Improving Concept-Based Image Retrieval with Training Weights Computed from Tags

VASILEIOS PAPAPANAGIOTOU, Aristotle University of Thessaloniki
CHRISTOS DIOU, Aristotle University of Thessaloniki
ANASTASIOS DELOPOULOS, Aristotle University of Thessaloniki

This paper presents a novel approach to training classifiers for concept detection using tags and a variant of Support Vector Machines that enables the usage of training weights per sample. Combined with an appropriate tag weighting mechanism, more relevant samples play a more important role in the calibration of the final concept-detector model. We propose a complete, automated framework that (i) calculates relevance scores for each image-concept pair based on image tags, (ii) transforms the scores into relevance probabilities and automatically annotates each image according to this probability, (iii) transforms either the relevance scores or the probabilities into appropriate training weights and finally, (iv) incorporates the training weights and the visual features into a Fuzzy Support Vector Machine classifier to build the concept-detector model. The framework can be applied to online public collections, by gathering a large pool of diverse images, and using the calculated probability to select a training set and the associated training weights. To evaluate our argument, we experiment on two large annotated datasets. Experiments highlight the retrieval effectiveness of the proposed approach. Furthermore, experiments with various levels of annotation error show that using weights derived from tags significantly increases the robustness of the resulting concept detectors.

CCS Concepts: •**Information systems** → **Image search;** *Uncertainty;* •**Computing methodologies** → **Ranking;**

Additional Key Words and Phrases: concept-based image retrieval, fuzzy support vector machine, weighted training sample

ACM Reference Format:
Vasileios Papapanagiotou, Christos Diou and Anastasios Delopoulos, 2015. Improving Concept-Based Image Retrieval with Training Weights Computed from Tags. *ACM Trans. Multimedia Comput. Commun. Appl.* 0, 0, Article 0 (2015), 23 pages.
DOI: http://dx.doi.org/10.1145/2790230

1. INTRODUCTION

Concept-based image retrieval aims at enabling indexing and subsequent retrieval of images based on *concepts* that are automatically detected from the visual content of images, as well as from any accompanying metadata [Zhang and Rui 2013]. Examples of concepts include image scene elements (“sky”, “sea”), actions (“person running”, “smiling face”) or objects (“car”, “flower”). The use of concepts allows textual queries on non-annotated image collections.

Indexing of images using concepts requires the mapping of low-level features that are extracted from visual data to high-level features (concepts) that are directly per-

Authors’ address: V. Papapanagiotou and C. Diou and A. Delopoulos, Electrical and Computer Engineering Dpt., Aristotle University of Thessaloniki, GR-54124, Greece.
 1551-6857/2015/-ART0 $15.00
DOI: http://dx.doi.org/10.1145/2790230

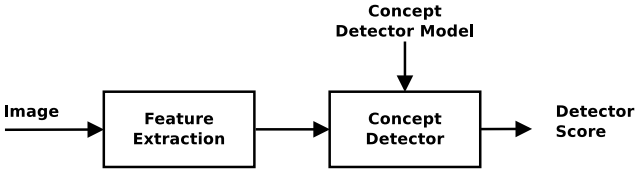


Fig. 1: Concept detection on an image. The image is visually processed and a feature vector is extracted. A concept detector algorithm uses the image feature vector and a trained concept detector model to calculate a score that indicates the detector estimation of concept relevance for the image.

ceived by the searchers. This is typically achieved with the processing pipeline depicted in Figure 1 that ranks images with respect to each concept.

Several types of low-level features have been proposed in the bibliography [Snoek and Worring 2009] with the "bag of visual words" approach applied on local key-point descriptors [van de Sande et al. 2008] and, more recently, VLAD and Fisher vectors [Habibian and Snoek 2014], achieving remarkable results. For the concept detection stage an array of binary classifiers is used, one for each concept, that produce one detection score per concept for each input feature vector. Typically, concept detectors rely on Support Vector Machines (SVM) and are built via a supervised training procedure: a set of training images, labelled on whether they depict the concept (positive samples) or not (negative samples) are used to "train" the concept detector model that is then used for image ranking. Thus, concept-specific non-antagonistic detector models are produced, and then deployed in image indexing and retrieval.

Effective concept detector training requires a large number of both positive and negative examples. In addition, the diversity of images and the accuracy of labelling are important factors that affect the performance of the detector model. Robust training data can result from manual image labelling, although even for manually generated data different annotators may not agree (e.g. in [Nowak and Rüger 2010] an average of $79,6\%$ inter-annotator agreement is reported per concept). In any case, manual labelling is a laborious task, non-economic both in financial resources and time, especially if maintenance and update requirements are taken into account. For example, enabling retrieval on new concepts requires new training samples. Furthermore, concept models can be domain or application specific, and thus require different training samples for each domain or application. Constantly or periodically updated models also require a stream of new training data. Finally, in order to achieve high performance on the detection procedure, a large volume of training samples is required.

In order to overcome these problems, several approaches have been proposed for automatic or semi-automatic generation of concept training sets [Wang et al. 2007]. These include active and semi-supervised learning, use of click-through data, and use of accompanying textual information. Active learning [Wang and Hua 2011] involves a set of repeated training cycles. At the beginning of each cycle (or loop), some feedback from a user is required, and is used in the following processing steps. Sufficient number of loops can produce satisfying results, at the cost of the higher involvement of the user in the annotation process. Click-through data approaches [Tsikrika et al. 2009; 2010] exploit information provided implicitly by the user to assess the relevance of images to concepts for detector training. Thus, the user provides useful information without actively participating in training set generation. Click-through data, however, is only available to search engines that are already being used, while no click-through data becomes publicly available.

Complementary to these training set generation approaches is the use of tags [Mandel et al. 2011; Tang et al. 2013]. The term *tag* or *user tag* refers to a word or a short phrase that users have assigned to images hosted in publicly available community repositories [Datta et al. 2005] (such as Flickr[1] or Picasa[2]). Tags however are not strictly related to concepts. Concepts are always specific, and may required multiple words, or even complete sentences, to be properly defined. Tags, on the other hand, are usually single words, often ambiguous. Thus, inferring a concept form a tag (or even a set of tags) is not trivial.

Users continuously upload new content, and also provide tags for the uploaded images. It has been noted that users do tend to annotate images, mainly according to their content [Sigurbjörnsson and Van Zwol 2008]. Furthermore, these repositories support a search functionality, where users provide textual queries and the system responds with lists of images with relevant tags. Search using only the concept name as a query is usually not sufficient for collecting training images, however, since results are usually small in quantity, and quite often lack the visual diversity required to adequately describe the concept. A more involved method is therefore preferred for exploring image repositories to harvest useful training material.

In addition, user tags are not always reliable. There exist tags that are related to the image context and not to the directly observable content. Examples include tags that might be capturing device metadata, descriptions of personal feelings or thoughts of the user who assigned them, non existent words, ambiguity and even completely irrelevant or misleading information. In addition, tag descriptions are incomplete, since many of the tags that could describe the image content are missing. All these factors introduce errors in the training set that lead to reduced effectiveness of concept detectors that are produced on the basis of the selected training samples.

Works that rely on online collections for training set generation use various ways to to cope with erroneous tags and increase concept detector resilience to noise. Clustering methods for tags are sometimes used, and concept spaces are created, while other approaches build cross modal spaces that combine both textual and visual information [Yang et al. 2012]. Such methods are summarised in the review work of [Rafiee et al. 2010]. Another major issue for such approaches is the un-tagged image content. As already mentioned, users assign tags based on their judgment, and usually leave out a lot of information they consider irrelevant. In all these cases however, all training samples are equally treated. Assigned labels are binary and crisp, and once their values are decided, images are considered as positive or negative samples with respect to each specific concept.

This work introduces an approach for quantifying and handling label relevance, as well as improving concept detector effectiveness under the presence of label noise. More specifically, we propose an approach for collecting candidate training images from a community repository and the use of a significantly improved variant of the algorithm presented in [Tsirelis and Delopoulos 2011] to assess concept relevance for each image. Subsequently, a new method is presented for transforming the relevance scores into probabilities that are used for automatic image labelling. Once the automatic label assignment based on these probability estimations has been performed, either of relevance scores or probabilities can be used to introduce confidence weights to each training image. These weights represent a level of certainty for the individual label assignment. This information is taken into account during the concept detector training by the Fuzzy SVM. Experimental results show that models created with

[1]https://www.flickr.com
[2]http://www.picasa.com

Fuzzy SVM lead to increased concept detector effectiveness, compared to the original SVM that uses crisp label assignment.

From a user perspective, the proposed work-flow is entirely automated. The only user input required is a set of words, or concept terms, that define the concept; three to six words are usually sufficient to unambiguously define the concept. The output is the concept detector model.

The rest of the paper is organised as follows. Related work is discussed in Section 2. Section 3 presents the proposed approach for weighted training, and the method for producing such training weights. In Section 3.3 we describe the automated framework that implements the proposed method. Section 4 presents various experiments demonstrating improved performance both for the novel tag algorithm, and for the Fuzzy SVMs over conventional ones. Finally, Section 5 summarises the presented work and concludes the paper.

2. RELATED WORK

In this section we identify previous research in two different areas that are related to the work of this paper. We first review the use of Fuzzy SVM [Lin and Wang 2002] for classification, focusing on multimedia retrieval applications, and then present works that use publicly available images along with their user tags to automatically generate training sets and build concept detection models.

The Fuzzy SVM classifier has been used in various works. In [Min and Cheng 2009], a membership function is constructed based on the Euclidean visual distance of images of the training set, and is used in a Fuzzy SVM. The Fuzzy SVM is incorporated in a semi-supervised architecture that requires manually labelled images, and relies on user feedback to balance the fuzzy membership of each image. In contrast, our work requires no manual labelling, and training weights are calculated from user tags instead of internal properties of the dataset.

