scheme related to image domain and inference-based algorithms for annotation refinement and for scene recognition are implemented as Java classes. The elements of the scheme are generated utilising the proposed data-driven algorithms for acquisition of fuzzy knowledge, which are implemented in Matlab.

## 4. Multi-label classification in the first tier

In the general case, an image can contain multiple objects in both the foreground and the background. All objects appearing on the image, or many of them, can be useful or important for automatic image interpretation.

All objects appearing on the image are potentially useful and hold important information for automatic image interpretation. To preserve as much information about the image as possible, many object labels are usually assigned to an image. Therefore, image annotation at the object level is an inherently multi-label classification problem. Here we have used different feature sets with appropriate multi-label classification methods.

### 4.1. Feature sets

The variety of perceptual and semantic information about objects in an outdoor image could be contained in global low-level features such as dominant colour, spatial structure, colour histogram and texture. For classification in the first tier, we used a feature set made up of dominant colours of the whole image (global dominant colours) and of parts of the image (region based dominant colours), colour moments and the GIST descriptor. The used set of features is constructed to represent the varied aspects of image representation including colour and structure information.

The dominant colours were selected from the colour histogram of the image. The colour histogram was calculated for each of the RGB colour channels of the whole image. Next, histogram bins with the highest values for each channel were selected as dominant colours. After

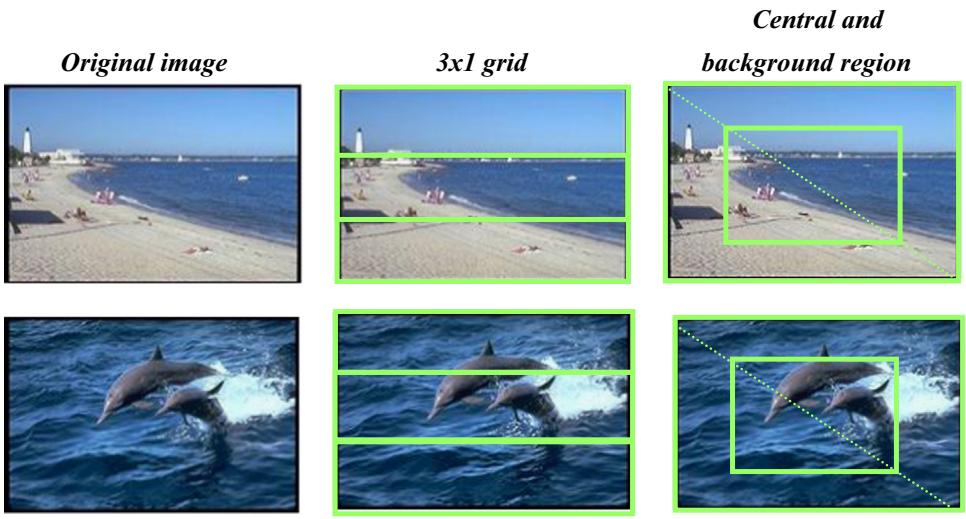


**Fig. 4.** Original images and the arrangement of a 3x1 image grid and central and background regions from which the dominant colours features were computed.

experimenting with different number of different colours (3–36) per channel we chose to use 12 of them in each image as features (referred to as DC) for our classification tasks.

The information about the colour layout of an image was preserved using 5 local RGB histograms, from which dominant colours were extracted in the same manner as for the whole image. The local dominant colours are referred to as DC1 to DC5. To calculate the DC1, DC2 and DC3 local features, a histogram was computed for each cell of a 3x1 grid applied to each image. The DC4 feature was computed in the central part of the image, presumably containing the main image object, and the DC5 feature in the surrounding part that would probably contain the background, as shown in Fig. 4. The size of the central part was 1/4 of the diagonal size of the whole image, and of the same proportions. There are 12 dominant colours with 3 channels giving a 36-dimensional feature vector of global DC. The size of local dominant colours vector (DC1 to DC5) is 180 (12 dominant coloursx3 channelsx5 image parts). Additionally, we computed the colour moments (CM) for each RGB channel: mean, standard deviation, skewness and kurtosis. The size of the CM feature vector is 12 (4 colour moments for 3 channels).

The GIST image descriptor [23] that was proven to be efficient for scene recognition was also used as a region-based feature. It is a structure-based image descriptor that refers to the dominant spatial structure of the image characterized by the properties of its boundaries (e.g., the size, degree of openness, perspective) and its content (e.g., naturalness, roughness). The spatial properties are estimated using the global features computed as a weighted combination of Gabor-like multi scale-oriented filters. In our case, we used $8 \times 8$ encoding samples in the GIST descriptor within 8 orientations per 8 scales of image components, so the GIST feature vector has 512 components.

We performed the classification tasks using all the extracted features (DC, DC1..DC5, CM, GIST). The size of the feature vector used for classification was 740. Since the size of the feature vector is large in proportion to the number of images, we also tested the classification performance using five subsets.

### 4.2. Image annotation at object level

We attempt to label images with both foreground and background object labels, assuming that they are all useful for image annotation. Since an image may be associated with multiple object labels, the annotation at the object level in the first tier is treated as a multi-label classification problem. Multi-label image classification can be formally expressed as $\varphi$: $E \rightarrow \mathcal{P}(C)$, where E is a set of image examples, $\mathcal{P}(C)$ is a power set of the set of class labels $C$ and there is at least one example $e_j \in E$ that is mapped into two or more classes, i.e. $\exists e_j \in E: |\varphi(e_j)| \geq 2$.

The methods most commonly used to tackle a multi-label classification problem can be divided into two different approaches [29]. In the first, the multi-label classification problem is transformed into more single-label classification problems [20], known as the data transformation approach. The aim is to transform the data so that any classification method designed for single-label classification can be applied. However, data transformation approaches usually do not exploit the patterns of labels that naturally occur in multi-label setting. For example, *dolphin* usually appears together with *sea*, while observing the labels *cheetah* and *fish* is much less common. On the other hand, algorithm adaptation methods extend specific learning algorithms in order to handle multi-label data directly and can sometimes utilise the inherent co-occurrence of labels. Another issue in multi-label classification is the fact that typically some labels occur more rarely in the data and it may be difficult to train individual classifiers for those labels.

For the multi-label classification task, we used the Multi-label k-Nearest Neighbour (ML-kNN) [34], a lazy learning algorithm derived from the traditional kNN algorithm, and RAKEL (RAndom k-labELsets) [28] which is an example of data adaptation methods. The kNN classifier is a nonlinear and adaptive algorithm that is rather well suited for rarely occurring labels. The RAKEL algorithm considers a small random subset of labels and uses a single-label classifier for the prediction of each element in the power set of this subset. In this way, the algorithm aims to take into account label correlations using single-label classifiers. In our case, kNN classifier and the C4.5 classification tree are used as base single-label classifiers for RAKEL. The base classifiers are applied to subtasks with a manageable number of labels. It was experimentally shown that for our task the best results are obtained using RAKEL with the C4.5 as base classifier. We also used the Naïve Bayes (NB) classifier along with data transformation. The data were transformed so that each instance with multiple labels was replaced with elements of the binary relation $\rho \subseteq E \times C$ between a set of samples $E$ and a set of class labels $C$. An ordered pair $(e, c) \in E \times C$ can be interpreted as "$e$ is classified into $c$" and is often written as $e\rho c$. For example, if an image example $e_{15} \in E$ was annotated with labels $\varphi(e_{15}) = \{lion, sky, grass\}, lion, sky, grass \in E$, it was transformed into three single-label instances $e_{15}\rho lion, e_{15}\rho sky, e_{15}\rho grass$. With NB, a model is learned for each label separately, and the result is the union of all labels that are obtained as a result of a binary classification in one-vs.-all fashion and with RAKEL for small subsets of labels. Unlike the NB, with RAKEL and ML-kNN a model is learned for small subsets of labels and therefore as a result of the classification only such subsets can be obtained. In both cases, incorrect labels can also be assigned to an image along with correct ones if they are in the assigned subset for RAKEL and kNN case, or if they are a result of binary misclassification for NB. For the one-vs.-all case, the inherent correlations among labels are not used, but an annotation refinement post-processing step is proposed here that incorporates label correlations and excludes from results labels that do not normally occur together with other labels in images.

The object labels and the corresponding confidence obtained as a result of the multi-label classification in the first tier of the image annotation system are used as features for the recognition of scenes in the second tier.

In case when the Naïve Bayes classifier is used, confidence of each object is set according to the posterior probability of that object class. Otherwise, when classifier used in the first tier cannot provide information on the confidence of the classification, e.g. when the ML-kNN classifier is used, the confidence value is set to 1. Object labels from the annotation set $\varphi(e) \subseteq C$ and the corresponding confidence values make the set $\Phi(e) \subset \varphi(e) \times [0, 1]$ that is input to the inference based algorithm for scene recognition in the second tier.

## 5. Knowledge representation in the second tier

The assumption is that there are typical objects of which scenes are composed, so each scene is treated as a composition of objects that are selected as typical, based on the used training dataset. To enable making inferences about scenes, relations between objects and scenes and relations between objects in an image are modelled and stored in a knowledge base. Facts about scenes and objects are collected

automatically from the training dataset and organised in a knowledge representation scheme. It is a faster and more effective way of collecting specific knowledge then manually building the knowledge base, since it is a broad, uncertain domain of knowledge and expertise is not clearly defined. An expert only defines facts related to general knowledge like hierarchical relationships between concepts so the need for manual definition of facts is reduced drastically. However, an expert can still review generated facts and modify them if necessary. Due to the inductive nature of generating facts, the acquired knowledge is often incomplete and unreliable, so the ability to draw conclusions about scenes from imprecise, vague knowledge becomes necessary. Additionally, the inference of an unknown scene relies on facts about objects that are obtained as classification results in the first tier and that are subject to errors. Apart from using a knowledge representation scheme that allows representation of fuzzy knowledge and can handle uncertain reasoning, the algorithm for object annotation refinement is proposed to detect and discard those objects that do not fit the context and so to prevent error propagation.

For the purpose of knowledge representation, a scheme based on KRFPN formalism [26], which is in turn based on the Fuzzy Petri Net, is defined to represent knowledge about outdoor images as well as to support subsequent reasoning about scenes in the second tier of the system. The definition of a fuzzy knowledge representation scheme adapted for automatic image annotation, named KRFPNs and the novel data-driven algorithms for automatic acquisition of knowledge about object and scenes, and relations between them, are detailed below.

### 5.1. Definition of the KRFPNs scheme adapted for image annotation

The elements of the knowledge base used for the annotation of outdoor images, particularly fuzzy knowledge about scenes and relationships among objects, are presented using the KRFPNs scheme.

The KRFPNs scheme concatenates elements of the Fuzzy Petri Net (*FPN*) with a semantic interpretation and is defined as:

$$KRFPNs = \left(FPN, \hat{\alpha}, \hat{\beta}, D, \Sigma\right). \tag{1}$$

$FPN = (P, T, I, O, M, \Omega, f, c)$ is a Fuzzy Petri Net where

$P = \{p_1, p_2, ..., p_n\}, n \in \mathbb{N}$ is a finite set of places, $T = \{t_1, t_2, ..., t_m\}, m \in \mathbb{N}$ is a finite set of transitions, $I : T \to \mathcal{P}(P)/\varnothing$ is the input function and $O : T \to \mathcal{P}(P)/\varnothing$ is the output function, where $\mathcal{P}(P)$ is the power set of places. The sets of places and transitions are disjunctive, $P \cap T = \varnothing$, and the link between places and transitions is given with the input and output functions. Elements $P, T, I, O$ are parts of an ordinary Petri Net (PN).

