A Discriminative Kernel-based Model to Rank Images from Text Queries

David Grangier and Samy Bengio

***Abstract*— This paper introduces a discriminative model for the retrieval of images from text queries. Our approach formalizes the retrieval task as a ranking problem, and introduces a learning procedure optimizing a criterion related to the ranking performance. The proposed model hence addresses the retrieval problem directly and does not rely on an intermediate image annotation task, which contrasts with previous research. Moreover, our learning procedure builds upon recent work on the online learning of kernel-based classifiers. This yields an efficient, scalable algorithm, which can benefit from recent kernels developed for image comparison. The experiments performed over stock photography data show the advantage of our discriminative ranking approach over state-of-the-art alternatives (e.g. our model yields 26.3% average precision over the Corel dataset, which should be compared to 22.0%, for the best alternative model evaluated). Further analysis of the results shows that our model is especially advantageous over difficult queries such as queries with few relevant pictures or multiple-word queries.**

***Index Terms*— image retrieval, ranking, discriminative learning, kernel-based classifier, large margin**

I. Introduction

In this paper, we address the problem of retrieving pictures from text queries. In this task, the retrieval system is given a set of pictures and a few word query, it then outputs a picture ranking in which the pictures relevant to the query should appear above the others. This type of setup is common in several application domains, including web search engines, news wire services or stock photography providers. So far, the most widely-used approach to this problem consists in applying text retrieval techniques over a set of manually-produced captions that describe each picture. Although effective, this solution is expensive, as it requires a significant manual labeling effort.

Consequently, several automatic annotation approaches have been proposed in the literature. These approaches rely on a set of captioned pictures to learn a model, which can then predict textual annotations for any unlabeled picture. Two main types of auto-annotation models have been introduced: *concept classification models* and *bi-modal generative models*. In the case of concept classification, a classifier is learned for each vocabulary term, or concept, t. This classifier takes as input a picture and outputs a confidence value indicating whether the term t should occur in the predicted picture caption. This classification problem is typically addressed using Support Vector Machine (SVM) [33], [46] or boosting classifiers [43], as these large margin approaches enjoy good generalization properties [45]. In the case of bi-modal generative models, the training procedure learns a distribution estimating the joint probability $P(p, c)$ of a picture p (i.e. a set of visual features) and a caption c (i.e. a set of terms describing the picture). Given a test picture p, the learned distribution can then be used to infer the most likely caption, or a distribution over the whole vocabulary. Compared to concept classification, this generative approach hence learns a single model for all vocabulary terms, which notably yields a better modeling of term dependencies. Several bi-modal generative models have been proposed in the recent years, each model relying on different conditional independence assumptions between the observation of the text and the visual features.

Besides their differences, both concept classification and bi-modal generative models address the image retrieval problem through an intermediate task, auto-annotation. Image retrieval is performed by applying text retrieval techniques over the textual outputs of the auto-annotation model. Therefore, their learning procedure does not maximize a criterion related to the final *retrieval* performance, instead it maximizes a criterion related to the *annotation* performance. In this work, we adopt an alternative approach and introduce a model to learn an image retrieval model directly, without relying on auto-annotation. The proposed model, Passive-Aggressive Model for Image Retrieval (PAMIR), adopts a learning criterion related to the final retrieval performance, based on recent advances on discriminative learning for text retrieval [8], [17], [25]. PAMIR learning approach hence takes as input a set of training queries, as well as a set of pictures, and outputs a trained model likely to achieve high ranking performance on new data. Moreover, PAMIR also enjoys an efficient learning algorithm, which builds upon recent work on online learning of kernel-based classifiers [12]. The advantages of the proposed approach are several: our model parameterization can benefit from effective kernels for pictures comparison, while its optimization procedure permits an efficient learning over large training sets. Furthermore, our ranking criterion yields a discriminative retrieval model that does not rely on an intermediate annotation task, which is theoretically appealing [45]. These advantages are actually supported by our experiments, in which PAMIR is shown to outperform various state-of-the-art alternatives. For instance, the precision at top 10 of PAMIR reaches 10% over the Corel dataset [14], which should be compared to 9.3% for SVM for concept classification, the best alternative.

The remainder of this paper is organized as follows. Section II briefly describes previous related research. Section III introduces the proposed approach. Section IV presents the

Manuscript received December 22, 2006; revised September 12, 2007.

David Grangier is with the IDIAP Research Institute (Switzerland), Samy Bengio is with Google Inc. (USA). Part of this research has been performed while Samy Bengio was at the IDIAP Research Institute, part of this research has been performed while David Grangier was at Google Inc.

features used for image and query representation. This section also describes different picture kernels from which PAMIR could benefit. Section V reports the experiments comparing PAMIR to the alternative approaches. Finally, Section VI draws some conclusions.

II. Related Work

With the advent of the digital photography era, image retrieval has increasingly received attention. This study focuses on an important part of this research domain, the *query-by-text* task. This task aims at identifying the pictures relevant to a few word query, within a large picture collection. Solving such a problem is of particular interest from a user perspective since most people are used to efficiently access large textual corpora through text querying and would like to benefit from a similar interface to search collections of pictures. In this section, we briefly describe prior work focussing on this task.

So far, the *query-by-text* problem has mainly been addressed through automatic annotation approaches. In this case, the objective is to learn a model that can predict textual annotations from a picture. Such a model permits the retrieval of unlabeled pictures through the application of text retrieval techniques over the auto-annotator outputs. In the following, we briefly describe the two main types of approaches adopted in this context, *concept classification* and *bi-modal generative models*.