A combination of several Fuzzy SVMs was used in [Rao et al. 2006] to create a system that simulates user feedback in the form of predefined levels of relevance between images and concepts. A bagging system of Fuzzy SVMs is used to create training weights, which are subsequently used in a final Fuzzy SVM. However, such fuzzy training samples were only partially used (about 20%). Similarly in [Wu and Yap 2008], a relevance feedback method is used to assign soft labels on the output of a first Fuzzy SVM. A membership function is constructed as a product of concept relevance, based on the output of this Fuzzy SVM, and a visual distance ratio from cluster centroids, that have been calculated in an initialisation step using k-NN. The authors of [Lin and Wang 2004] used Fuzzy SVMs with two different strategies (based on kernel alignment and k-NN) for estimating sample fuzziness. Two thresholds are used, an upper threshold for separating completely valid samples and a bottom threshold for noisy ones. Samples lying between those two thresholds are treated as noise with probability values. However, thresholds and other parameters require exhaustive trial and error. Our work differentiates significantly by (a) automatically calculating training weights and labels, and (b) calculating these parametres using external information (tags) of the dataset, instead of features.

A new type of SVM, the Power SVM (PSVM) is introduced in the work of [Zhang and Ye 2009] that produced promising results. The main difference of PSVM with Fuzzy SVM is that the sample weights are used in the constraints of Equation (1) (instead of the objective function). Weight values of training images are calculated using the output of multiple SVMs. Each training sample is characterised by different importance in the training procedure.

Some of the approaches presented above compute the training weights based on the feature vectors of the training set. These methods therefore assume that the training

labels are correct, however there is uncertainty associated with the feature representation (internal uncertainty). Other methods, like [Wu and Yap 2008], rely on relevance feedback provided by users in order to produce the training weights. In contrast, this current paper relies on automatically produced relevance scores of images from their tags, with respect to each concept, in order to compute appropriate training weights for the concept detectors. These relevance scores are produced in terms of user tags, as well as on visual information and represent external uncertainty associated with the training labels.

The Fuzzy SVM has been used on a variety of classification tasks for applications beyond concept-based image retrieval. For example, the work of [Bovolo et al. 2010] uses a Fuzzy Input Fuzzy Output variant of Fuzzy SVM for image subpixel classification. In [Redi and Merialdo 2012], a multimedia framework is proposed, and applied to both video retrieval and scene recognition tasks. It does not employ the Fuzzy SVM, but the conventional SVM to rank the training set, which is subsequently split into three groups and one SVM is trained on each group. Given a query image, the final decision is calculated using a fusion of the three SVM outputs. Finally, the work of [Liu and Zheng 2007] employs Fuzzy SVM for video-object extraction. The weight used at the Fuzzy SVM is a ratio of the object, background and whole image pixel number.

In [Xian 2010; Sohail et al. 2011; Sun et al. 2009], Fuzzy SVM has been used for medical image classification. The work of [Leng and Wang 2008] uses a facial database to predict gender using Fuzzy SVM, while a stochastic SVM was constructed in the work of [Li et al. 2012], and its effectiveness is demonstrated on both artificial and real data. In the field of economics, the bilateral SVM has been proposed [Wang et al. 2005], that treats each training sample as both positive and negative with a fuzzy membership.

Beyond the specific case of Fuzzy SVM for multimedia analysis, a rich literature exists on using publicly available multimedia collections, in order to either generate training samples for concept detectors, or directly rank image/video collections. However these approaches do not account for the noise that is inherent to all automatic training set generation methods.

In [Sang et al. 2012], a personalised retrieval system is proposed, using Flickr tags. Tag similarity is calculated as a weighted sum of two parameters: tag co-occurrence in the collection, and WordNet distance. The importance of each parameter is determined experimentally however. The work of [Zhu et al. 2012] proposes an ontological categorisation of concepts also based on WordNet, to increase relevance, and establishes the Flickr context similarity, as the estimated co-occurrence of tag pairs. In [Ewerth et al. 2012], an automated system is built that periodically updates concept models using the web, as a means to track the changing trends and concepts. It also uses a vocabulary based on the image tags and the surrounding text of the source web-page. Concept detectors are constructed on crisp training sets (without the use of training weights).

In the work of [Tsirelis and Delopoulos 2011], user tags have been used to automatically generate ground truth for concept-based image retrieval. Each concept is defined as a set of words, and a relevance score is calculated for each image using its tags. Pairs of words are assigned similarity values according to their (co)-occurrences against a corpus. Subsequently, max and average operators are used to compute the final score. Using these score values to automatically annotate images and use them in training SVM classifiers yields better results even when compared to using manual annotation.

Authors of [Ulges et al. 2009] have applied similar strategies in the area of video retrieval. They have used tags assigned to YouTube videos, and have created and automated system that uses query expansion on the tags of the first results page to acquire a sufficient number of videos and train classifiers for concept-based video retrieval.

In this paper we present a novel method for building robust concept detectors from tags, that combines ideas from automatic training set generation and training with fuzzy SVM. We implement both an improved version of [Tsirelis and Delopoulos 2011] to calculate relevance scores, and a method to automatically generate training sets. We subsequently use the Fuzzy SVM to adapt the training procedure and account for the errors propagated from the user tags into the automatic labelling procedure, and build robust and effective concept detection models. We perform extensive experiments to evaluate the effectiveness of the Fuzzy SVM compared to the conventional SVM, and also additional experiments to study the effect of additional noisy training samples, and training set size on the resulting effectiveness.

3. FUZZY SVM TRAINING FROM TAGS

Concept detectors based on Fuzzy SVM are built by solving the following optimisation problem

$$\begin{aligned} &\min_{\mathbf{w},b,\xi_i} \left(\frac{1}{2}\mathbf{w}^T\mathbf{w} + Q^+ \sum_{i=1}^{l} q_i \xi_i + Q^- \sum_{i=l+1}^{N} q_i \xi_i \right) \\ &\text{s.t.} \begin{cases} y_i(\mathbf{w}^T \boldsymbol{\phi}\left(\mathbf{x}_i\right) + b) > 1 - \xi_i \\ \xi_i \geq 0 \end{cases} \end{aligned} \tag{1}$$

where $\mathbf{x}_i$, $i = 1, \ldots, n$ is the feature vector of the i-th training image, y_i is equal to 1 for images 1 to l, indicating that these images depict the concept (i.e., are relevant), and -1 for images $l+1$ to N (indicating that these images do not depict the concept), and q_i is the image training weight. $K(\mathbf{x}_i, \mathbf{x}_j) = \langle \boldsymbol{\phi}\left(\mathbf{x}_i\right), \boldsymbol{\phi}\left(\mathbf{x}_j\right) \rangle$ is a (possibly nonlinear) kernel function corresponding to the inner product in a higher-dimensional Hilbert space [Burges 1998] and Q^+ and Q^- are per class misclassification penalty factors. The slackness variables ξ_i allow the optimisation problem to remain feasible, even when the set of training images cannot be separated (with respect to the concept) with kernel K.

By setting all q_i equal to 1, the conventional SVM optimisation problem is obtained. The addition of q_i does not increase the computational complexity of solving (1), compared to the conventional SVM optimisation problem [Lin and Wang 2002].

The resulting SVM model is used to assign a score

$$f = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}) + b \tag{2}$$

to each image $\mathbf{x}$, thus allowing the ranking of images in an unlabelled collection with respect to the target concept.

In the rest of this Section, we present a method for automatically shaping the training weights q_i based on scores s_i of a ranking of the images derived from tags (which is present in Section some relevance ranking of the 3.2).

Given an image, consider the binary random variable C that indicates the relevance of the image to c (i.e., either $C = c$ or $C = \bar{c}$) and the random variable S that corresponds to its relevance score, as it is computed from tags. The probability that an image with score s is relevant to concept c is therefore $p(C = c|S = s)$, denoted simply as $p(c|s)$ in the following.

For a successful ranking mechanism, $p(c|s)$ is a monotone increasing function of s [Nottelmann and Fuhr 2003]. As a result, high scores yield high probability, implying high confidence in a positive label assignment. Low scores yield low probability, which also implies high confidence, however in negative label assignment. Finally, scores that yield probability close to 0.5 imply minimum confidence of any label assignment. Let

s_{thr} be a score value that satisfies

$$p(c|s_{\text{thr}}) = 0.5 \tag{3}$$

Given the monotonicity of $p(c|s)$ with respect to s it is possible to automatically label the images of the ranked list using

$$y_i = \begin{cases} +1, & \text{if } s_i > s_{\text{thr}} \\ -1, & \text{if } s_i < s_{\text{thr}} \end{cases} \tag{4}$$

and assign training weights that are a linear transformation of the absolute distance of each score s_i from the threshold score s_{thr}

$$q_i = \alpha \, |s_i - s_{\text{thr}}| + \beta \tag{5}$$

Parametres α and β are calculated so that the pairs $(s_i = s_{\text{thr}}, q_i = 0)$ and $(s_i = \max_{s_i} |s_i - s_{\text{thr}}|, q_i = 1)$ satisfy Equation 5. These choices are based on our motivation that (a) images with scores equal to s_{thr} are equally relevant and irrelevant and thus should be discarded completely from the training procedure (through the assignment of zero weight), and (b) the image score with the maximum absolute distance from s_{thr} is the most relevant or irrelevant (according to Equation 4) and should be assigned the maximum training weight.