$M = \{m_1, m_2, \ldots, m_r\}, r \geq 0$ is a set of tokens used to define the state of a PN in discrete steps. The state of the net in step $w$ is defined with distribution of tokens:$\Omega_w$: $P \to \mathcal{P}(M)$, where $\mathcal{P}(M)$ is a power set of $M$. The initial distribution of tokens is denoted as $\Omega_0$ and in our case, in the initial distribution a place can have no or at most one token, so $|\Omega_0(p)| \in \{0, 1\}$. A place $p$ is marked in step $w$ if $\Omega_w(p) \neq \varnothing$. Marked places are important for the firing of transitions and the execution of the net. Firing of the transitions changes the distribution of tokens and causes the change of the net state.

Fuzziness in the scheme is implemented with functions $c : M \to [0, 1]$ and $f : T \to [0, 1]$ that associate tokens and transitions with truth values in the range [0, 1], where zero means "not true" and one "always true".

Semantic interpretation of places and transitions is implemented in the scheme with functions $\hat{\alpha}$ and $\hat{\beta}$ in order to link elements of the scheme with concepts related to image annotation. The function $\hat{\alpha}$: $P \to \mathrm{D} \times [0, 1]$, maps a place $p_i \in P$ to a concept and the corresponding confidence value$(d_j, c(d_j)) : d_j \in D, c(d_j) = \max c(m_l), m_l \in \Omega(p_i)$. If there are no tokens in the place, the token value is zero. An alternative function that differs in the codomain is defined, $\alpha$: $P \to D$. The function $\alpha$, is a bijective function, so its inverse exists$\alpha^{-1}$: $D \to P$. Those functions are useful for mapping a place $p_i \in P$ in the scheme to a concept $d_j$ and vice versa, when no information about the confidence of a concept is needed.

The set of concepts $D$ contains object and scene labels used for annotation of outdoor images,$D = C \cup SC$. A set $C$ contains object labels such as, $\{Airplane, Train, Shuttle, Ground, Cloud, Sky, Coral, Dolphin, Bird, Lion, Mountain\} \subset C$ and set $SC$ contains scene labels on different is $SC$ = {SceneAirplane, SceneBear, .., Seaside, Space, Undersea}.

The function $\hat{\beta}$: $T \to \Sigma \times [0, 1] \subset \mathrm{D} \times \mathrm{D} \times [0, 1]$,maps a transition $t_j \in T$to a pair $(d_{pk}\sigma_i d_{pl}, c_j)$:$\sigma_i \in \Sigma, d_{pk}, d_{pl} \in D, \alpha^{-1}(d_{pk}) = p_k, \alpha^{-1}(d_{pl}) = p_l$ , $p_k \in I(t_j), p_l \in O(t_j), c_j = f(t_j)$ where the first element is the relation between concepts linked with input and output places of the transition $t_j$ and the second is its confidence value. The function $\beta$: $T \to \Sigma$ that differs in the codomain is also defined and is useful for checking whether there is an association between a transition and a relation.

The set $\Sigma$ contains relations and is defined as $\Sigma =$ o*ccurs_ with*, *is_part_of*, *is_a*. The relation $\sigma_1 = occurs\_with$ is a pseudo-spatial relation that is defined between objects and it models the joint occurrence of objects in images. The compositional relation $\sigma_2 = is\_part\_of$ is defined between scenes and objects, reflecting the assumption that objects are parts of a scene. The hierarchical relation $\sigma_3 = is\_a$ is defined between scenes to model a taxonomic hierarchy between scenes. The confidence values of $\sigma_1 = occurs\_with$ and $\sigma_2 = is\_part\_of$ relations are defined according to the used training dataset, and in case of $\sigma_3 = is\_a$ it is defined by an expert.

### 5.2. Data-driven knowledge acquisition and automatic definition of KRFPNs scheme

The process of knowledge acquisition related to annotation refinement and scene recognition is supported with algorithms that enable automatic extraction of knowledge from existing data in the training set and arrangement of acquired knowledge in the knowledge representation scheme. Automated knowledge acquisition is a fast and economic way of collecting knowledge about the domain and can be used to estimate the confidence values of acquired facts especially when the domain knowledge is by nature broad as is the case with the image annotation. In that case experts are not able to provide a comprehensive overview of possible relationships between objects in different scenes or objectively evaluate their reliability. Automatically accumulated knowledge is very dependent on the examples in the data set, and to make it more general the estimated confidences of facts are further adjusted. In addition, to improve the knowledge base consistency, completeness and accuracy, and to enhance the inference process, an expert can review or modify the existing facts and rules or add new ones.

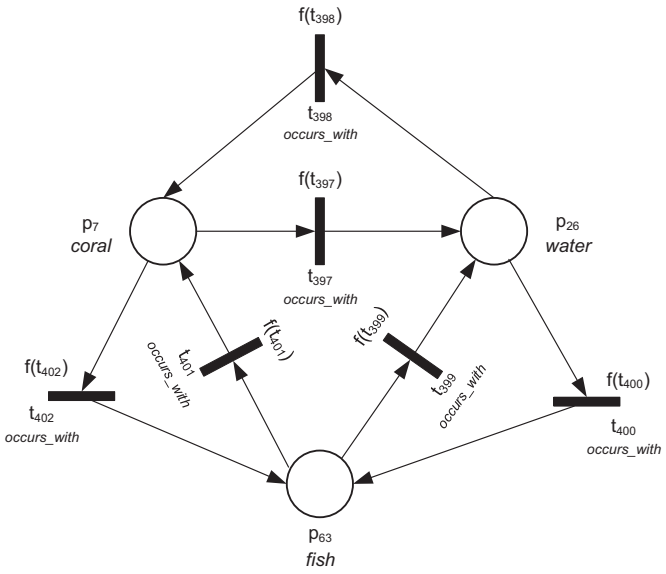


**Fig. 5.** Elements of KRFPN scheme related to relation $\sigma_1 = occurs\_with$ and the annotation set $\varphi(e) = \{water, fish, coral\}$.

*5.2.1. Automatic definition of KRFPNs scheme elements used for annotation refinement*

To check whether the objects obtained as a result of classification are consistent, the *occurs_with* relation that is defined between objects that can appear together on the same scene is used. This validation step is important to prevent further error propagation in the second tier, since reasoning about scenes is based on the objects present in the image.

To automatically generate the *occurs_with* relation between objects, as well as the corresponding confidence values, a data-driven algorithm is proposed (Algorithm 1). The input to the Algorithm 1 is the training set $\Gamma = \{\varphi(e)\colon e \in E\}$ containing annotations of images at the object level. The steps of the algorithm for definition of elements of the KRFPNs scheme related to the *occurs_with* relation are as follows:

**Algorithm 1.** Definition of elements of the KRFPNs scheme related to the *occurs_with* relation.

**Input:**

The training set containing annotations of images $\Gamma = \{\varphi(e)\colon e \in E\}$ at the object level.

**Steps:**

1. Define the set $C = \{d_i\colon \exists e \in E, d_i \in \varphi(e), \varphi\colon E \to \mathcal{P}(C)$.
2. $\forall d_i \in C \subset D$ pick a place $p_i \in \mathrm{P}$ and set $\alpha(p_i) = d_i, \alpha\colon P \to D$.
   - Note 1: $\alpha$ is a bijective function, so $\alpha^{-1}(d_a) = p_a, \alpha^{-1}\colon D \to P$
   - Note 2: place $p_i \in \mathrm{P}$ is not marked, so $\hat{\alpha}(p_i) = (d_i, 0), \hat{\alpha}\colon P \to \mathrm{D} \times [0,1]$
3. Let the relation $\sigma_1 = occurs\_with \in \Sigma$ be defined as
   $\sigma_1 = \{(d_a, d_b)\colon d_a, d_b \in C, a \neq b, \exists e \in E\colon \{d_a, d_b\} \subseteq \varphi(e); \varphi\colon E \to \mathcal{P}(C)\}$.
4. $\forall (d_a, d_b) \in \sigma_1, \sigma_1 \in \Sigma$ do:
   4.1 compute $c_j = P(d_b d_a) = \frac{P(d_a \cap d_b)}{P(d_a)} \cong \frac{f_{d_a d_b} + \hat{m} v_{d_a}}{f_{d_a} + \hat{m}}, a \neq b, c_j \in [0,1]$.
   $v_{d_a}$ is prior estimate of $P(d_b d_a)$ that is experimentally set to $v_{d_a} = \frac{1}{|D|}$, $\hat{m}$ is the weight that says how confident we are of our prior estimate $v_{d_a}$.
   4.2 define $t_j \in \mathrm{T}\colon \hat{\beta}(t_j) = (d_a \sigma_1 d_b, c_j), \hat{\beta}\colon T \to \Sigma \times [0,1]$.
   - Note 3: it holds $\beta(t_j) = d_a \sigma_1 d_b$, since $\beta\colon T \to \Sigma$
   4.3 set the transition value $f(t_j) = c_j, f\colon T \to [0,1]$.
   4.4 set the input function $I(t_j) = \alpha^{-1}(d_a) = p_a, I\colon T \to \mathcal{P}(P) \setminus \varnothing$.
   4.5 set the output function $O(t_j) = \alpha^{-1}(d_b) = p_b, O\colon T \to \mathcal{P}(P) \setminus \varnothing$.

**Output:**

Elements $\mathrm{C}, \mathrm{P}, T, I, O, f, \hat{\alpha}, \hat{\beta}$ of the KRFPNs scheme related to the $\sigma_1 = occurs\_with$ relation.

The process of defining the elements of the KRFPNs scheme related to the occurs_with relation by means of the Algorithm 1 is illustrated using an example annotation set φ(e). Let $\varphi(\mathrm{e}) = \{\mathrm{water}, \mathrm{fish}, \mathrm{coral}\}$ be the annotation set containing object labels of an example image e from the set E. The training set $\Gamma$ of all annotations φ(e) is first used to determine the set $\mathrm{C} = \mathrm{Airplane}, \mathrm{Bird}, \mathrm{Cloud}, \ldots, \mathrm{Shuttle}, \mathrm{Sky}, \mathrm{Train}$ of object labels used to annotate images in the first tier of the automatic image annotation system.

Each element $d_i \in C$, is mapped to a place $p_i$ in the KRFPNs scheme. For instance, let $d_i = coral \in C$ be mapped to the place $p_7$, $\alpha^{-1}(coral) = p_7$.