If the probability $p(c|s)$ is known for every score s, we can also substitute s_i with $p(c|s_i)$ and s_{thr} with 0.5 in Equation (5) and obtain an alternative set of training weights q'_i as

$$q'_i = \alpha' \, |p(c|s_i) - 0.5| + \beta' \tag{6}$$

where parametres α' and β' are calculated so that the pairs $(p_i = 0.5, q'_i = 1)$ and $(p_i = \max_{s_i} |p(c|s_i) - 0.5|, q'_i = 1)$ satisfy Equation 6, based on the same motivation for Equation 5. Equations (5) and (6) correspond to two alternatives for computing the weights of Fuzzy SVM. We onwards refer to the process of creating training weights from relevance scores and using a Fuzzy SVM as s-SVM, and respectively to the process of creating training weights from relevance probabilities and using a Fuzzy SVM as p-SVM.

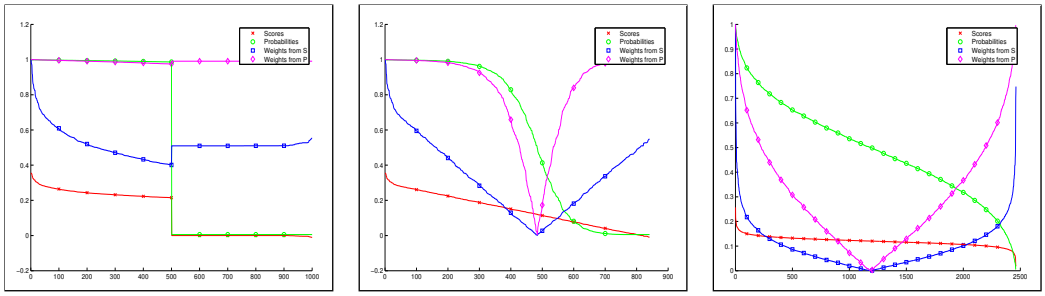
Figure 2 shows the scores of three training sets selected with different selection strategies (details are provided in the experiments of Section 4.3) and their resulting weights q_i (from the scores s_i according to Equation (5)) and q'_i (from the probabilities $p(c|s_i)$, according to Equation (6)).

According to the above, the probabilities $p(c|s_i)$ are required in order to compute the threshold s_{thr} that determines the assignment of training labels, as well as the weights q'_i of Equation (6). Accurate estimation of s_{thr} (through the probabilities estimation) is vital in order to determine proper training weights that enable the Fuzzy SVM classifiers to incorporate each image label fuzziness in the training process. A method for estimating these score probabilities from tags is presented in the following Section 3.1.

3.1. Transforming Ranking Scores to Probabilities

Similarly to the approaches presented in [Manmatha et al. 2001; Arampatzis and van Hameran 2001; Arampatzis and Kamps 2009; Arampatzis and Robertson 2011] we assume that $p(S = s|C = c)$ and $p(S = s|C = \bar{c})$ can be approximated independently using known families of probability density functions (PDFs).

Originally in [Manmatha et al. 2001] the exponential distribution was proposed for $p(s|c)$, and the Gaussian for $p(s|\bar{c})$. However, more recent work of [Markov et al. 2012] concludes that the most general model is to use Gamma distributions for both density



(a) beach (b) elk (c) street

Fig. 2: Ranking scores (derived from the algorithm described in Section 3.2) and estimated probabilities, and shaped training weights from both. In (a) we have selected the 500 images with the highest score and the 500 images with the lowest. In (b) we have chosen approximately 800 images so that their score distribution approaches the uniform, and in (c) $2,500$ images so that their probabilities distribution approaches a Gaussian with $\mu = 0.5$ and $\sigma = 0.2$.

functions. In this work, we experiment with Gaussian, Weibull and Gamma distributions, on each of the two PDFs. We observe (Section 4.3) that the best fit for $p(s|c)$ is the Gaussian whereas the best fit for $p(s|\bar{c})$ is the Gamma.

Estimating the parameters of the relevant and non-relevant distributions requires two image sets, one containing only relevant and one containing only non-relevant images. In order to avoid the need for manual labelling, we propose that all images that contain the exact concept name in their tags are treated as relevant. An equal number of images from the remaining images of the collection is randomly chosen and are treated as non-relevant. These two sets are used to fit the Gaussian and Gamma distributions. Figure 3 shows an example for concept "beach" from the datasets used in the experiments of Section 4. This approach is effective since (i) a small number of samples is required to estimate the one dimensional Gaussian and Gamma distributions, (ii) creating a positive set with "exact matching" of the concept name leads to high accuracy and (iii) unless a concept has a very high prior in the collection the selected non-relevant set will also be accurate. These arguments are also supported by the experimental results presented in Section 4.3.

From the definition of conditional probability and the law of total probability, it can be easily shown that the probability density function of an image being relevant to concept c given its score s_i is given of the form

$$p(c|s_i) = \left(1 + \lambda \frac{p(s_i|\bar{c})}{p(s_i|c)}\right)^{-1} \tag{7}$$

where $\lambda = (1 - p(c))/p(c)$. Since the exact value for λ cannot be accurately obtained, we propose an estimation of the concept prior probability as the number of candidate images that include the concept word in their tags, divided by the total number of candidate images. This approximation of the prior can be used to produce an estimation of the real λ. The selection of λ has a low impact on the threshold s_{thr}. Figure 4 shows an example calculation of the threshold using the real prior probability on a manually annotated dataset, the corresponding value calculated based on the proposed estimation method as well as the threshold for a range of different values of λ. Figure 5 shows the parametrically computed $p(c|s_i)$ for concept four different concepts using the estimated λ. They all approach the sigmoid.

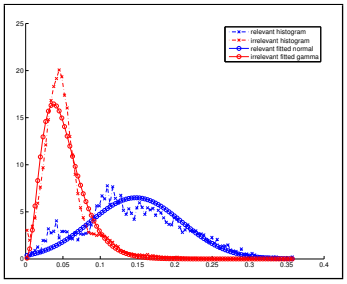


Fig. 3: Histograms of ranking scores for positive (relevant) and negative (irrelevant) images, and fitted Gaussian and Gamma distributions, for concept "beach".

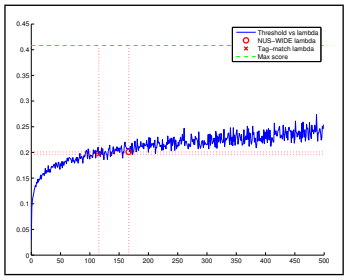


Fig. 4: Variation of s_{thr} versus λ for concept "cars". The pairs $(s_{\text{thr}}, \lambda)$ using the NUS-WIDE annotation prior and the tag-match estimation are also shown.

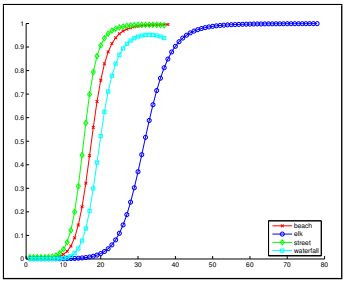


Fig. 5: Probability versus score sigmoid for four concepts, as estimated by our proposed method.

In general, there is no analytic solution to Equations (3) and (7). In practice, we initially estimate the probabilities $p(s|c)$ and $p(s|\bar{c})$ as previously described and then find the two images with the closest probability $p(c|s)$ to 0.5, using Equation (7). We then select as s_{thr} the average of the two scores. In the special case that there exists an image in the list with probability exactly equal to 0.5, s_{thr} is set equal to the score of that image.

Another problem may arise for concepts with extremely low prior probability. In such cases, it is possible that extremely few pictures exist with a ranking score greater than s_{thr}, or even worse, that $p(c|s)$ is never greater than 0.5. If relevant images do exist, then this can occur when the assumption made in [Nottelmann and Fuhr 2003] (that the probability is a monotone preserving transformation of s) does not hold, or when probability estimation is not accurate enough. To deal with such problems, we can manually adjust s_{thr} such that $p(c|s_{\text{thr}}) < 0.5$ in order to select some candidate relevant images.

Even though scores and probabilities can be used to produce training weights independently, it should be noted that the estimation of probabilities is essential, even in the case of training weights derived from scores. Estimating the probabilities allows for calculation of s_{thr} that is required to transform scores to training weights. In a sense, s_{thr} allows a physical interpretation of the raw score values.

3.2. Relevance Assessment from Tags Algorithm

In Sections 3 and 3.1 we have presented a method for producing training weights q_i for the Fuzzy SVM given a relevance ranking score s_i for each image, using Equations (3), (5) and (6) and a set of images with tags. In this Section we present an improved version of the tag-based algorithm proposed in [Tsirelis and Delopoulos 2011] to produce these ranking scores. Probability estimates and s_{thr} are calculated subsequently using (7) and (3), to produce training weights based on the ranking scores or the probability estimates.

Given two words w_1 and w_2, their similarity is approximated with the help of a corpus (a collection of extracts of real world texts). In particular, a commonly used word similarity metric is Pointwise Mutual Information (PMI) [Bouma 2009]. PMI is

defined as

$$PMI = \ln \frac{p(w_1, w_2)}{p(w_1)p(w_2)} = \ln N \frac{N_{1,2}}{N_1 N_2} \tag{8}$$

where $p(w_i)$ is the probability of the word i appearing in a text, and $p(w_1, w_2)$ the probability of the words w_1 and w_2 appearing together in a text. For a corpus with N texts, out of which N_1 contain w_1, N_2 contain w_2 and $N_{1,2}$ contain both, PMI can be calculated as shown in Equation (8).

In this work we also experiment with Mutual Information (MI) defined as

$$MI(w_1, w_2) = p(w_1, w_2) \ln \frac{p(w_1, w_2)}{p(w_1)p(w_2)} = \frac{N_{1,2}}{N} \ln \left(N \frac{N_{1,2}}{N_1 N_2} \right) \tag{9}$$

Furthermore, a variant of the above two metrics is defined based on set difference (i.e. on texts that contain w_1 and do not contain w_2 and vice versa) in order to perform similar computations, the Modified PMI is

$$MPMI(w_1, w_2) = \ln \frac{p(w_1, w_2)}{p(w_1, \bar{w}_2)p(\bar{w}_1, w_2)} = \ln \left(N \frac{N_{1,2}}{M_1 M_2} \right) \tag{10}$$

while the Modified MI is

$$MMI(w_1, w_2) = p(w_1, w_2) \ln \frac{p(w_1, w_2)}{p(w_1, \bar{w}_2)p(\bar{w}_1, w_2)} = \frac{N_{1,2}}{N} \ln \left(N \frac{N_{1,2}}{M_1 M_2} \right) \tag{11}$$

where $p(w_i, \bar{w}_j)$ is the probability that w_i appears in a text and w_j does not, and M_i is the number of texts that contain w_i and do not contain w_j. Finally, we also test the Correlation Coefficient (CR), which is defined as

$$CR(w_1, w_2) = \frac{\mathbf{E}\left\{(I_1 - \bar{I}_1)(I_2 - \bar{I}_2)\right\}}{\sigma_{I_1}\sigma_{I_2}} \tag{12}$$

where I_i is an indicator binary random variable that is 1 when word w_i is in a text and 0 otherwise, $\bar{I}_i$ is the mean value of random variable I_i and σ_{I_i} is the standard deviation of I_i. CR achieves the best results in the experiments of Section 4.2.

Regardless of the metric used to assess similarity between two words, the relevance of an image to a concept can be calculated by extending this similarity to sets of words, namely the set $\mathbf{T}$ of image tags and a set of words $\mathbf{W}$ that are associated with the concept.

To populate $\mathbf{W}$, we use a set $\mathbf{T}_c$ of words that includes the concept name (usually a single word) as well as a set of extra a priori selected words that are closely related to the concept, and are part of the concept's definition, as per [Tsirelis and Delopoulos 2011]. We use multiple words for the concept definition in order to disambiguate between different possible semantic interpretations of the concept word, as in [Diou et al. 2010a]. The words in $\mathbf{W}$ are then retrieved by querying WordNet with each of the words in $\mathbf{T}_c$ and collecting all the resulting synsets. An example of this process is demonstrated in Table I. It is important to note that in all experiments of Section 4, the words in $\mathbf{T}_c$ were selected at the concept definition stage, and the same set is used for both Fuzzy SVMs and SVM, allowing the valid comparison between the two methods.

Similarity between a word w and a set of words $\mathbf{T}$ is defined as the maximum similarity of w with each of the words in $\mathbf{T}$

$$R(w, \mathbf{T}) = \max_{t \in \mathbf{T}} r(w, t) \tag{13}$$

Table I: Example of a concept, the provided concept terms, and the produced set of words for the concept

Concept word	coral
Concept terms $\mathbf{T}_c$ (part of concept definition)	coral, sea, seabed marine, sealife, reef
Concept words $\mathbf{W}$	coral, sea, ocean, marin, leatherneck, nautic, maritim, reef

where r can be any of MI, MMI, PMI, MPMI and CR, or any other metric that estimates word-pair similarity.

Tsirelis et al. defined in [Tsirelis and Delopoulos 2011] a symmetric similarity function between sets of words as

$$\mathrm{sim}_0(\mathbf{T}, \mathbf{W}) = \frac{1}{|\mathbf{W}|} \sum_{w \in \mathbf{W}} R(w, \mathbf{T}) + \frac{1}{|\mathbf{T}|} \sum_{t \in \mathbf{T}} R(t, \mathbf{W}) \tag{14}$$

This metric takes into account both the similarity of each word w of the concept with a set of image tags $\mathbf{T}$, and conversely, the similarity of each tag t of the image with the set of concept words $\mathbf{W}$.

In this current work, we extend the definition of Equation (14) so that words of $\mathbf{W}$ are weighted according to their relevance to the concept. To achieve this, the concept terms $\mathbf{T}_c$ are used to factor the two sums of the similarity. The main reason for this choice is that not all words of $\mathbf{W}$ are equally related to the concept c. More specifically, we update the similarity of each word w of the concept with the image tags to be

$$R'_1(w, \mathbf{T}) = R(w, \mathbf{T}) \cdot R(w, \mathbf{T}_c) \tag{15}$$

while the similarity of a tag t to the concept words $\mathbf{W}$ becomes

$$R'_2(t, \mathbf{W}) = R(t, \mathbf{W}) \cdot R(w^*(t), \mathbf{T}_c) \tag{16}$$

where $w^*(t)$ is the word of $\mathbf{W}$ most similar to t, i.e.

$$w^*(t) = \arg \max_{w \in \mathbf{W}} r(w, t)$$

The similarity between sets of words is now defined as

$$\mathrm{sim}(\mathbf{T}, \mathbf{W}) = \frac{1}{|\mathbf{W}| + |\mathbf{T}|} \left(\sum_{w \in \mathbf{W}} R'_1(w, \mathbf{T}) + \sum_{t \in \mathbf{T}} R'_2(t, \mathbf{W}) \right) \tag{17}$$

Equation (17) is used to produce ranking scores for every image-concept pair, as required for training set selection and Fuzzy SVM training discussed in the beginning of this Section.

3.3. Concept Detector Training Pipeline

Figure 6 summarises the proposed concept detector training pipeline, putting together all the components discussed in the previous (i.e. computation of relevance scores, estimation of probabilities, training set selection and weight shaping).

The only input of the pipeline is the concept-terms set $\mathbf{T}_c$ (that defines the concept c). The set of words $\mathbf{W}$ is then constructed as described in Section 3.2 and a large list of images is collected from the community repository, along with the associated image tags. In case the resulting image set is not sufficiently large, one can use the following procedure to further expand the set of candidate training images: Each word of $\mathbf{W}$ is submitted as a query to WordNet to retrieve synsets resulting from all possible

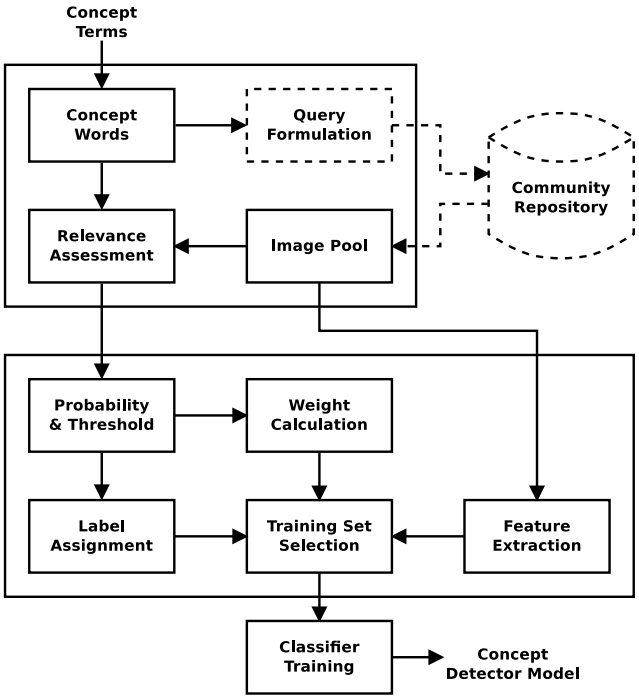


Fig. 6. Concept detector training pipeline: The process starts by providing the concept terms. An optional step (denoted with dashed lines) can use community repositories to retrieve user tagged images. Images are then processed textually (label and weight assignment) and visual (feature extraction), and a training set is selected. A concept detector model is then built using the Fuzzy SVM classifier.

relations such as synonyms, antonyms, meronyms, generalisations, etc. All words of all returned synsets are used to query the community repository, one at a time.

The training set formulation is bootstrapped by using the similarity between tags of each image and words of the concept, as presented in Section 3.2 and Equation (17). As a result, each image is assigned a score, and the probability estimation approach of Section 3.1 is applied afterwards, resulting in a probability estimate for each image. The threshold s_{thr} can also be computed according to Equation (3). At this point, training weights can be shaped using either the ranking scores or the probabilities, and images can be automatically labelled by comparison of each image score to s_{thr}. Furthermore, instead of using the entire image list as a training set, a selection strategy can be applied. Out of the many possible options, we choose to select the N top-score and the N bottom-score images as in the original paper of [Tsirelis and Delopoulos 2011] (we also experiment with additional training set selection methods to further evaluate the effectiveness of the constructed classifiers, in Section 4.5). Note that selecting according to scores or according to probabilities results in the same set of images. For both s-SVM and p-SVM the values of the training weights depend on the selected threshold s_{thr}.

Having constructed the concept's training set, visual features of the selected images are computed and are supplied to the Fuzzy SVM training procedure to produce the output concept detector model.

4. EXPERIMENTS

4.1. Experimental Setup

We apply our framework on two different datasets, the NUS WIDE [Chua et al. 2009] image collection and the extended MIR FLICKR [Huiskes and Lew 2008; Huiskes et al. 2010] dataset. Both provide manual groundtruth annotations for various concepts per

Table II: Mean Average Precision of tag-based relevance assessment algorithm for five word-pair similarity metrics over the 81 NUS-WIDE concepts for the entire image collection

Metric	MI	MMI	PMI	MPMI	CR
Mean MAP	0.166	0.129	0.0418	0.150	**0.2438**

image. We use these annotations in order to create comparable results of the effectiveness of our method.