In the step 3, the relation $\sigma_1 = occurs\_with$ is defined between pairs of concepts that appear together at least once among all annotation sets for the images in *E*. For the annotation set $\varphi(e) = water, fish, coral$ in this example, the relation $\sigma_{1|\varphi(e)}$ is defined as $\sigma_{1|\varphi(e)} =$

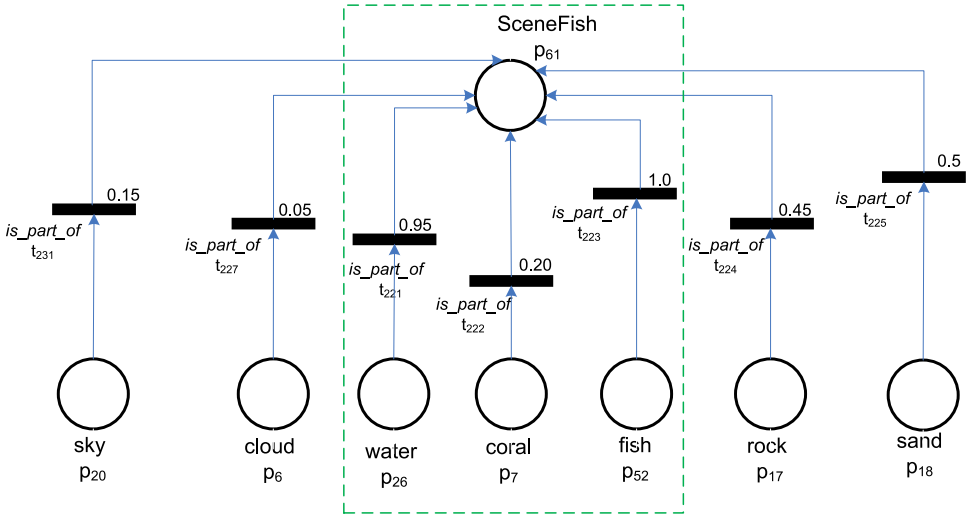


**Fig. 6.** Relations *is_part_of* among the scene "*SceneFish*" and its object components.

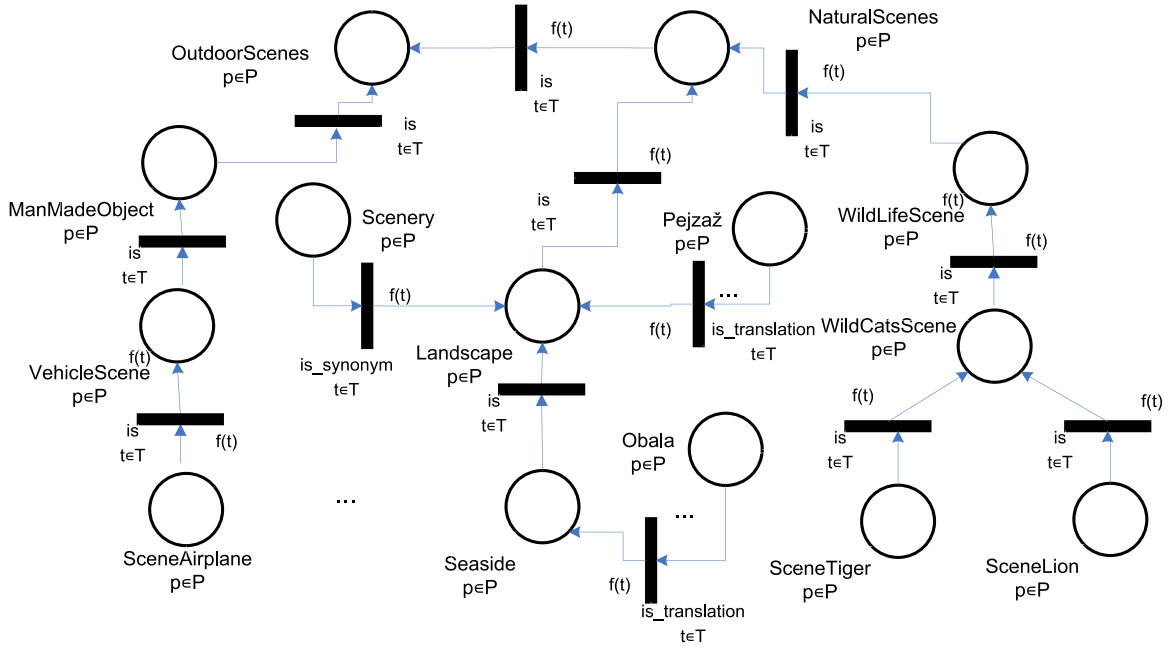


**Fig. 7.** A part of KRFPNs scheme expanded with *is_translation* and *is_synonym* relations.

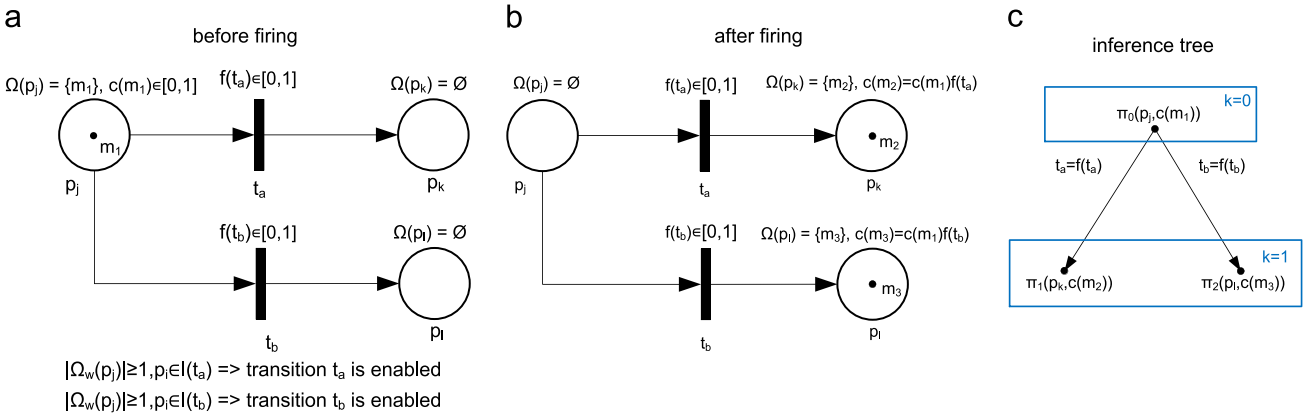


**Fig. 8.** Generic form of FPN formalism before (a) and after (b) firing enabled transitions and the corresponding inference tree (c).

**Fig. 9.** Example of an unlabelled (raw) image.

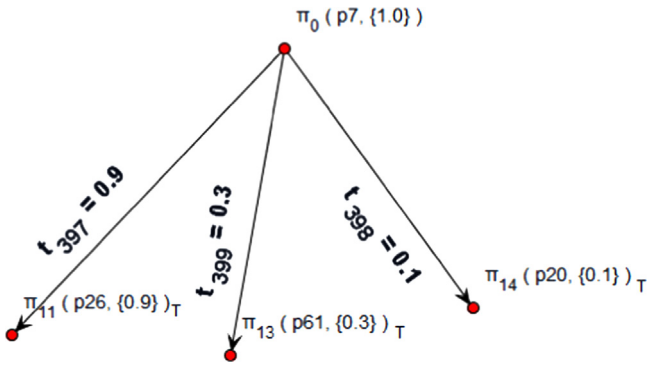


**Fig. 10.** Inheritance tree obtained with Algorithm 3 for the class "*coral*", with leaf nodes corresponding to objects in the set *D*.

$\{(water, fish), (fish, water), (water, coral), (coral, water), (fish, coral), (coral, fish)\}$. For each pair from $\sigma_{1|\varphi(e)}$ the corresponding transition, transition value, input and output functions are defined in step 4. For instance, for the pair *(coral, water)* the corresponding places are found, $\alpha^{-1}(coral) = p_7$, $\alpha^{-1}(water) = p_{26}$ and the transition $t_{397}$ is defined, so that $p_7$ is the input place and $p_{26}$ is the output place, $I(t_{397}) = p_7$, $O(t_{397}) = p_{26}$.

The confidence in a relationship $\sigma_1 = occurs\_with$ between objects in $C$ is expressed by a transition value. The transition value $f(t_{397})$ is set to the conditional probability of occurrence of the label *water* in all annotation sets, given that the label *coral* has occurred. It is estimated using the ratio of empirical relative frequencies of mutual occurrence of both objects *coral* and *water* and occurrence of object *coral,* among all annotation sets for images from the set *E*. In cases when there are very few or no examples of mutual occurrence of objects in the training annotation sets, an m – estimate of probabilities is implemented [11]. The m-estimate is used to fix the problem of zero posterior probability and to generalise the empirical relative frequencies beyond the used training set. The idea behind it is to pretend that we have a virtual set of an extra $\hat{m}$ training examples of object$d_a$, where $\hat{m}v_{d_a}$ of them co-occur with object $d_b$.

The transitions and places of the KRFPNs scheme can be presented by a directed bipartite graph $G = (P, T, A)$ whose vertices are divided into two disjoint sets $P$ and $T$, such that every directed edge in $A$ connects a vertex in $P$ with one in $T$. The vertices $p_i \in P$ are graphically represented by circles and the vertices $t_j \in T$ by bars. The set $A$ is a set of directed edges that contains ordered pairs of vertices $(I(t), t), t \in T, I(t) \subseteq P$ considered to be directed from $p$ to $t$ and ordered pairs of vertices $(t, O(t)), t \in T, O(t) \subseteq P$ considered to be directed from $t$ to $p$. An example of graphical representation of elements of KRFPN scheme related to relation $\sigma_1 = occurs\_with$ and the annotation set $\varphi(e) = water, fish, coral$ is shown in the Fig. 5.

The confidence of a concept $d_i$ can also be shown on the graph by a value of a token $c(m_k)$ when that token is within the place mapped to that particular concept. Initially marked places correspond to concepts that are results of classification in the first tier. The token values of each of those places can be set to the confidence of classification in the first tier to reflect the degree of uncertainty that the result of classification is that particular concept. A token is presented with a black dot within a place.

*5.2.2. Automatic definition of elements of KRFPNs scheme related to scene recognition*

Recognition of scenes in the second tier of the annotation system is realised on the assumption that a scene is a composition of several objects that can appear together on the same image. Between objects belonging to the same image and the scene the *is_part_of* relation is defined. The common occurrence of objects in the scenes, as well as the corresponding confidence values are determined automatically from the training set by analysing the annotation sets comprising both object and scene labels. The objects that are the most representative and discriminative for the given scene should have more impact on scene recognition, so the confidence of *is_part_of* relation is computed according to the weighted Bayes rule.

The steps of the data-driven Algorithm 2 for definition of elements of the KRFPN scheme related to the *is_part_of* relation are as follows:

**Algorithm 2.** Definition of elements of the KRFPN scheme related to is_part_of relation.

**Input:**

The training set containing annotations of images $\Gamma'' = \{\varphi(e) \cup \varphi''(e): e \in E\}$ at the object and scene level.

**Steps:**

1. Define the set $SC = d_j: \exists e \in E, d_j \in \varphi''(e), \varphi'': E \to \mathcal{P}(SC)$ of scene labels.

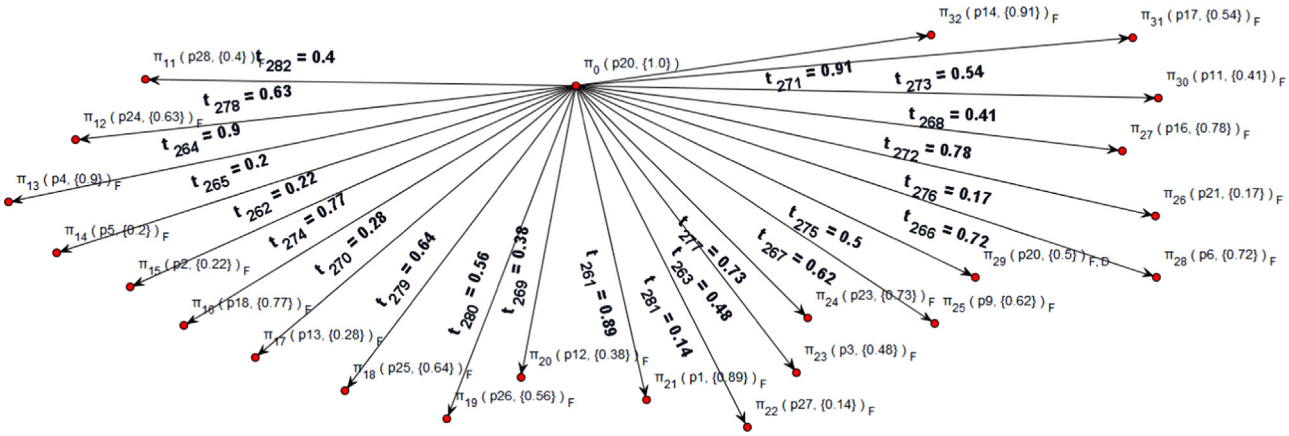


**Fig. 11.** Inheritance tree for the class "*sky*".

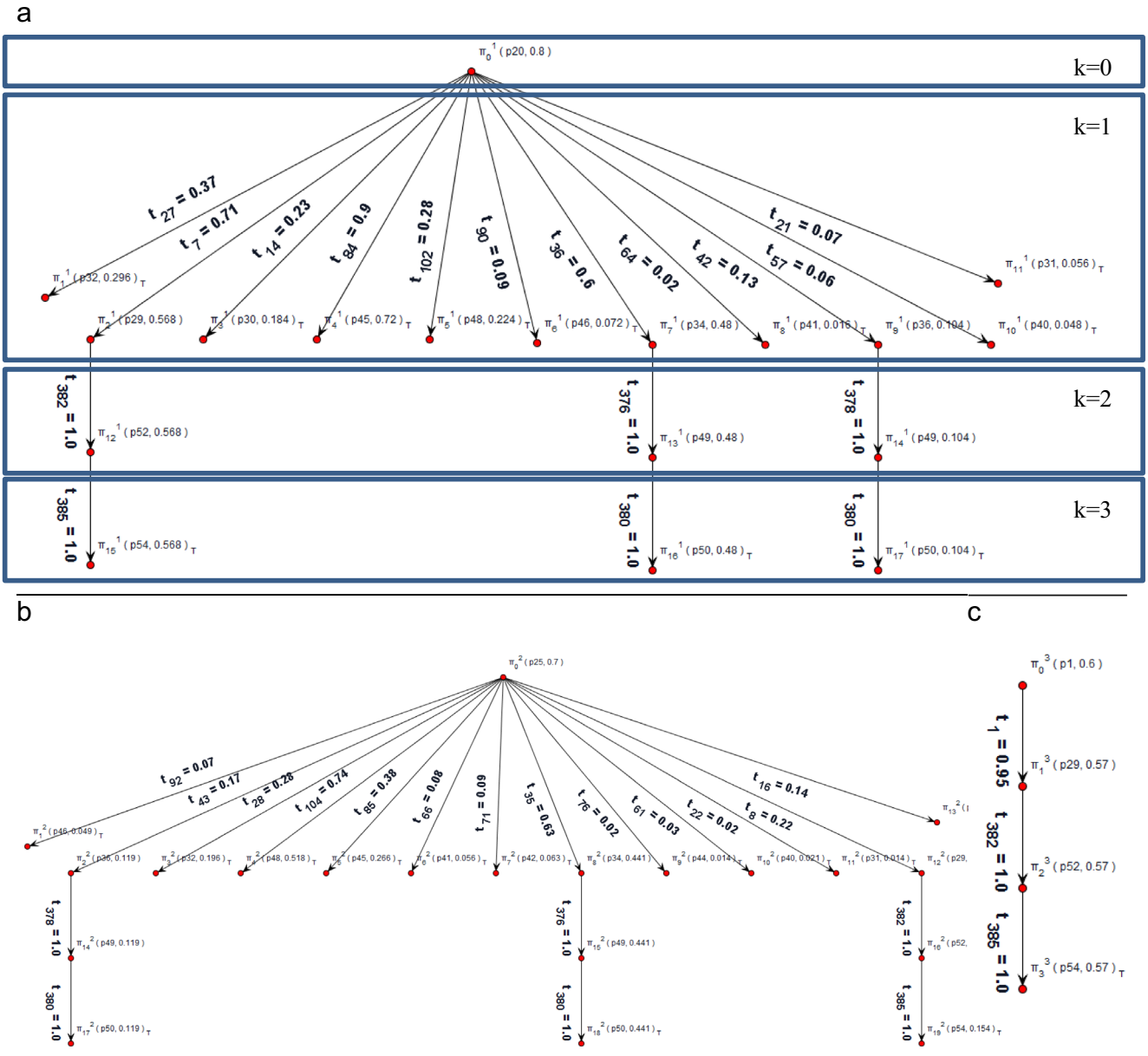


**Fig. 12.** Inference trees for concepts a) sky, b) tree and c) airplane.

2. $\forall d_i \in SC \subset D$ define $\alpha(p_{si}) = d_i, \alpha: P \to D$.
   - Note 1: $\alpha$ is a bijective function, so $\alpha^{-1}(d_i) = p_{si}, \alpha^{-1}: D \to P$
   - Note 2: place $p_i \in P$ is not marked, so $\hat{\alpha}(p_{si}) = (d_i, 0), \hat{\alpha}: P \to D \times [0,1]$
3. Let the relation $\sigma_2 = is\_part\_of \in \Sigma$ be defined as
$\sigma_2 = \{(d_a, d_b): d_a \in C, d_b \in SC, \exists e \in E: \{d_a, d_b\} \subseteq \varphi(e) \cup \varphi''(e); \varphi: E \to \mathcal{P}(C), \varphi'': E \to \mathcal{P}(SC)\}$, and is said that $d_a$ *is_part_of* $d_b$.
4. $\forall (d_a, d_b) \in \sigma_2, \sigma_2 \in \Sigma$ do:

   4.1 compute set $c_j = w_{d_a} P(d_b(d_a) = w_{d_a} \frac{P(d_a \cap d_b)}{P(d_a)} \cong w_{d_a} \frac{f_{d_a d_b} + \hat{m} v_{d_a}}{f_{d_a} + \hat{m}}, d_a \in C, d_b \in SC, c_j \in [0,1]$
   where $w_{d_a} = \log \frac{\sum_{d_x \in C} f_{d_x} + 1}{f_{d_a} + 1}$, $\hat{m}$ is a virtual set of samples chosen experimentally, and $v_{d_a}$ is the estimated probability that a sample in a virtual set is the object $d_a$.

   4.2 define $t_j \in T: \hat{\beta}(t_j) = (d_a \sigma_1 d_b, c_j), \hat{\beta}: T \to \Sigma \times [0,1]$.
   - Note 3: it holds $\beta(t_j) = d_a \sigma_2 d_b$, since $\beta: T \to \Sigma$

   4.3 set the transition value $f(t_j) = c_j, f: T \to [0,1]$.
   4.4 set the input function $I(t_j) = \alpha^{-1}(d_a) = p_a, I: T \to \mathcal{P}(P) \setminus \varnothing$.
   4.5 set the output function $O(t_j) = \alpha^{-1}(d_b) = p_{sb}, O: T \to \mathcal{P}(P) \setminus \varnothing$.

**Output:**

Elements $SC, P, T, I, O, f, \hat{\alpha}, \hat{\beta}$ of the KRFPNs scheme related to is_part_of relation

The *is_part_of* relation between scenes and objects is used to define scenes as compositions of objects. The definition of elements of the KRFPN scheme related to *is_part_of* relation is illustrated using an example. The objects are already associated with places in the scheme in Algorithm 1. Upon the same principle, using the function $\alpha^{-1}$ the places are assigned to scenes. For instance, $d_i = SceneFish$ will be mapped to the place $p_{61}$, $\alpha^{-1}(SceneFish) = p_{61}$.

To define the *is_part_of* relation, the annotation sets of images in $E$ that contain both scenes and object labels are used. If scene and object labels appear together in an image annotation set, it is assumed that objects are part of that scene and that *is_part_of* relation exists between them. Let $\varphi(e) \cup \varphi''(e) = \{water, fish, coral\} \cup \{SceneFish\} \subset C \cup SC$ be the annotation set of an example image $e$ comprising object and scene labels. Therefore, the relation $\sigma_2 = is\_part\_of$ for this annotation set $\varphi(e) \cup \varphi''(e)$ is defined as $\sigma_{2|\varphi(e) \cup \varphi''(e)} = \{(water, SceneFish), (fish, SceneFish), (coral, Scene\ Fish)\}$. In step 4, the transitions, transition values, input and output functions are defined for each pair from $\sigma_{2|\varphi(e) \cup \varphi''(e)}$. For example, the places corresponding to the pair *(coral, SceneFish),* are $\alpha^{-1}(coral) = p_7$ and $\alpha^{-1}(SceneFish) = p_{61}$ and the transition $t_{222}$ is defined so that $p_7$ is the input place, and $p_{61}$ is the output place, $I(t_{222}) = \{p_7\}$, $O(t_{222}) = \{p_{61}\}$.

Confidence in a relationship $\sigma_2 = is\_part\_of$ between scenes and objects is expressed by a transition value. The transition value $f(t_{222})$ is computed according to the weighted Bayes rule of occurrence of the scene *SceneFish* in all annotation sets, given that the object *coral* has occurred. It is estimated using the ratio of empirical relative frequencies of mutual occurrence of both scene *SceneFish* and object *coral* and occurrence of object *coral,* among all annotation sets for images from the set E. To generalise the empirical relative frequencies beyond used training set, particularly in cases when there are very few or no examples of mutual occurrence of objects in the training annotation sets, an m – estimate of probabilities is implemented [11] similarly as in Algorithm 1. In addition, the weight computed using the logarithm of the inverse object frequencies is used to increase the influence of those objects that are discriminant for a particular scene and are parts of a small number of scenes.

In Fig. 6. a part of the KRFPN scheme is presented, showing the *is_part_of* relations between the scene *SceneFish* and its components. The part representing the relation $\sigma_{2|\varphi(e) \cup \varphi''(e)}$ defined by the former procedure is highlighted with a dashed line.

Additionally, after the prototype of the knowledge base is automatically built, the expert can be involved to add to the models of domain knowledge at varying levels of generality by defining new relations and rules in the knowledge base. In the KRFPNs scheme, the $\sigma_3 = is\_a$ relation defined between scenes and their generalisations, expands the set of scenes *SC* with more abstract scene classes. Formally, the relation $\sigma_3 = is\_a \in \Sigma$ is defined as $\sigma_3 = \{(d_a, d_b): d_a, d_b \in SC, : \exists d_a \to \exists d_b\}$ and is said that $d_b$ *is a generalization of* $d_a$. Elements of the relation $\sigma_3 = is\_a$ are defined by the expert. For example, at the top level of the hierarchy are classes *OutdoorScene*, *NaturalScene*. Between classes in a hierarchy *is_a* relation is defined, e.g. *WildCatsScene is_a WildLifeScene*.

Generalisations of scenes are not the only concepts that can be added to the KRFPNs scheme; any other concept can also be added to the scheme by an expert and linked to existing concepts with the corresponding relations. For example, in Fig. 7a generic scheme is expanded with *is_synonym* relation between synonyms and *is_translation* relation to name the concept in a different language. Also, *is_a* relation between scenes at different levels of abstraction is shown.

## 6. Inference based annotation refinement

Image annotation at object level should be as precise as possible to avoid error propagation through the hierarchical structure of concepts, where a misclassified object in the first tier leads to a wrong conclusion about the whole scene in the second tier. In addition to improving the performance of classification itself, the classification results can be improved in a post-processing annotation refinement step. A group of authors has examined several different approaches for annotation refinement: Markov chains in [33], ontology in [18], conditional random fields (CRF) in [32], normalised Google distance (NGD) in [6], and re-ranking the annotations using the random walk with restarts algorithm in [31].