NUS WIDE contains about $27,0000$ images collected from Flickr, that have been split into a development set and a test set. For each image, six visual features are provided, along with the images' URLs, user tags, and manual labelling for 81 concepts. We use this development set as the collected image list (or image pool) described in Section 3.3 (i.e. the set of images that have been gathered after an initial set of queries to Flickr). We create training sets by selecting images from this development set using various selection methods (like N top and N bottom). Concept detector effectiveness is evaluated at the NUS-WIDE test set. As a baseline comparison, the average prior probability over the 81 concepts in the test set is 0.0232. Prior probabilities for each individual concept can be found in [Chua et al. 2009].

MIR FLICKR contains 1 million images also collected from Flickr. For each image, the dataset provides tags, image URLSs, EXIF data and two visual features. Out of the 1 million images, $25,000$ have been manually labelled against 14 concepts. We therefore selected to use these $25,000$ images as test set, and the remaining $975,000$ images as pool for training set selection (development set). As a baseline comparison, the average prior probability is 0.0714 on the $25,000$ annotated images of MIR FLICKR.

4.2. Relevance Assessment from Tags

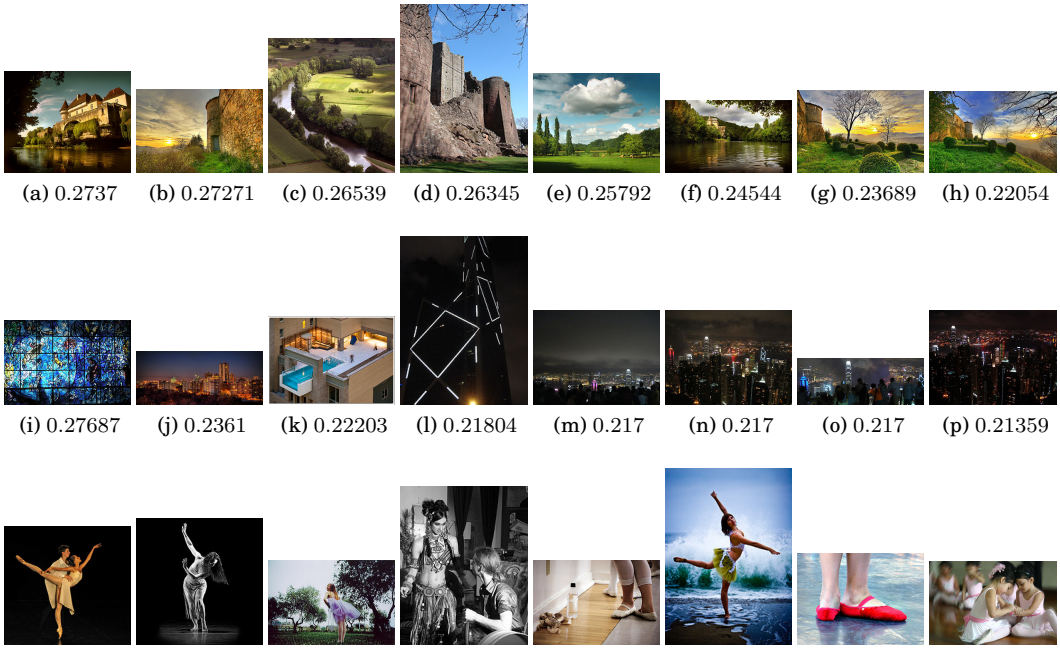
In this experiment we evaluate the effectiveness of the ranking algorithm for relevance assessment from tags, using the five word-pair similarity functions described in Section 3.2. We process each image by stemming its tags using the Porter Stemmer [Porter 1980] and removing stop words and duplicate tags. Relevance scores are then calculated for each image-concept pair. For each concept, images are sorted descendingly according to score. Average Precision (AP) results at the entire NUS-WIDE collection for the five versions of the proposed algorithm are shown in Table II. These results evaluate the tag-based ranking of the NUS-WIDE development set, as it results from the algorithm presented in Section 3.2. The manual annotations provided by the NUS-WIDE dataset are used as groundtruth and the CR achieves the highest Mean Average Precision (MAP) over the 81 concepts.

Figure 7 shows the top eight images and their scores for three concepts ("castle", "cityscape" and "dancing"). These images are taken from the ranking produced by the algorithm using the CR for word-pair similarity.

4.3. Experiments on Probability Estimation

In order to evaluate different options for the parametric probability density functions for $p(s|c)$ and $p(s|\bar{c})$ in Equation (7), we use the score histograms on the set of manually annotated relevant and non-relevant images respectively as baseline score PDFs.

The method presented in 3.1 is used to obtain data for estimating different types of parametric distributions. Gaussian, Weibull and Gamma distributions are used for both relevant and non-relevant score distributions. Jensen-Shannon (JS) divergence is used to select the most effective distribution compared to the baseline distributions derived from the histograms. Table III illustrates the mean JS divergence for each distribution type. A voting evaluation is also presented, where each concept votes the



(a) 0.2737 (b) 0.27271 (c) 0.26539 (d) 0.26345 (e) 0.25792 (f) 0.24544 (g) 0.23689 (h) 0.22054

(i) 0.27687 (j) 0.2361 (k) 0.22203 (l) 0.21804 (m) 0.217 (n) 0.217 (o) 0.217 (p) 0.21359

(q) 0.3983 (r) 0.35368 (s) 0.35128 (t) 0.35018 (u) 0.34775 (v) 0.3453 (w) 0.34511 (x) 0.33624

Fig. 7: Top eight ranked images for the concepts “castle”, “cityscape” and “dancing” (per row) using the relevance assessment from tags algorithm with the Correlation Coefficient metric for word pair similarity. Missing images were omitted. Images courtesy of Flickr.com.

Table III: Jensen-Shannon divergence for three distributions compared to a baseline histogram distribution, and number of concepts achieving minimum divergence, both for relevant and non-relevant images

	Jensen-Shannon div.		**No. of concepts**	
	relevant	**irrelevant**	**relevant**	**irrelevant**
Gaussian	**0.7053**	0.3845	**39**	2
Weibull	0.7147	0.2159	28	24
Gamma	0.8143	**0.1848**	14	**55**

distribution type that achieves minimum JS divergence (one vote for the relevant images distribution and one for the non-relevant). Based on these results, we select the Gaussian-Gamma distributions model for relevant and non-relevant images respectively.

The effect of λ on the computation of s_{thr} in Equations (7) and (3) is measured by comparing this threshold for different values of λ. More specifically, we use the thresholds resulting from λ_a, obtained using the prior probability from the NUS-WIDE manual annotation, and λ_b, obtained using the prior probability estimated through tags (Section 3.1). The mean value of the difference of the two thresholds ($s_{\text{thr}}(\lambda_b) - s_{\text{thr}}(\lambda_a)$) over the 81 concepts is found to be -0.0076 and its standard deviation 0.0395. Furthermore, we calculate s_{thr} for a range of λ_i for each concept, and subtract $s_{\text{thr}}(\lambda_a)$. We then obtain the mean threshold difference for each concept and average across the 81 concepts. The result is 0.0151 with standard deviation 0.0235. These results indicate that

the threshold is only slightly affected by the choice of λ, given that the mean range of scores on the NUS-WIDE development set is 0.3125 for the 81 concepts.

4.4. Training with s-SVM and p-SVM

In these experiments we evaluate the performance of the proposed s-SVM and p-SVM training procedure, compared to training without weights using the conventional SVM. We apply our entire framework separately on the NUS WIDE and MIR FLICKR datasets. The training set for each concept is selected by sampling from each development set. We calculate relevance scores, probabilities and s_{thr} for each concept, and select the N top and N bottom images according to their score. In all stages of this pipeline we use only the images and image tags available in the development sets of each dataset.

In the experiments with NUS WIDE we use the following five feature vectors, provided by the dataset: 64-D colour histogram (CH), 144-D colour correlogram (CORR), 75-D edge direction histogram (EDH), 128-D wavelet texture (WT), and 255-D block-wise colour moments (CM55). A total of five classifiers are trained, one on each feature, and final ranking is performed by average fusion. This late fusion scheme achieved the highest effectiveness in the paper that introduced the NUS WIDE dataset [Chua et al. 2009]. We experiment both with the linear kernel for all features, as well as with a combination of non-linear kernels (RBF kernel on WT and CM55, and the χ^2 kernel on CH, CORR and EDH). In the experiments with the MIR FLICKR dataset we use the edge histogram feature that is provided, using both the linear and the RBF kernel.

Three experiments are performed for each concept. In the first, a conventional SVM is trained on the $2N$ automatically labelled images, and is used as a baseline. In the second, s-SVM is trained on the same images with the same labels, incorporating training weights of Equation (5). Finally, in the third test, p-SVM is used on the same images and same labels, with the estimated probabilities as training weights (Equation (6)).

We select N to be $2,000$ for the $161,789$ images of NUS WIDE. For MIR FLICKR, the development set contains $975,000$ and we therefore experiment with N equal to $2,000$, $4,000$ and $6,000$, to study the effect of training set size to concept detector effectiveness. We also set the misclassification penalty factors $Q^+ = 5$ and $Q^- = 1$ for all concepts and training set sizes in order to boost the effect of the positive class.

The resulting models are used to rank the two test sets. Results in terms of MAP over the 81 concepts for the NUS WIDE are presented in Table IV. Similarly, results for the MIR FLICKR over the 14 concepts are presented in Table V for training set sizes similar to NUS WIDE ($N = 2,000$). Tables VI and VII show results for the MIR FLICKR dataset using larger training sets ($N = 4,000$ and $N = 6,000$ respectively).