We propose an inference-based annotation refinement algorithm to automatically detect and discard misclassified object labels obtained with multi-label classification in the first tier. Annotation refinement is based on the consistency checking of object labels that relies on the execution of the Petri nets. The execution of the Petri net occurs in discrete steps. For each step a new token distribution and token values are defined. During the execution, the enabled transitions are fired, causing the change of the number and of the distribution

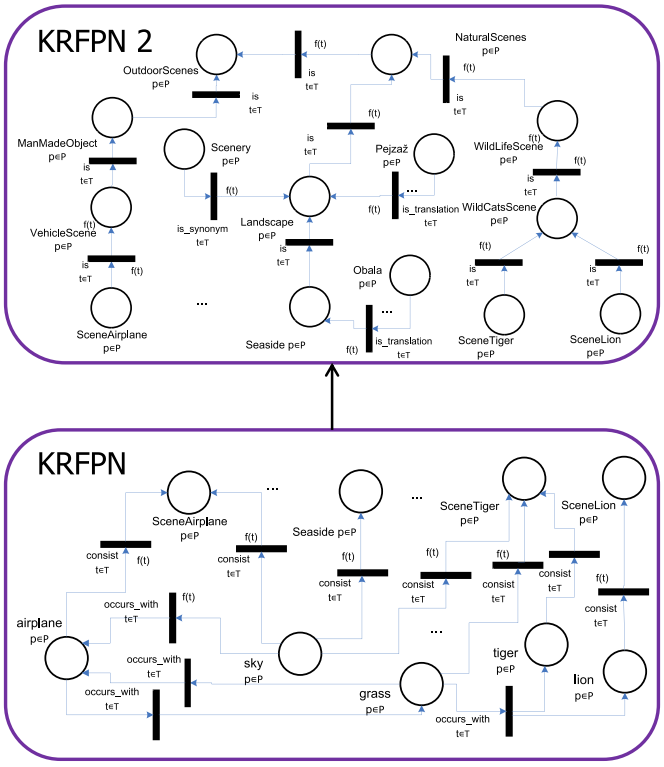


**Fig. 13.** Hierarchical connection of two KRFPNs schemes.

**Table 1**
Statistics of original and simplified datasets.

| Statistic | Original data | | Simplified data | |
|---|---|---|---|---|
| | Objects | Scenes | Objects | Scenes |
| No. of labels | 54 | 20 | 22 | 12 |
| Max images per label | 248 | 81 | 220 | 77 |
| Min images per label | 1 | 1 | 9 | 15 |
| Mean images per label | 26 | 25.2 | 50 | 32 |
| Median images per label | 7.5 | 19 | 28 | 25.5 |
| Std. dev. per label | 50 | 22 | 56 | 21 |

of tokens. A transition $t_j$ is enabled in step $w$ when every input place of the transition is marked, $|\Omega_w(p_i)| \geq 1, \forall p_i \in I(t_j)$,where $\Omega_w$ is distribution of tokens in step $w$. After firing the transition $t_j$, the new number of tokens in the place $p_i \in I(t_j) \cup O(t_j)$, in the step $w+1$ is given by: $|\Omega_{w+1}(p_i)| = |\Omega_w(p_i)| - |I(t_j) \cap p_i| + |O(t_j) \cap p_i|$. Firing a transition is in accordance with the basic firing rules of the original PN [25]. After the transition firing, a new token value $c(m_{k+1})$ at the output place is obtained as $c(m_k)f(t_j)$.

A generic example of the FPN scheme, depicting the state of the FPN before and after firing enabled transitions, is shown in Fig. 8. In the example, both of the transitions $t_a, t_b$ have one input and one output place; the input place of the transition$t_a$ is $I(t_a) = p_j$ and the output place is $O(t_a) = p_k$. By firing the transitions $t_a$ and $t_b$, the token $m_1$ is removed from the input place and the new tokens $m_2$ and $m_3$ are generated in output places with new token values.

The execution of the FPN is closely related to the inference process using fuzzy knowledge stored in the knowledge base. Inference process can be graphically presented by an inference tree related to inheritance and recognition trees [26] and derived from the reachability tree of the Petri net [5,25. The nodes in the each level of the inference tree represent the marked places of the net in each state. The nodes of the inference trees have the form$\pi\left(p_j, c(m_l)\right) j = 1, 2, \ldots, p,\ l = 1, 2, \ldots, r, 0 \leq r \leq |M|$, where the first component specifies the place $p_j$ where the token $m_l$ is located and the second one its value, $c(m_l)$. The arcs that link the nodes correspond to the fired transitions. The inference tree corresponding to the generic example of FPN scheme in Fig. 7a and b is shown in Fig. 7c. The level $k=0$ contains the root node containing the input place of the transitions $t_a$ and $t_b$and corresponds to the state of the net shown in Fig. 8a. The leaf nodes in the level $k=1$ correspond to the output places of transitions $t_a$ and $t_b$ that are marked in the state shown in Fig. 8b.

The inference tree is built in the breadth-first manner, so that for the initial token distribution $\Omega_0$ all enabled transitions are fired and then new directly reachable token distributions are generated. The inference tree is generated until there are no more unprocessed nodes (frontier nodes) or can be manually limited to a predefined level $k$. During the execution of algorithm, all nodes are classified as frontier, frozen (F), terminal (T), duplicate (D) or internal. A duplicate node is a node that has already appeared in the lower level of the tree, a terminal node is a node for which there are no enabled transitions and an internal node is a processed node that is neither duplicate, frozen nor terminal. A node is frozen if it is an output node of a transition associated with the spatial relations in order to stop further firing of the transitions where the hierarchical structure ends. Here, the output places of the transitions that represent the *occurs_with* relation are frozen.

**Table 2**
Evaluation results for object annotation level in the first tier.

| Feature subset | Classification method | Label-based results for object-level annotation | | | Instance-based results for object-level annotation | | |
|---|---|---|---|---|---|---|---|
| | | Precision | Recall | F1 score | Precision | Recall | F1 score |
| All | RAKEL-C4.5 | 0.39 | 0.28 | 0.30 | 0.61 | 0.54 | **0.54** |
| | RAKEL-kNN | **0.49** | 0.41 | 0.40 | 0.57 | 0.53 | 0.53 |
| | ML-kNN | 0.32 | 0.2 | 0.23 | **0.66** | 0.42 | 0.48 |
| | NB | 0.38 | 0.64 | **0.46** | 0.22 | 0.34 | 0.25 |
| GIST | RAKEL-C4.5 | 0.35 | 0.26 | 0.27 | 0.58 | 0.50 | 0.50 |
| | RAKEL-kNN | 0.48 | 0.45 | 0.43 | 0.57 | **0.55** | **0.54** |
| | ML-kNN | 0.26 | 0.17 | 0.19 | 0.60 | 0.38 | 0.44 |
| | NB | 0.31 | 0.65 | 0.40 | 0.21 | 0.38 | 0.26 |
| DC+CM | RAKEL-C4.5 | 0.40 | 0.27 | 0.29 | 0.60 | 0.50 | 0.52 |
| | RAKEL-kNN | 0.31 | 0.32 | 0.29 | 0.50 | 0.48 | 0.47 |
| | ML-kNN | 0.16 | 0.10 | 0.11 | 0.63 | 0.34 | 0.41 |
| | NB | 0.28 | 0.65 | 0.36 | 0.22 | 0.44 | 0.27 |
| DC1..DC5 + CM | RAKEL-C4.5 | 0.37 | 0.27 | 0.29 | 0.61 | 0.53 | **0.54** |
| | RAKEL-kNN | 0.37 | 0.31 | 0.30 | 0.51 | 0.46 | 0.47 |
| | ML-kNN | 0.19 | 0.11 | 0.12 | 0.63 | 0.33 | 0.41 |
| | NB | 0.32 | **0.67** | 0.41 | 0.22 | 0.39 | 0.26 |
| DC+ DC1..DC5 + CM | RAKEL-C4.5 | 0.39 | 0.27 | 0.30 | 0.59 | 0.52 | 0.52 |
| | RAKEL-kNN | 0.39 | 0.33 | 0.31 | 0.51 | 0.46 | 0.47 |
| | ML-kNN | 0.20 | 0.11 | 0.12 | 0.64 | 0.34 | 0.41 |
| | NB | 0.31 | 0.66 | 0.41 | 0.22 | 0.4 | 0.26 |

The *occurs_with* relation is used for annotation refinement to check if the object labels from the annotation set $\varphi(e)$ in the first tier are mutually consistent. The labels that cannot appear together with other labels in the set $\varphi(e)$ are considered intruders and are discarded. The *occurs_with* relation and facts about objects are included in the KRFPNs scheme with the Algorithm 1.

Object labels from the annotation set $\varphi(e) \subseteq C \subset D$ and the corresponding confidence values make the set $\Phi(\mathrm{e}) \subset \varphi(e) \times [0,1]$. The confidence values are provided either by the classifier used in the first tier of the image annotation system or are set to 1 if the used classifier doesn't give this information. Set $\Phi$(e) is input to the inference based algorithm for annotation refinement. The steps of the proposed inference based algorithm for annotation refinement are:

**Algorithm 3.** Inference based algorithm for annotation refinement.

**Input:**

The set $\Phi(\mathrm{e}) = \{(d_i, v_i), i = 1, 2, \ldots\} \subset \varphi(e) \times [0,1] \subset D \times [0,1]$.
Threshold value $\epsilon$ of relation $\sigma_1 = occurs\_with$ (default value $\epsilon = 0.05$).
Intruder threshold value $\delta = \frac{1}{2}(|\Phi_{|D}| - 1)$.
Elements of the KRFPNs scheme.

**Steps:**

1. Define totally ordered set $(\Phi, \psi)$; where relation $\psi$
$$\psi = \{((d_a, v_a), (d_b, v_b)) : (d_a, v_a), (d_b, v_b) \in \Phi \times \Phi, v_a \leq v_b\}$$
2. Let $\Phi' = \Phi$.
3. For each element $(d_i, v_i) \in (\Phi, \psi)$ do:
   3.1. Form the root node $\pi_0^i$ in the form $\pi_0^i\left(\mathrm{p_j}, c_j\right) : \mathrm{p_j} = \alpha^{-1}(\mathrm{d_i}), c_j = 1$.
      3.1.1. $\mathrm{p_j}$ is marked place. $\Omega_0(\mathrm{p_j}) = \mathrm{m_0}, \Omega_0(\mathrm{p_l}) = \varnothing, \mathrm{l} \neq \mathrm{j}$, $c(\mathrm{m_0}) = c_j$
   3.2. Let the list $L$ of objects occurring with $d_i$ be $L = \varnothing$.
   3.3. Fire all enabled transitions $t_l$: $I(t_l) = p_j, 0 \leq l \leq |C|, \beta(t_l) \in \sigma_1$, where $p_j$ is the first component of the node $\pi_0^i$. $\Omega\left(\mathrm{p_j}\right) = \varnothing$.
   3.4. For all output places of transitions$t_l$ form step 3.3, $p_q = \mathrm{O}(t_l)$, create new nodes $\pi_y\left(\mathrm{p_q}, c_q\right) : c_q = c_j f(t_l), p_q = \mathrm{O}(t_l)$ and link them by directed arcs with node $\pi_0^i$. The arc value is set to transition value, $f(t_l)$. $\Omega\left(p_q\right) \neq \varnothing$. All new nodes are frozen nodes (F).
   3.5. For each frozen node $\pi_y\left(\mathrm{p_q}, c_q\right)$:
      3.5.1. Insert $\alpha\left(\mathrm{p_q}\right)$ to $L$ if $c_q > \epsilon$.
   3.6. Examine the intersection $\Delta = L \cap \varphi', \varphi' = \Phi'_{|D}$:
      3.6.1. If $|\Delta| < \delta$, $\Phi' = \Phi' \setminus (d_i, c(d_i))$
      3.6.2. If $\Delta = \varphi' \setminus d_i$, continue from step 4.

**Output:**

The set $\Phi'$(e) containing elements of $\Phi'$(e) that have passed the consistency check,
The set $\varphi'(e) = \Phi'_{|D}(e)$ containing object labels that have passed the consistency check.

For instance, let the unlabelled image $e$ in Fig. 9, represented with a feature vector **x**, be given for automatic annotation. The multi-label classifiers generate a set of object classes $\mathrm{C_i} \in C$ for a given feature vector **x**, as explained in Section 3.2. For this example, the obtained classification

| Image example: | a  | b  |
|---|---|---|
| Object labels before annotation refinement | *coral, fish*, ***flower*** | ***airplane, cloud,*** *dolphin*, ***sky*** |
| Object labels after annotation refinement | *coral, fish* | *airplane, cloud, sky* |

**Fig. 14.** A positive (a) and negative and (b) example of inconsistency checking of object labels in an annotation refinement procedure.

**Table 3**
Label-based evaluation measure using the proposed two-tier image annotation system and NB for image annotation at the scene level.

| Annotation method | Feature subset | Label-based evaluation measure for scene-level annotation | | |
|---|---|---|---|---|
| | | Precision | Recall | F1 score |
| Second tier of image annotation system | Object labels | **0.68** | 0.59 | **0.63** |
| NB | All | 0.40 | **0.69** | 0.5 |
| | GIST | 0.28 | 0.68 | 0.39 |
| | DC+CM | 0.22 | 0.64 | 0.32 |
| | DC1..DC5 + CM | 0.28 | 0.68 | 0.39 |
| | DC+ DC1.. DC5 + CM | 0.27 | 0.66 | 0.37 |

| Image example: |  |  |  |  |
|---|---|---|---|---|
| First tier | *elephant, trees, ground* | *airplane, sky, grass* | *water. dolphin* | *water, trees, sky, cloud* |
| Second tier | *SceneElephant, Wild Life Scene, Natural Scene* | *SceneAirplane, Vehicle Scene, Man Made Object* | *SceneDolphin, Sea Scene, Natural Scene* | *Seaside, Natural Scene* |

**Fig. 15.** Examples of two-tier image annotation.

result in the first tier of the automatic image annotation system is $\Phi(e)=\{(sky,0.8),(airplane,0.6),(tree,0.7),(coral,0.2)\}$. Note that the label *coral* is a result of misclassification because it is not present in the image.

The initial refined annotation set is identical to the set $\Phi(e)$. During the execution of the algorithm, the detected intruders will be removed.

It is assumed that the object with the highest confidence corresponds to the context, so the object label that has the lowest truth value, in this case (*coral*, 0.2), is analysed first in order to verify whether there is a *occurs_with* relationship defined between that object class and other object classes in $\Phi(e)$. The root node $\pi_0^1(p_7,1)$of the inheritance tree is formed using the function $\alpha^{-1}$ to find the corresponding place, $\alpha^{-1}(\text{coral})=p_7$ and the confidence value is set to 1. The place $p_7$ is marked, so the initial token distribution is $\Omega_0(p_7)=m_0,\Omega_0(p_l)=\varnothing,l\neq 7$. The list of objects $L$ occurring with coral is initially empty.

Then, the inference tree is formed by running the steps 3.3–3.4. The resulting inheritance tree for *coral* is shown in Fig. 10.

After the inference tree is formed, the set $L$ contains all objects in terminal and frozen nodes for which *coral* has the *occurs_with* relation defined with confidence value greater than $\epsilon=0.05$, $L=water,fish,sky$. To determine if *coral* is consistent with other objects from the set $\Phi(e)$, i.e. $\Phi_{|D}/coral$, the intersection $\Delta$ is examined. In this case, the intersection is $\Delta=L\cap\Phi'_{|D}=water,fish,sky\cap\{sky,airplane,trees\}=sky$. There is only one element in $\Delta$, less than the threshold value $\delta=1.5$, so it is conclude that the object *coral* does not belong to the context and is considered an intruder and removed from the extended annotation set $\Phi'(e)$. The most reliable case for declaring an object intruder is in the case when the intersection $\Delta$ is empty, meaning that the object $d_i$ cannot appear together with any other object from $\Phi'(e)$.

The steps 3.1–3.6 are repeated for the rest of the elements in the set $(\Phi,\psi)$. For example, the inheritance tree for *sky* is shown in Fig. 11. The corresponding intersection is $\Delta=L\cap\Phi'_{|D}=airplane,\ lion,\ldots,water\cap\{airplane,trees\}=airplane,trees$, so it concluded that sky is consistent with other objects in $\Phi'$. The more elements are in the intersection $\Delta$, the greater is the reliability that the object $d_i$ fits the context of the image.

Since the intersection $\Delta$ is equal to $\Phi'_{|D}/sky$, the consistency checking for the remaining objects in $\Phi'$ is not necessary and the algorithm stops. The final refined annotation set is thus $\Phi'(e)=\{(sky,0.8),(airplane,0.6),(tree,0.7)\}$.

## 7. Inference based scene classification

The procedure of further image annotation for the image $e$ in the second tier uses the refined annotation set $\Phi'(e)$ as feature for scene recognition. Unlike the specific objects that can be well characterized with global and regional low-level features, it is very difficult to recognise higher-level concepts that do not have specific low-level features using ordinary classification algorithms. These kinds of concepts can be effectively represented by the defined fuzzy-knowledge scheme KRFPNs capable of handling imprecise information about the domain and the relationships between objects and scenes.

For the task of scene classification of a new, unknown image, an inference based algorithm for scene classification is defined. The algorithm uses the set $\Phi'$ containing object labels that have passed the consistency check and the corresponding confidence values as input.

The steps of the inference based algorithm for scene classification are:

**Algorithm 4.** Inference based algorithm for scene classification.

**Input:**

The set $\Phi'(\mathrm{e})=\Phi'(d_i,v_i)\subset\varphi'(e)\times[0,1]\subset D\times[0,1], i=1,2,\ldots$
Depth of search $k$.
Elements of the KRFPN scheme

**Steps:**

1. Let $\boldsymbol{Z}=(Z_1,Z_2,\ldots,Z_{|SC|})=0_{1\times|SC|}$ be the vector of scene recognition scores.
2. For each element $(d_i,v_i)\in\Phi'(e)$ do:
   2.1. Form the root node $\pi_0^i$ in the form $\pi_0^i\left(\mathrm{p_j},c_j\right):\mathrm{p_j}=\alpha^{-1}(\mathrm{d_i}),c_j=v_i$.
      2.1.1. $\mathrm{p_j}$ is marked place $\Omega_0(\mathrm{p_j})=\mathrm{m}_0,\Omega_0(\mathrm{p_l})=\varnothing,\mathrm{l}\neq\mathrm{j}$, $c(\mathrm{m}_0)=c_j$
      2.1.2. Let $\boldsymbol{z^i}=(z_1^i,z_2^i,\ldots,z_{|SC|}^i)=0_{1\times|SC|}$
      2.1.3. Let the list of frontier nodes $F=(\pi_0^i)$
   2.2. Let $\pi_x\left(\mathrm{p_p},c_p\right)=\pi_0^i\left(\mathrm{p_j},c_j\right)$.
   2.3. Examine the node $\pi_x$:
      2.3.1. If there are no enabled transitions for the place $\mathrm{p_p}$ of the node $\pi_x$, $\pi_x$ is a terminal node (T); continue from step 2.8.
      2.3.2. If $\pi_x$ lies at the k-th level of the recognition tree, $\pi_x$ is a k-terminal node (k-T); continue from step 2.8.
      2.3.3. If $\alpha\left(\mathrm{p_p}\right)\in\mathrm{SC}$, where $\mathrm{p_p}$ is the first component of the node $\pi_x$, then $z_p^i:=max(z_p^i,c_p)$ , where $c_p$ is the second component of $\pi_x$.
   2.4. Remove $\pi_x$ from $F$.
   2.5. Fire all enabled transitions $t_l: I(t_l)=p_p, 0\le l\le|SC|, \beta(t_l)\notin\sigma_1$, where $p_p$ is the first component of the node $\pi_x$. $\Omega\left(\mathrm{p_p}\right)=\varnothing$.
   2.6. For all output places of transitions$t_l$ form step 2.5, $p_q=\mathrm{O}(t_l)$, create new nodes $\pi_y\left(\mathrm{p_q},c_q\right)\big):c_q=c_p f(t_l),p_q=\mathrm{O}(t_l)$ and link them by directed arcs with node $\pi_x$. The arc value is set to transition value, $f(t_l)$. $\Omega\left(p_q\right)\neq\varnothing$.
   2.7. Add the new nodes to the list of frontier nodes $F$.
   2.8. If the list $F$ is not empty, pick the first element in F as the node $\pi_x$; continue from step 2.3.
   2.9. $\boldsymbol{Z}=\boldsymbol{Z}+\boldsymbol{z^i}$
3. Define the set $I^*=\left\{i_h^*:i^*=arg\max_{i=1,\ldots,|SC|}Z_i\right\},1\le h\le|SC|$
4. Define the set $\varphi''(e)=\left\{\alpha\left(p_{i_h^*}\right)\right\}$. The elements of the set $\varphi''$ are scenes from $SC$ that best match the given set $\Phi'(\mathrm{e})$.

**Output:**

The set $\varphi''(\mathrm{e})$ containing the concepts from $\mathrm{SC}\subset\mathrm{D}$ that best match the elements from the set $\Phi'(\mathrm{e})$.

The procedure will be illustrated using an example image $e$ depicted in Fig. 8. The object classes with the corresponding confidence values obtained by object classification and after inconsistency checking in the first tier are contained in the set $\Phi'(\mathrm{e})=\{(sky, 0.8), (tree, 0.7), (airplane, 0.6)\}$. The goal is to find the most likely scene for the given set $\Phi'(\mathrm{e})$. A vector $\mathbf{Z}$ is formed to represent the recognition score of each scene for the given set $\Phi'(\mathrm{e})$, so that each of its components corresponds to a scene from $SC$. At the beginning, the vector $\mathbf{Z}$ is initialised to zero. In each iteration of the Algorithm 4 inference trees are formed for each object in $\Phi'(\mathrm{e})$. The values of $\mathbf{Z}$ are increased in each iteration by scene recognition scores considering only the current object from $\Phi'(\mathrm{e})$. At the end of the algorithm, the components of the vector $\mathbf{Z}$ contain the scene recognition scores for each scene in the whole set $\Phi'(\mathrm{e})$, with the highest scores representing the most likely scenes. In the first iteration in this example, the root node for $(sky,0.8)\in\Phi(e)$ is formed by first determining the corresponding place of the object $sky$ using the function $\alpha^{-1}$: $\alpha^{-1}(sky)=\mathrm{p}_{20}$. The place $\mathrm{p}_{20}$ is marked, so the initial token distribution is $\Omega_0(\mathrm{p}_{20})=\mathrm{m}_0,\Omega_0(\mathrm{p_l})=\varnothing,\mathrm{l}\neq20$, and the corresponding token value is $c(m_0)=0.8$. The formed root node is thus $\pi_0^1(\mathrm{p}_{20},0.8)$ and the initial list of frontier nodes $F$ is $F=(\pi_0^1(\mathrm{p}_{20},0.8))$. The vector $\boldsymbol{z}^1$ of scene recognition scores considering the object $sky$ is initialised to zero.

The root node is also the first frontier node and is thus examined first. The node has enabled transitions and is below the limited depth of search $k$, so it is neither terminal nor k-terminal. The concept $\alpha(p_{20})=sky$ is not the element of the set of scenes SC, so it does not contribute to

the values of $\boldsymbol{z}^1$. All enabled transitions from the place $p_{20}$ of the current node are fired, and new nodes $\pi_1^1$(p32, 0.296), $\pi_2^1$(p29, 0.568), …, $\pi_{10}^1$ (p40, 0.048), $\pi_{11}^1$(p31, 0.056) connected with arcs to the current node are created. The places of the new nodes are the output places of transitions whose input place is $p_{20}$. The node values are calculated as the product of the parent node value and the value of transition of that node, e.g. for the node $\pi_1^3$(p40, 0.048) the value is obtained as $c_p f(t_{57}) = 0.8 * 0.06 = 0.048$. This point of the algorithm corresponds to the root (k=0) and first level (k=1) of the recognition tree shown in Fig. 12. a). The newly created nodes are inserted while the current node is removed from the list of frontier nodes *F*. By repeating the steps 2.3–2.8 of the Algorithm 4 until there are no more frontier nodes in list *F*, the inference tree for the current element (*sky*, 0.8) ∈ Φ(*e*) is constructed, Fig. 12. a). For each node of the inference tree that has a place that belongs to the set of scenes *SC*, the component of the vector $\boldsymbol{z}^i$ corresponding to the same scene is increased by that node value. However, if more nodes have the same place, only the maximum node value is retained in $\boldsymbol{z}^i$. For the object *sky*, the components of the resulting vector $\boldsymbol{z}^1$ are $z_1^1 = 0$, $z_2^1 = 0$, …, $z_{29}^1 = 0.568$, $z_{30}^1 = 0.184$,…, $z_{50}^1 = \max(0.104, 0.48) = 0.48$,…, $z_{54}^1 = 0.568$, …. The values of the vector $\boldsymbol{z}^1$ are then added to the vector **Z.** The step 2 is repeated for the remaining elements of Φ(*e*), (tree, 0.7) and (airplane, 0.6), starting from the root nodes $\pi_0{}^i, i = 1, ..., 3$,yielding trees shown in Fig. 12b) and c). The values of $\boldsymbol{z}^i$ are accumulated into **Z**, so the obtained vector **Z** represents the ranking of scene classes according to the scores obtained by the algorithm for scene classification. The components of the vector **Z** with the highest values, determined in the step 3, correspond to the most likely scenes for the image *e*.