Column "C" presents the mean percentage improvement of each concept's AP when using the s-SVM or the p-SVM over the conventional SVM, over the 81 concepts, and column "D" the probability of a concept AP improving when using the s-SVM or the p-SVM instead of the SVM. This probability is calculated as the number of concepts with positive improvement divided by the number of concepts. On average, concept AP is improved with the s-SVM and p-SVM, as indicated by columns "D" being over 50%. The linear kernel benefits more than the non-linear kernels, as indicated by the increased mean percentage improvement (column "C") in row "A" compared to row "B". In particular, using the s-SVM on the MIR FLICKR with a linear kernel and the top-bottom training set selection strategy yields a mean 46% improvement, with 93% probability of improvement (meaning that AP improved for 13 out of the 14 concepts).

Figure 9 shows number of concepts that correspond to AP improvement bins, arranged on the horizontal axis, for both s-SVM and p-SVM and both linear and non-linear kernels. The majority of concept classifiers perform better when trained with

Table IV: SVM, s-SVM and p-SVM performance for (A) the linear and (B) the non-linear kernels, in terms of MAP, (C) mean percentage improvement of AP, and (D) percentage of concepts that improve with s-SVM or p-SVM (NUS WIDE).

Selecting from the top and bottom							
	SVM	s-**SVM**			p-**SVM**		
	MAP	MAP	C	D	MAP	C	D
A	0.0760	0.0802	5.5862%	60.4938%	0.0772	2.6102%	62.9630%
B	0.0829	0.0789	−5.8440%	33.3333%	0.0812	−2.7296%	56.7901%
Uniform distribution on relevance scores							
	SVM	s-**SVM**			p-**SVM**		
	MAP	MAP	C	D	MAP	C	D
A	0.0605	0.0840	56.2212%	86.4198%	0.0771	32.5218%	75.3086%
B	0.0712	0.0839	30.5509%	77.7778%	0.0806	19.5237%	74.0741%
Gaussian distribution $\mathcal{N}(0.5, 0.2)$ on probabilities							
	SVM	s-**SVM**			p-**SVM**		
	MAP	MAP	C	D	MAP	C	D
A	0.0290	0.0320	32.5621%	74.0741%	0.0313	23.8333%	70.3704%
B	0.0382	0.0405	32.2053%	69.1358%	0.0412	26.5280%	66.6667%

Table V: SVM, s-SVM and p-SVM performance for (A) the linear and (B) the non-linear kernels, in terms of MAP, (C) mean percentage improvement of AP, and (D) percentage of concepts that improve with s-SVM or p-SVM (MIR FLICKR, $N = 2,000$).

Selecting from the top and bottom							
	SVM	s-**SVM**			p-**SVM**		
	MAP	MAP	C	D	MAP	C	D
A	0.1109	0.1313	46.0246%	92.8571%	0.1229	21.3080%	85.7143%
B	0.1254	0.1362	14.0110%	71.4286%	0.1324	9.2340%	64.2857%
Uniform distribution on relevance scores							
	SVM	s-**SVM**			p-**SVM**		
	MAP	MAP	C	D	MAP	C	D
A	0.1070	0.1191	38.0572%	78.5714%	0.1157	25.5499%	71.4286%
B	0.1135	0.1273	28.7751%	85.7143%	0.1248	20.5357%	71.4286%
Gaussian distribution $\mathcal{N}(0.5, 0.2)$ on probabilities							
	SVM	s-**SVM**			p-**SVM**		
	MAP	MAP	C	D	MAP	C	D
A	0.0713	0.0778	11.5050%	78.5714%	0.0766	5.3514%	71.4286%
B	0.0783	0.0702	−8.3344%	28.5714%	0.0754	1.5586%	42.8571%

Table VI: SVM, s-SVM and p-SVM performance for (A) the linear and (B) the non-linear kernels, in terms of MAP, (C) mean percentage improvement of AP, and (D) percentage of concepts that improve with s-SVM or p-SVM (MIR FLICKR, $N = 4,000$).

Selecting from the top and bottom							
	SVM	s-**SVM**			p-**SVM**		
	MAP	MAP	C	D	MAP	C	D
A	0.1090	0.1275	73.4119%	71.4286%	0.1241	50.3541%	71.4286%
B	0.1256	0.1381	24.6561%	85.7143%	0.1406	30.2030%	78.5714%
Uniform distribution on relevance scores							
	SVM	s-**SVM**			p-**SVM**		
	MAP	MAP	C	D	MAP	C	D
A	0.1123	0.1255	50.2851%	71.4286%	0.1193	39.9371%	64.2857%
B	0.1202	0.1298	20.6564%	78.5714%	0.1259	15.7055%	71.4286%
Gaussian distribution $\mathcal{N}(0.5, 0.2)$ on probabilities							
	SVM	s-**SVM**			p-**SVM**		
	MAP	MAP	C	D	MAP	C	D
A	0.0739	0.0765	5.5018%	50.0000%	0.0769	19.3738%	71.4286%
B	0.0728	0.0759	−0.7722%	35.7143%	0.0727%	3.4715%	64.2857%

Table VII: SVM, s-SVM and p-SVM performance for (A) the linear and (B) the non-linear kernels, in terms of MAP, (C) mean percentage improvement of AP, and (D) percentage of concepts that improve with s-SVM or p-SVM (MIR FLICKR, $N = 6,000$).

Selecting from the top and bottom							
	SVM	**s-SVM**			**p-SVM**		
	MAP	MAP	C	D	MAP	C	D
A	0.1071	0.1308	95.2771%	78.5714%	0.1210	63.4586%	64.2857%
B	0.1211	0.1371	36.0014%	85.7143%	0.1384	36.0795%	78.5714%
Uniform distribution on relevance scores							
	SVM	**s-SVM**			**p-SVM**		
	MAP	MAP	C	D	MAP	C	D
A	0.1126	0.1292	41.9082%	78.5714%	0.1239	33.2484%	78.5714%
B	0.1219	0.1311	15.9130%	85.7143%	0.1324	16.5582%	78.5714%
Gaussian distribution $\mathcal{N}(0.5, 0.2)$ on probabilities							
	SVM	**s-SVM**			**p-SVM**		
	MAP	MAP	C	D	MAP	C	D
A	0.0743	0.0716	7.1019%	42.8571%	0.0771	16.7360%	64.2857%
B	0.0775	0.0799	8.5190%	57.1429%	0.0779	10.7727%	57.1429%

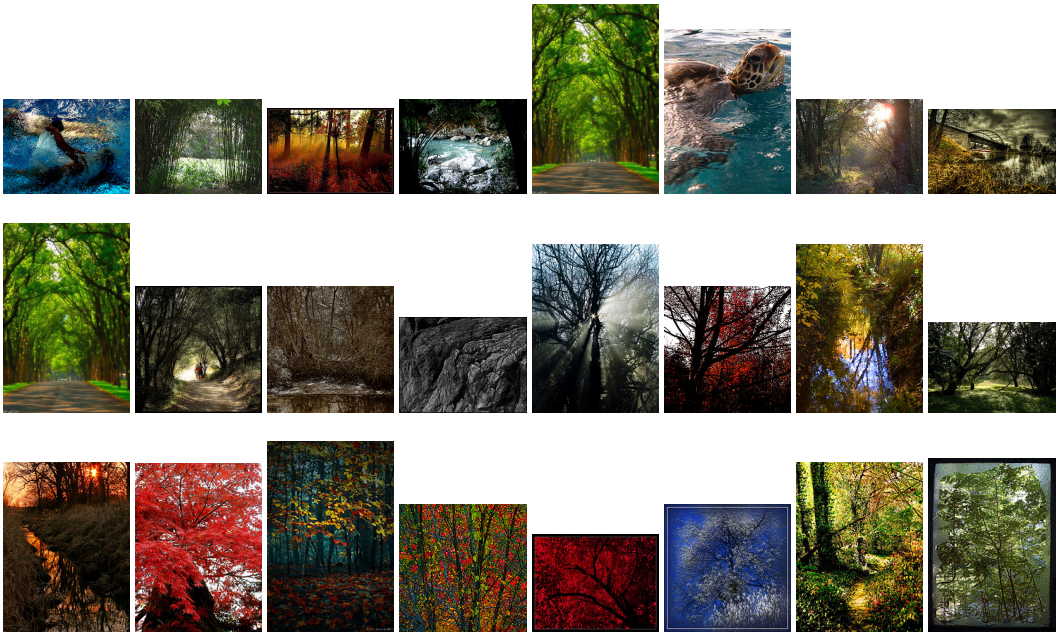


Fig. 8: Top retrieved images for the concept “tree” on a moderately noisy training set (uniform distribution of relevance scores). Rows correspond to trained SVM, s-SVM (using weights from relevance) and p-SVM (using weights from probability). Missing images were omitted. Note that the completely non-relevant images (a) and (f) have been been replaced, and more concept-representative images like (e) have been placed in higher ranking positions. Images courtesy of Flickr.com.

s-SVM or p-SVM, and some concept AP show improvement of as high as 900% on the NUS WIDE and 280% for the MIR FLICKR.