If there are several components of the vector Z with the same maximum value, more than one scene class can be assigned to the image. In this example, the components $Z_{29}, Z_{52}$ and $Z_{54}$ of the vector Z have the same maximum value 1.292, so scene classes *α(p29) = SceneAirplane, α(p52) = VehicleScene* and *α(54) = TransportScene* are chosen as the best match. The annotation set $\varphi'(e)$ for the image *e* at the second tier is thus $\varphi'(e) = SceneAirplane\ , VehicleScene, TransportScene$.

By merging the labels that are so far associated with the image, a two-tiered annotation of the image is formed. The scene recognition scores in Z, obtained with the proposed inference algorithm, have a different range than the confidence values of object classes in Φ(e), so the final annotation set is formed by joining only labels from the first and second tier. For example, for the image *e* (Fig. 8), the annotation set is $\varphi'''(e) = \varphi'(e) \cup \varphi''(e) = \{sky, airplane, tree\} \cup SceneAirplane, VehicleScene, TransportScene$.

More abstract concepts can be inferred using the hierarchical relationships among the scene concepts. The number of hierarchical levels can be changed, and, if necessary, a new KRFPNs 2 scheme with more abstract concepts can be defined and connected with the KRFPNs scheme we already use, Fig. 13. To enable uninterrupted inference through a hierarchical structure of schemes, the output nodes of the lower scheme should correspond to the input nodes of the scheme at the higher level of abstraction.

## 8. Evaluation of the proposed two-tier model

We compared the results of the proposed two-tier automatic image annotation system at object and scene levels with previously published results.

The performance of the annotation in the first tier refers to multi-label classification at object level and is evaluated using different subsets of features. The obtained object labels are refined using the proposed annotation refinement algorithm, and the impact of that process is analysed at the scene level where the objects are treated as features for scene inference. The performance of the inference-based scene annotation in the second tier is also compared with scene classification with the ordinary classification approach using the Naïve Bayes classifier with the same sets of features that were used in the first tier for object classification. The achieved results are averaged over 3 runs, since 3-fold cross validation was used. The classification performance at the object level was measured in terms of instance-based and label-based accuracy, precision, recall and F1 score [29], while the classification performance at the scene level was measured using the label-based precision and recall metrics.

### 8.1. Evaluation measures

The instance-based evaluation measures are based on the average differences of the actual and the predicted sets of labels over all examples in the test dataset. The label-based evaluation measures assess the predictive performance for each label separately and then average the performance over all labels [29]. These measures are used since an image may not only be correctly or incorrectly annotated, but also partially correctly annotated in the case of multi-label classification. For example, if an image should be annotated with the object labels *grass, sky, wolf*, and is automatically annotated with *tree, sky, dog, cloud,* then the evaluation measure should reflect the insertion of wrong labels (*tree, dog, cloud*), missing labels (*wolf, grass*) and correct labels (*sky*).

To define the evaluation measures, we assume that an image $e_j \in E, j = 1..N$ should be classified into the set of true object labels $Y_j = \{C_l, C_m, \ldots, C_r\}$, $Y_j \subseteq C$, where *E* is a set of images, *C* is a set of all class labels and $N = |E|$ corresponds to the number of images in the set *E*. For an example image $e_j$, the set of labels that are predicted by a classifier is denoted as $\varphi_j$.

Instance-based accuracy is defined as the average ratio of correctly assigned labels and all labels assigned to each example by the classifier and the true labels:

$$Accuracy_{ins} = \frac{1}{N}\sum_{i=1}^{N}\frac{|Y_i \cap \varphi_i|}{|Y_i \cup \varphi_i|}.$$

Instance-based precision is defined as the average ratio of correctly assigned labels and all labels assigned to each example by the classifier:

$$Precision_{ins} = \frac{1}{N}\sum_{i=1}^{N}\frac{|Y_i \cap \varphi_i|}{|\varphi_i|}.$$

Instance-based recall is defined as the average ratio of labels correctly assigned by the classifier and all labels in the ground truth for each example:

$$Recall_{ins} = \frac{1}{N}\sum_{i=1}^{N}\frac{|Y_i \cap \varphi_i|}{|Y_i|}.$$

The instance-based F-Measure is the harmonic mean of precision and recall:

$$F1_{ins} = \frac{1}{N}\sum_{i=1}^{N}\frac{2|Y_i \cap \varphi_i|}{|\varphi_i| + |Y_i|}.$$

These measures reach their best value at 1 and the worst value at 0.

Label-based measures are computed by averaging instance-based measures over all labels. The case of scene-level annotation in the second tier is treated as single-label classification, $\left|Y_j\right| = \left|\varphi_j\right| = 1$, and label-based measures are used.

### 8.2. Data

To evaluate the proposed two-tier model for image annotation, a part of the Corel image dataset related to outdoor scenes [2] was used.

The features were extracted from images that were sized $128 \times 192$ pixels or $192 \times 128$ pixels. We used images labelled with one or more labels from the set $C$ of 54 object classes related to natural and artificial objects such as *Airplane*, *Bird*, *Lion*, *Train*, etc., and background objects like *Ground*, *Sky*, *Water*, etc., provided by [10]. Additionally, we labelled the images with one of the 20 elements from the set *SC* related to outdoor scenes such as *SceneTrain*, *SceneLion*, *Inland,* etc.

Some labels were too rare to effectively train the classifier, and images that correspond to those labels were excluded from the data, leading to the number of object and scene labels to be reduced and the average number of images for object and scene labels to be increased. The resulting simplified dataset is more suitable for the learning of classification models. The details of the dataset before and after simplification are presented in Table 1.

### 8.3. Annotation results

The label-based and instance-based evaluation results for object-level annotation in the first tier of the proposed system are presented in Table 2, where different feature subsets and different classifiers (RAKEL, ML-kNN and NB along with the data transformation) are considered. Overall, the best results considering the label-based F1 score are obtained using the NB classifier independently of the used feature subset, due to significantly better achieved recall than with other methods. However, in instance-based measures NB obtained the worst results with both precision and recall. The RAKEL-C4.5 performed best in instance-level measures but performed worse than NB in label-based measures. For both instance-based and label-based measures, the RAKEL-kNN achieved slightly lower results than the best classifier. The RAKEL-C4.5 also performed well with all subsets of features, although their dimensions varied between 48 in the case of dominant colours and 740 when all features were used.

The achieved results are better than the published results for 28 object classes, in the same dataset [13,16,4]. In [4], the authors reported average precision for the task of automatic image annotation at object level achieved with the dMRF model based on Markov random fields defined in [4] and the dInd translation model from [10]. The dMRF model achieves a label-based precision of 21%, while the dInd model achieves a label-based precision of 20%. When the original framework based on Fuzzy Petri Nets was used for image annotation at object level, the achieved label-based precision rate was 37%, while the label-based recall rate was 48% [13], but with additional expert involvement in building the models. With the more straightforward classical machine-learning approach using the same feature set and the Naïve Bayes classifier using image segments, somewhat lower scores were achieved, with label-based precision of 32.6% and recall of 27.5% [16].

Considering that the results of automatic image annotation in the first tier can include object labels that do not correspond to the context of an image, the obtained annotation set is refined using the proposed inference based annotation refinement algorithm and the facts from the knowledge base related to the co-occurrence relationships between objects. In the proposed model, objects that do not fit the likely context of the image are discarded to reduce the propagation of errors in the second tier. As a consequence, the average precision of the image annotation can be increased if there is only one intruder class among the object-level annotation as in Fig. 14(a), where the *flower* is the intruder class. In Fig. 14(b), the majority of labels for an image are wrong (*airplane*, *cloud*, *sky*), so the annotation refinement procedure discards the true label (*dolphin*), and the precision falls.

Automatic image annotation in the second tier of the proposed model is performed by the inference-based scene classification algorithm (Algorithm 4) given that the facts about the domain are known. The input to the algorithm is the refined annotation set comprising object classes obtained using RAKEL-kNN and all features. The obtained label-based precision of scene-level annotation is 68% and the recall is 59% (F1 score 58%). The results depend on the results of the first tier that are used as input. For those scenes for which there is one main object class which is highly discriminant for that scene (e.g. *train* for *SceneTrain)*, it is crucial to detect that object in the first tier. In this kind of scene, background objects that are common to most scenes do not play an important role, but in scenes without one prominent object (e.g. *Sea*, *Inland*) they are important. For example, in the case of object-level annotation, the best F1 score is obtained for train (0.8), tracks (0.77) and *polarbear* (0.68), and the worst for *wolf* (0.07) classes. This is reflected in the scene-level classification, where *SceneTrain* has the best precision (0.86) and *SceneWolf* among the worst (0.30). For background objects, the best F1 scores are for *sky* (0.65) and *grass* (0.66) and the worst F1 for *mountain* (0.11) and *clouds* (0.13). Differences in results at the object level may be due to an imbalance in the number of examples per class and the fact that in case of multi-label classification, partially correct annotation is possible. In the case of similar classes, e.g. *lion* and *tiger*, *cloud* and *sky* in multi-label classification, both labels can be assigned.

The obtained results on second tier of the proposed annotation system outperform the results on the scene-level annotation obtained on the same data set and published in [16] where knowledge-based approach of automatic image annotation was combined with ordinary classification of segmented images. This was to some extent been expected, since the obtained results on the object level with multi-label

classification were better and have been used as inputs to the inference algorithms for scene classification. The part of the improvement in results can also be attributed to using of an m-estimate of probability when modelling the confidence of relations and weighting of the computed probabilities, to make accumulated knowledge more representative and less dependent on the examples in the data set.

For comparison with ordinary classification approach, scene-level annotation was also performed using the Naive Bayes classifier and the same feature sets that were used for object classification. The evaluation measures for both the proposed two-tier image annotation system and for the Naive Bayes are presented in Table 3.

The best classification results for Naive Bayes were obtained using the whole feature set, giving a precision of 40%, a recall of 69% and an F1-score of 50%. It can be noted that for the Naive Bayes classifier 36 features (DC1..DC5 + CM) performed in a similar manner to the GIST descriptor with 512 elements, which suggests that dominant colours contain important information about scene classes. The best overall performance considering the F1 score was obtained using the proposed two-tier image annotation system, thanks to much higher precision and despite somewhat reduced recall. The effects of the knowledge-based annotation refinement step, which can reduce the error propagation from the object classification, although in some cases reducing the useful information available for the inference of scenes, can explain such results. Fig. 15 shows examples of image annotation obtained by the proposed model.

## 9. Conclusion

The aim of this paper has been to present a two-tier image annotation model where the first tier corresponds to object-level and the second tier to scene-level annotation. The tiers are defined to separate different levels of image interpretation, one that is very dependent on the visual content and the other which contains knowledge about the domain in general. In the first tier, images are annotated with labels of objects present in them, using multi-label classification methods for low-level features extracted from images. In the second tier, the fuzzy-knowledge-representation scheme based on the Fuzzy Petri Net (KRFPN) is proposed to represent knowledge about the image domain and to support reasoning about scenes and objects. Two data-driven algorithms are proposed to automatically extract knowledge about the domain from the training data set. The first is used to define relationships between objects and the corresponding confidence values, and the second defines the scenes as compositions of objects and sets the corresponding confidence values. To make the accumulated knowledge more general and less dependent on the examples in the data set, especially when some of the data are incomplete or missing or when training set is not large enough, the computed probabilities are weighted and an m-estimate of probability is introduced when modelling the confidence of relations. This increases the quality of acquired knowledge and enables addressing the annotation of images from a broader domain. If automatically generated facts and rules are not representative enough or if their reliability is not properly set, an expert can review or modify rules or add new ones to improve the knowledge base completeness and accuracy, as well as to enhance the inference process.