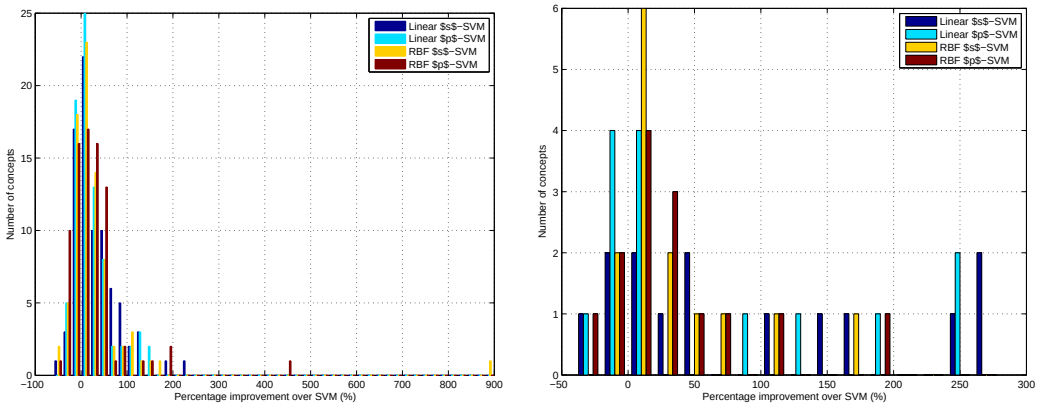


Fig. 9: Number of concepts versus percentage improvement of s-SVM and p-SVM classifiers versus the conventional SVM. Left figure shows results on the NUS WIDE, using the third training set selection choice (Gaussian distribution of probabilities). Right figure shows results on the MIR FLICKR, using the first training set selection choice (N top and N bottom images, $N = 6,000$). Only a small number of concepts show inferior effectiveness, while some concept achieve percentage improvement of 280%, 450% and almost 900%.

4.5. Training on Noisier Setups

In order to further demonstrate the improved effectiveness of s-SVM and p-SVM over the conventional SVM, we perform all of the experiments described in Section 4.4 two more times, changing the method of selecting the training set images. The motivation is to create training sets with different levels of noise (additional errors in automatic label assignment).

In the first experiment, images are selected so that the distribution of relevance scores in the produced training set approximates the uniform. This is achieved by splitting the range of scores into short bins of equal length and randomly sampling an equal amount of images from each. As a result, the produced training set contains images with varying confidence for the automatic label assignment, creating moderately noisy training sets. Approximately $2N$ images (due to the fact that some bins are sparsely populated) are selected for each concept, to produce comparable results with the training sets of Section 4.4.

In the second experiment, images are selected so that the distribution of their probabilities approximates a Gaussian with mean $\mu = 0.5$ and standard deviation $\sigma = 0.2$. This is performed by also splitting the score range into equal bins, and random sampling a different number of images from each bin. This selection scheme results in training sets that include more images with lower confidence for the automatic label assignment, creating highly noisy training sets. Each training set contains approximately $2N$ images (usually slightly less than $2N$, again due to the fact that some bins are sparsely populated).

Results for these two sets of experiments are also available in Tables IV and V, in the middle and bottom part accordingly. For the moderately noisy selection scheme, similar MAPs are observed for all classifiers, with s-SVM and p-SVM demonstrating increased effectiveness. Even though training sets are noisier compared to the ones of the previous Section, the conventional SVM sometimes achieves marginally higher effectiveness across the two experiments. This is probably a result of the increased

diversity of images introduced in the training sets by the uniform sampling method used to create them. Finally, in the highly noisy training sets (lower part of Table IV), conventional SVM effectiveness has significantly dropped, because of the introduced mislabelled images of the training sets. However, both s-SVM and p-SVM clearly outperform the conventional SVM, and s-SVM achieve significantly higher effectiveness, as indicated by the increased number of concepts that improved (column "D") and the average percentage improvement of AP of each concept (column "C").

4.6. Increasing the training set size

Tables VI and VII show results for the same experiments shown in Table V using larger values for the training set size. In general, classifier effectiveness does not change significantly the conventional SVM while it has an overall increasing trend for s-SVM and p-SVM. For example, using a linear kernel and selecting the top N and bottom N images for the training set (the best performing strategy as indicated both from these results and the work of [Tsirelis and Delopoulos 2011]), mean percentage improvement of s-SVM over the conventional is 46% for $N = 2,000$, 73% for $N = 4,000$ and 95% for $N = 6,000$. This indicates that enriching the training set with additional noisy examples leads to improvement for the proposed s-SVM and p-SVM approaches, even if they do not lead to improvement for the conventional SVM. The reason for this is that the weighting mechanism assigns higher importance weights to samples that are likely to be correctly labelled, thus leading to better concept detector models. In this sense, these results are particularly encouraging, since they demonstrate that the additional gain of extra training samples can be greatly amplified (almost doubled) by proper incorporation of their reliability through the use of training weights of s-SVM and p-SVM.

4.7. Comparison with cross-domain concept detectors

One alternative to using tags (and automatically constructing concept detectors) is the direct application of pre-trained concept models. More specifically, one can train a set of concept models in an existing, annotated training set, and then apply these models to the target dataset. We argue [Diou et al. 2010b] that such cross-domain concept detection approaches cannot be directly applied effectively and that the proposed approach, that exploits tags, can lead to significantly better results.

To support our argument we performed an additional experiment that compares the effectiveness of the proposed automatic concept detector training method, with the direct application of a set of VIREO-374 [Jiang et al. 2007; Jiang et al. 2010] concept detectors. VIREO-374 provides several pre-trained SVM concept detector models, that have been trained on the TRECVID2005 development set using the RBF kernel and a "bag-of-visual-words" feature. Feature extraction software is also provided and was used for this experiment. The overlap of VIREO-374 and MIR FLICKR is 11 concepts.

The VIREO-374 concept detectors were directly applied to rank the test set of MIR FLICKR for these concepts, achieving 0.052 MAP. On the other hand, we applied our method by training RBF models for those 11 concepts using the same feature vector. We used all three presented strategies for training set selection: top and bottom (Section 4.4), uniformly distrubuted scores (Section 4.5), and Gaussianly distributed probabilities (Section 4.5). Results are presented in Table VIII for medium training set sizes ($N = 4,000$). Our method achieves almost four times the MAP of the VIREO-374 detectors, at the first two experiments. This is particularly motivating, especially when taking into account that, in the second experiment, a medium level of noise has been introduced in the training set. Finally, at the third experiment, where the training set suffers from heavily noised annotations, our method still achieves double MAP compared to VIREO-374 concept detectors.

Table VIII: Comparison with cross domain concept detectors using the same feature vector. MAP over the 11 common concepts of MIR FLICKR and VIREO 374. For our method, we have used the medium training set sizes ($N = 4,000$, similar results are achieved for $N = 2,000$ and $N = 6,000$) and performed all three training set selection methods. For the VIREO-374, we have used the provided trained models directly on the test set.

	SVM	**s-SVM**	**p-SVM**	**VIREO-374**
Top & bottom	0.2013	0.2013	**0.2041**	
Uniform dist.	0.1902	0.1950	**0.1972**	0.0520
Gaussian dist.	0.1080	**0.1282**	0.1252	

5. CONCLUSIONS

We have presented a novel approach for concept detector training using confidence values automatically derived from tags as weights in the training procedure of Fuzzy SVM. First, a highly effective method for ranking candidate training images was outlined, that uses existing image tags, a reference corpus and wordnet to assign scores with respect to a concept. Then, two alternatives for computing the training weights were introduced and evaluated. The first (s-SVM) uses the relevance assessment scores, while the second (p-SVM) is based on relevance probability estimates.

Component-level evaluation experiments indicate the effectiveness of all elements of the proposed architecture. With respect to the collection of training images, our relevance assessment algorithm achieves a MAP of 24.38%, compared to the mean prior probability of 2.32% for the NUS-WIDE dataset. For probability mapping, experiments have evaluated different distributions for estimating probabilities from the relevance scores of the automatically annotated positive and negative samples. These distributions require a minimal set of estimated parameters, thus increasing the estimation accuracy when the number of available samples is small.

The combination of relevance scores and probability is used to produce training weights for the Fuzzy SVMs. The proposed strategy for automatic training sample selection, labelling, weighting and Fuzzy SVM training, yields concept detector models with increased effectiveness. In fact, our method, based on s-SVM and p-SVM, achieves higher average precision values for the NUS-WIDE dataset, compared to the conventional SVM. Additional experiments further demonstrate the training error resilience of the proposed concept detection mechanism. In these experiments, we experimented with training sets of lower quality, by introducing additional mislabelled images. In all cases, the effectiveness of s-SVM and p-SVM remained higher than that of the conventional SVM (up to 64.12% improvement of concept AP for the s-SVM and 29.85% for the p-SVM, with as many as 89% and 79% of the concepts improving respectively).

Experiments on an additional dataset (MIR FLICKR) indicate that our framework is general and is not dataset-dependent. We applied the proposed framework directly, and observed that the s-SVM and p-SVM concept detectors clearly outperform the conventional SVM, achieving 95% and 63% improvement of MAP for s-SVM and p-SVM, with as many as 93% and 86% of the concepts improving respectively.

Future work includes the extension of the presented ideas to improving concept-based retrieval using other sources of evidence for training sample relevance, such as clickthrough data and user-provided feedback.

REFERENCES

Avi Arampatzis and Jaap Kamps. 2009. A signal-to-noise approach to score normalization. In *Proceedings of the 18th ACM conference on Information and knowledge management*. ACM, 797–806.

Avi Arampatzis and Stephen Robertson. 2011. Modeling score distributions in information retrieval. *Information Retrieval* 14, 1 (2011), 26–46.

Avi Arampatzis and André van Hameran. 2001. The score-distributional threshold optimization for adaptive binary classification tasks. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 285–293.

Gerlof Bouma. 2009. Normalized (pointwise) mutual information in collocation extraction. In *Proceedings of the Biennial GSCL Conference*. 31–40.

Francesca Bovolo, Lorenzo Bruzzone, and Lorenzo Carlin. 2010. A novel technique for subpixel image classification based on support vector machine. *Image Processing, IEEE Transactions on* 19, 11 (2010), 2983–2999.

Christopher JC Burges. 1998. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery* 2, 2 (1998), 121–167.

Tat-Seng Chua, Jinhui Tang, Richang Hong, Haojie Li, Zhiping Luo, and Yantao Zheng. 2009. NUS-WIDE: a real-world web image database from National University of Singapore. In *Proceedings of the ACM International Conference on Image and Video Retrieval*. ACM, 48.