Inference-based algorithms are defined for refinement of annotation set obtained with multi-label classification in the first tier and for automatic scene classification in the second tier. The algorithms draw conclusions using stored fuzzy knowledge.

Specifically, the proposed inference based annotation refinement algorithm provides reasoning about consistency of object classes to detect misclassified object labels and to exclude them from the annotation set, and the proposed inference based scene classification algorithm provides inference about relationships between scene classes and their object components for scene classification. Consistency checking of object labels can improve the precision of object-level annotation and reduce the propagation of errors through the hierarchical structure of concepts during the inference of scenes. More abstract scene concepts can be inferred with the same inference-based scene classification algorithm if domain knowledge is structured at varying levels of generality and if appropriate relations between concepts are defined. However, an expert should provide this kind of relations in the scheme, as they cannot be automatically generated from training data.

The proposed inference-based algorithms rely on the execution of the Petri nets and utilise the concept of finite inference trees. Their complexity is $O(nm)$, where $n$ is the number of places (concepts) and $m$ is the number of transitions (relations) in the scheme.. In practice, the whole net is not usually used for inference, but only parts that can be reached from the initially marked places. During the execution of the algorithm, the generation of inference trees can be manually limited to a predefined level or appropriate conditions can be defined to stop further generation of tree.

The proposed two-tier annotation model is adaptive and each tier can be independently used, modified and improved, so the proposed annotation refinement and scene classification algorithms can be used regardless of the classification method used in the first tier. Therefore, different types of classification methods along with different subsets of features were tested on an outdoor image dataset in the first tier, to get the best possible result since these results influence the results on the second tier as well as the overall annotation results.

The feature subsets used in the first tier are composed of dominant colours, image moments, and GIST descriptors, and were tested with the RAKEL, ML-kNN and Naïve Bayes classification methods. The obtained results are better than those already published and related to the same set of images and so are more useful as inputs to the proposed inference-based algorithms in the second tier.

This research was focused on developing the annotation system for images within the domain of outdoor scenes. For images from other domains, it is necessary to collect data that will be used to extract knowledge and to model a fuzzy knowledge representation scheme that supports reasoning about concepts in new domain.

The proposed algorithms for automatic acquisition of knowledge and for reasoning about objects and scenes are defined in a general manner and probably minor adjustments would be sufficient for use in other tasks. In the future, we plan to adapt the proposed knowledge representation scheme and the supporting algorithms for use in video annotation, by including the temporal dimension into the model.

## Conflict of interest

None declared.

## Acknowledgement

This work has been fully supported by Croatian Science Foundation under the project 6733 De-identification for Privacy Protection in Surveillance Systems (DePPSS), Grant no: 6733.

## References

[1] T. Athanasiadis et al., Integrating image segmentation and classification for fuzzy knowledge-based multimedia, in: Proceedings of the MMM 2009, France, 2009.
[2] K. Barnard, P. Duygulu, D. Forsyth, N. Freitas, D.M. Blei, M.I. Jordan, Matching words and pictures, J. Mach. Learn. Res. 3 (2003) 1107–1135.
[3] A. Binder, W. Samek, K.-R. Müller, M. Kawanabe, Enhanced representation and multi-task learning for image annotation, Comput. Vis. Image Understanding 117 (5) (2013) 466–478.
[4] P. Carbonetto, N. Freitas, K. de Barnard, A statistical model for general contextual object recognition, in: Proceedings of the ECCV 2004, Czech Republic, May 2004, pp. 350–362.
[5] S.M. Chen, J.S. Ke, J.F. Chang, Knowledge representation using fuzzy Petri nets, IEEE Trans. Knowl. Data Eng. 2 (3) (1990) 311–319.
[6] R.L. Cilibrasi, P.M. Vitanyi, The google similarity distance, IEEE Trans. Knowl. Data Eng. 19 (3) (2007) 370–383.
[7] R. Datta, D. Joshi, J. Li, Image retrieval: ideas, influences, and trends of the new age, ACM Trans. Comput. Surv. 20 (2008) 1–60.
[8] J. Deng, W. Dong, R. Socher, L.J. Li, K. Li, L. Fei-Fei, ImageNet: a large-scale hierarchical image database, in: Proceedings of thec IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009, 2009, pp. 248–255.
[9] A. Deruyver, Y. Hodé, L. Brun, Image interpretation with a conceptual graph: Labeling over-segmented images and detection of unexpected objects, Artif. Intell. 173 (2009) 1245–1265.
[10] P. Duygulu, K. Barnard, J.F.G. de Freitas, D.A. Forsyth, Object recognition as machine translation: learning a lexicon for a fixed image vocabulary, in: Proceedings of the European Conference on Computer Vision, 2002, pp. 97–112.
[11] S. Džeroski, B. Cestnik, I. Petrovski, Using the m-estimate in rule induction, J. Comput. Inf. Technol. 1 (1993) 37–46.
[12] J.S. Hare, P.H. Lewis, P.G.B. Enser, C.J. Sandom, 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, 2006.
[13] M. Ivasic-Kos, S. Ribaric, I. Ipsic, Low-and high-level image annotation using fuzzy petri net knowledge representation scheme, Int. J. Comput. Inform. Syst. Ind. Manag. (IJCISIM) 4 (2012) 428–436.
[14] M. Ivasic-Kos, I. Ipsic, S. Ribaric, Multi-level image classification using fuzzy Petri Net., in: Proceedings of the International Conference on Neural Networks-Fuzzy Systems (NN-FS'14), Venice, Italy, 2014a, pp. 39–45.
[15] M. Ivasic-Kos, M. Pobar, I. Ipsic, Multi-layered image repre sentation for image interpretation, VL Net 2014, pp. 115–117.
[16] M. Ivasic-Kos, I. Ipsic, S. Ribaric, A knowledge-based multi-layered image annotation system ISSN 0957-4174, Expert Syst. Appl. 42 (24) (2015) 9539–9553.
[17] J. Li, J.Z. Wang, Real-time computerized annotation of pictures, IEEE Trans. Pattern Anal. Mach. Intell. 30 (2008) 985–1002.
[18] J. Liu, M. Li, Q. Liu, H. Lu, S. Ma, Image annotation via graph learning, Pattern Recognit. 42 (2) (2009) 218–228.
[19] Jing Liu, Bin Wang, Hanqing Lu, Ma, Songde, A graph-based image annotation framework, Pattern Recognit. Lett. 29 (4) (2008) 407–415.
[20] G. Madjarov, D. Kocev, D. Gjorgjevikj, S. Džeroski, An extensive experimental comparison of methods for multi-label learning, Pattern Recognit. 45 (9) (2012) 3084–3104.
[21] V. Mezaris, G.T. Papadopoulos, A. Briassouli, I. Kompatsiaris, M.G. Strintzis, Semantic Video Analysis and Understanding. In: Mehdi Khosrow-Pour (Ed.), Encyclopedia of Information Science and Technology, Second ed., IGI Global, Hershey, PA, USA, 2009.
[22] F. Monay, and D. Gatica-Perez, On image auto-annotation with Latent Space Models, in: Proceedings of the ACM Multimedia, Berkeley, CA, 2003, pp. 275–278.
[23] A. Oliva, A. Torralba, Modeling the shape of the scene: a holistic representation of the spatial envelope, Int. J. Comput. Vis. 42 (3) (2001) 145–175.
[24] J.Y. Pan, H.J. Yang, C. Faloutsos, P. Duygulu, Gcap, Graph-based automatic image captioning, Computer Vision and Pattern Recognition Workshop, 2004 CVPRW'04, pp. 146–146.
[25] J.L. Peterson, Petri Net Theory and Modeling of Systems, Prentice-Hall, Upper Saddle River, NJ, USA, 1981.
[26] S. Ribarić, N. Pavešić, Inference procedures for fuzzy knowledge representation scheme, Appl. Artif. Intell. 23 (2009) 16–43.
[27] A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, R. Jain, Content-based image retrieval at the end of the early years, IEEE Trans. Pattern Anal. Mach. Intell. 22 (12) (2000) 1349–1380.
[28] G. Tsoumakas, I. Vlahavas, Random k-label sets: an ensemble method for multi-label classification, in: Machine Learning ECML, 2007, Springer, pp. 406–417.
[29] G. Tsoumakas, I. Katakis, Multi-label classification: an overview, Int. J. Data Warehous. Min. 3 (3) (2007) 1–13.
[30] A.M. Tousch, S. Herbin, J.Y. Audibert, Semantic hierarchies for image annotation: a survey, Pattern Recognit. 45 (1) (2012) 333–345.
[31] C. Wang, F. Jing, L. Zhang, H.-J. Zhang, Image annotation refinement using random walk with restarts, in: Proceedings of the ACM MM 06', Santa Barbara, California, USA, October 23–27, 2006, pp. 647–650.
[32] Y. Wang, S. Gong, Refining image annotation using contextual relations between words, in: Proceedings of the ACM CIVR 07', Nethelands, July 9–11, 2007, pp. 425–432.
[33] C. Wang, F. Jing, L. Zhang, H.-J, Zhang, Content based image annotation refinement, CVPR'07, 2007.
[34] M.L. Zhang, Z.H. Zhou, ML-KNN: a lazy learning approach to multi-label learning, Pattern Recognit. 40 (2007) 2038–2048.
[35] D. Zhang, M.M. Islam, G. Lu, A review on automatic image annotation techniques, Pattern Recognit. 45 (1) (2012) 346–362.
[36] S. Zhang, Q. Tian, G. Hua, Q. Huang, W. Gao, ObjectPatchNet: towards scalable and semantic image annotation and retrieval, Comput. Vis. Image Understanding 118 (2014) 16–29.

**Marina Ivasic-Kos** received her B.Sc. degree in mathematics and computer science from the Faculty of Philosophy, University of Rijeka and M.Sc. degree in information science from the Faculty of Philosophy, University of Zagreb in 1997 and 2201, respectively. Then, in 2012 she earned her Ph.D. in Computer Science at the Faculty of Electrical Engineering and Computing in Zagreb. Since 1998 she is working at the Department of informatics, University of Rijeka, now as Assistant Professor for computer science courses. She has participated in several business and research projects. Her current research interests belong to the field of pattern recognition, computer vision and knowledge representation. Her research work has been presented at several international scientific conferences and in journals.

**Miran Pobar** graduated in electrical engineering from the Faculty of Engineering, University of Rijeka in 2007. In 2008 he started his doctoral studies in Computer science at the Faculty of Electrical Engineering and Computing in Zagreb. In 2008. he joined the Department of Informatics at the University of Rijeka where he works as teaching assistant in the field of Computer science. His research interests are focused on artificial intelligence and speech technologies.

**Slobodan Ribaric** is a Full Professor at the Department of Electronics, Microelectronics, Computer and Intelligent Systems, Faculty of Electrical Engineering and Computing, University of Zagreb. His research interests include Pattern Recognition, Computer Architecture, Knowledge Representation and Biometrics. He has published more than one hundred and forty papers on these topics (http://bib.irb.hr), and he is author of six books and co-author of one (An Introduction to Pattern Recognition). Professor Ribaric was involved in COST Action 275 “Biometrics on the Internet” and Network of Excellence FP6 “Biosecure”. He was a leader of Working group 2 of COST Action 2101 “Biometrics for Identity Documents and Smart Cards”.