Ritendra Datta, Jia Li, and James Z Wang. 2005. Content-based image retrieval: approaches and trends of the new age. In *Proceedings of the 7th ACM SIGMM international workshop on Multimedia information retrieval*. ACM, 253–262.

Christos Diou, George Stephanopoulos, Panagiotis Panagiotopoulos, Christos Papachristou, Nikos Dimitriou, and Anastasios Delopoulos. 2010a. Large-Scale Concept Detection in Multimedia Data Using Small Training Sets and Cross-Domain Concept Fusion. *IEEE Transactions on Circuits and Systems for Video Technology* 20, 12 (2010), 1808–1821.

Christos Diou, George Stephanopoulos, Panagiotis Panagiotopoulos, Christos Papachristou, Nikos Dimitriou, and Anastasios Delopoulos. 2010b. Large-scale concept detection in multimedia data using small training sets and cross-domain concept fusion. *Circuits and Systems for Video Technology, IEEE Transactions on* 20, 12 (2010), 1808–1821.

Ralph Ewerth, Khalid Ballafkir, M Muhling, Dominik Seiler, and Bernd Freisleben. 2012. Long-Term Incremental Web-Supervised Learning of Visual Concepts via Random Savannas. *Multimedia, IEEE Transactions on* 14, 4 (2012), 1008–1020.

Amirhossein Habibian and Cees GM Snoek. 2014. Recommendations for recognizing video events by concept vocabularies. *Computer Vision and Image Understanding* 124 (2014), 110–122.

Mark J Huiskes and Michael S Lew. 2008. The MIR flickr retrieval evaluation. In *Proceedings of the 1st ACM international conference on Multimedia information retrieval*. ACM, 39–43.

Mark J Huiskes, Bart Thomee, and Michael S Lew. 2010. New trends and ideas in visual concept detection: the MIR flickr retrieval evaluation initiative. In *Proceedings of the international conference on Multimedia information retrieval*. ACM, 527–536.

Yu-Gang Jiang, Chong-Wah Ngo, and Jun Yang. 2007. Towards optimal bag-of-features for object categorization and semantic video retrieval. In *Proceedings of the 6th ACM international conference on Image and video retrieval*. ACM, 494–501.

Yu-Gang Jiang, Jun Yang, Chong-Wah Ngo, and Alexander G Hauptmann. 2010. Representations of keypoint-based semantic concept detection: A comprehensive study. *Multimedia, IEEE Transactions on* 12, 1 (2010), 42–53.

Xue-Ming Leng and Yi-Ding Wang. 2008. Gender classification based on fuzzy SVM. In *Machine Learning and Cybernetics, 2008 International Conference on*, Vol. 3. IEEE, 1260–1264.

Han-Xiong Li, Jing-Lin Yang, Geng Zhang, and Bi Fan. 2012. Probabilistic support vector machines for classification of noise affected data. *Information Sciences* (2012).

Chun-Fu Lin and Sheng-De Wang. 2002. Fuzzy support vector machines. *Neural Networks, IEEE Transactions on* 13, 2 (2002), 464–471.

Chun-fu Lin and Sheng-de Wang. 2004. Training algorithms for fuzzy support vector machines with noisy data. *Pattern Recognition Letters* 25, 14 (2004), 1647–1656.

Yi Liu and Yuan F Zheng. 2007. Soft SVM and its application in video-object extraction. *Signal Processing, IEEE Transactions on* 55, 7 (2007), 3272–3282.

Michael I Mandel, Razvan Pascanu, Douglas Eck, Yoshua Bengio, Luca M Aiello, Rossano Schifanella, and Filippo Menczer. 2011. Contextual tag inference. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)* 7, 1 (2011), 32.

R Manmatha, T Rath, and Fangfang Feng. 2001. Modeling score distributions for combining the outputs of search engines. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 267–275.

Ilya Markov, Avi Arampatzis, and Fabio Crestani. 2012. Unsupervised linear score normalization revisited. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*. ACM, 1161–1162.

Rui Min and HD Cheng. 2009. Effective image retrieval using dominant color descriptor and fuzzy support vector machine. *Pattern Recognition* 42, 1 (2009), 147–157.

Henrik Nottelmann and Norbert Fuhr. 2003. From uncertain inference to probability of relevance for advanced IR applications. In *Advances in Information Retrieval*. Springer, 235–250.

Stefanie Nowak and Stefan Rüger. 2010. How reliable are annotations via crowdsourcing: a study about inter-annotator agreement for multi-label image annotation. In *Proceedings of the international conference on Multimedia information retrieval*. ACM, 557–566.

Martin F Porter. 1980. An algorithm for suffix stripping. *Program: electronic library and information systems* 14, 3 (1980), 130–137.

Gholamreza Rafiee, Satnam Singh Dlay, and Wai Lok Woo. 2010. A review of content-based image retrieval. In *Communication Systems Networks and Digital Signal Processing (CSNDSP), 2010 7th International Symposium on*. IEEE, 775–779.

Yong Rao, Padma Mundur, and Yelena Yesha. 2006. Fuzzy SVM ensembles for relevance feedback in image retrieval. In *Image and Video Retrieval*. Springer, 350–359.

Miriam Redi and Bernard Merialdo. 2012. A multimedia retrieval framework based on automatic graded relevance judgments. In *Advances in Multimedia Modeling*. Springer, 300–311.

Jitao Sang, Changsheng Xu, and Dongyuan Lu. 2012. Learn to personalized image search from the photo sharing websites. *Multimedia, IEEE Transactions on* 14, 4 (2012), 963–974.

Börkur Sigurbjörnsson and Roelof Van Zwol. 2008. Flickr tag recommendation based on collective knowledge. In *Proceedings of the 17th international conference on World Wide Web*. ACM, 327–336.

Cees G. M. Snoek and Marcel Worring. 2009. Concept-Based Video Retrieval. *Foundations and Trends in Information Retrieval* 4, 2 (2009), 215–322. Invited review paper, covering 300 references.

Abu Sayeed Md Sohail, Prabir Bhattacharya, Sudhir P Mudur, and Srinivasan Krishnamurthy. 2011. Classification of ultrasound medical images using distance based feature selection and fuzzy-SVM. In *Pattern Recognition and Image Analysis*. Springer, 176–183.

Zheng Sun, Dianxu Ruan, Yun Ma, Xiaolei Hu, and Xiao-guang Zhang. 2009. Crack defects detection in radiographic weldment images using FSVM and beamlet transform. In *Fuzzy Systems and Knowledge Discovery, 2009. FSKD'09. Sixth International Conference on*, Vol. 3. IEEE, 402–406.

Jinhui Tang, Qiang Chen, Meng Wang, Shuicheng Yan, Tat-Seng Chua, and Ramesh Jain. 2013. Towards optimizing human labeling for interactive image tagging. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)* 9, 4 (2013), 29.

Theodora Tsikrika, Christos Diou, Arjen P. de Vries, and Anastasios Delopoulos. 2009. Image annotation using clickthrough data. In *8th ACM International Conference on Image and Video Retrieval, CIVR*.

Theodora Tsikrika, Christos Diou, Arjen P. de Vries, and Anastasios Delopoulos. 2010. Reliability and effectiveness of clickthrough data for automatic image annotation. *Multimedia Tools and Applications* (2010). available online, to appear in press.

Triantafillos Tsirelis and Anastasios Delopoulos. 2011. Automatic ground-truth image generation from user tags. In *12th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS 2011)*.

Adrian Ulges, Markus Koch, Damian Borth, and Thomas M Breuel. 2009. Tubetagger-youtube-based concept detection. In *Data Mining Workshops, 2009. ICDMW'09. IEEE International Conference on*. IEEE, 190–195.

K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek. 2008. A Comparison of Color Features for Visual Concept Classification. In *ACM International Conference on Image and Video Retrieval*. 141–150.

Meng Wang and Xian-Sheng Hua. 2011. Active learning in multimedia annotation and retrieval: A survey. *ACM Transactions on Intelligent Systems and Technology (TIST)* 2, 2 (2011), 10.

Surong Wang, Manoranjan Dash, Liang-Tien Chia, and Min Xu. 2007. Efficient sampling of training set in large and noisy multimedia data. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)* 3, 3 (2007), 14.

Yongqiao Wang, Shouyang Wang, and Kin Keung Lai. 2005. A new fuzzy support vector machine to evaluate credit risk. *Fuzzy Systems, IEEE Transactions on* 13, 6 (2005), 820–831.

Kui Wu and Kim-Hui Yap. 2008. Soft-Labeling Image Scheme Using Fuzzy Support Vector Machine. In *Computational Intelligence in Multimedia Processing: Recent Advances*. Springer, 271–290.

Guang-ming Xian. 2010. An identification method of malignant and benign liver tumors from ultrasonography based on GLCM texture features and fuzzy SVM. *Expert Systems with Applications* 37, 10 (2010), 6737–6741.

Linjun Yang, Bo Geng, Alan Hanjalic, and Xian-Sheng Hua. 2012. A unified context model for web image retrieval. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)* 8, 3 (2012), 28.

Jun Zhang and Lei Ye. 2009. Content based image retrieval using unclean positive examples. *Image Processing, IEEE Transactions on* 18, 10 (2009), 2370–2375.

Lei Zhang and Yong Rui. 2013. Image searchfrom thousands to billions in 20 years. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)* 9, 1s (2013), 36.

Shiai Zhu, Chong-Wah Ngo, and Yu-Gang Jiang. 2012. Sampling and ontologically pooling web images for visual concept learning. *Multimedia, IEEE Transactions on* 14, 4 (2012), 1068–1078.