A. Concept Classification

Concept classification formulates auto-annotation within a classification framework. Each vocabulary term t, also referred as a *concept*, defines a binary classification problem, whose positive examples are the pictures for which the term t should appear in the predicted annotation. In this case, the learning procedure hence consists in training a binary classifier for each vocabulary term, and each classifier is learned to minimize the error rate of its concept classification problem.

Efforts in concept classification started with the detection of simple concepts such as indoor/outdoor [41], or landscape/cityscape [44]. Then, significant research has been directed towards detecting more challenging concepts, notably in the context of the TREC video benchmark [40]. Large sets of various concepts have then been addressed in recent work, such as [9], [10]. Nowadays, popular approaches in concept classification mainly relies on large margin classifiers, such as Support Vector Machines (SVM) [1], [33], [46] or boosting approaches [43]. SVM for concept classification constitutes the state-of-the-art for single word queries. In this application scenario, the images of the test corpus are ranked according to the confidence scores outputted by the classifier corresponding to the query term [33], [46]. However, in the case of multiple word queries, concept classifiers are more difficult to apply since the independent training of each concept classifier requires to further define fusion rules to combine the scores of the different concept classifiers [1], [10]. [1] compares different fusion strategies and concludes that, for query-by-text tasks, it is generally effective to compute the average of the score of the concept classifiers corresponding to the query terms, after having normalized their mean and variance. Therefore, we will adopt this fusion procedure latter in our experiments. As an alternative to such ad-hoc fusion strategies, bi-modal generative approaches have been introduced to learn a single model over the whole vocabulary, yielding a solution which can natively handle multiple-word queries.

B. Bi-Modal Generative Models

Contrary to concept classification, bi-modal generative approaches do not consider the different vocabulary words in isolation. Instead, these approaches model the joint distribution $P(c, p)$ of the textual caption (c) and the picture visual features (p), $P(c, p)$. The parameters of such a distribution are typically learned through maximum likelihood training, relying on a set of picture/caption pairs. After this learning phase, the retrieval of unlabeled pictures can be performed by ranking the pictures according to their likelihood $P(p|q)$ given query q, which is derived from the joint $P(q, p)$ through Bayes rule. Alternatively, it is also possible to estimate a conditional multinomial over the vocabulary $\{P(t|p), \forall t \in V\}$, for each unlabeled picture. This enables to retrieve pictures through the application of text retrieval techniques over the inferred multinomials. In this case, each multinomial $P(\cdot|p)$ is considered to represent a textual item, in which the number of occurrences of term t is proportional to $P(t|p)$. This alternative retrieval technique is generally preferred since it is more efficient (the multinomials need to be inferred only once for all queries) and it has shown to be more effective [31].

Several approaches based on the bi-modal generative framework have been proposed in the recent years. These models mainly differ in the types of distributions chosen to model textual and visual features, as well as in the way they model the dependencies between both modalities. In the following, we have chosen to briefly describe three such models, Cross-Media Relevance Model (CMRM) [22], Cross-Media Translation Table (CMTT) [35] and Probabilistic Latent Semantic Analysis (PLSA) [31]. A longer survey could also have described alternative models such as Multimodal Hierarchical Aspect Model [4], [3], Multiple Bernoulli Relevance Model [16] or Latent Dirichlet Allocation [5]. However, for the sake of brevity, we decided to focus on models that have shown to be the most effective over the Corel dataset [14].

Cross Media Relevance Model (CMRM) [22], is inspired by Cross-Lingual Relevance Model [28], considering caption of an image as the translation of its visual properties into words. In this model, it is assumed that the visual properties of an image are summarized as a set of discrete visual features. Formally, the visual features of a picture p are represented as a vector,

$$p = (tf^v_{1,p}, \ldots, tf^v_{|C|,p}),$$

where $tf^v_{i,p}$ refers to the number of features of type i in picture p and $|C|$ is the total number of feature types.

Given such a representation, CMRM infers a multinomial $P(t|p^{test})$ over the vocabulary for any test picture p^{test}. For that purpose, the joint probability of term t and all the visual elements of p^{test} is estimated by its expectation over the training pictures in P_{train},

$$P(t, p^{test}) = \sum_{j=1}^{|P_{train}|} P(j) \cdot P(t, p^{test}|j).$$

It is then assumed that terms and visual elements are independent given a training picture, leading to

$$P(t, p^{test}) = \sum_{i=1}^{|P_{train}|} P(j) \cdot P(t|j) \prod_{v=1}^{|C|} P(v|j)^{tf^v_{v,p^{test}}}. \quad (1)$$

In this equation, the probability of a training picture $P(j)$ is assumed to be uniform over the training set, i.e. $P(j) = 1/|P_{train}|$, while the probability of a term given a training picture $P(t|j)$ and the probability of a visual element given a training pictures $P(v|j)$ are estimated through maximum likelihood, smoothed with the *Jelinek-Mercer* method [22]. From (1), $P(t|p^{test})$ can then be estimated through Bayes rule, $P(t|p^{test}) = P(t, p^{test})/P(p^{test})$. Although simple, this approach has shown to be more effective compared to other approaches inspired by translation models, e.g. [14].

Cross Media Translation Table (CMTT) also builds upon cross-lingual retrieval techniques [35]. This model considers textual terms and discrete visual features, or *visterms*, as words originating from two different languages and constructs a translation table containing $P(t|v)$ for any pair of term/visterm (t, v). This table allows for the estimation of $P(t|p^{test})$ for any term t and any picture p^{test}:

$$P(t|p^{test}) = \sum_{i=1}^{m} P(t|v_i)P(v_i|p^{test}),$$

where $P(v_i|p^{test}) = \frac{tf^v_{i,p^{test}}}{\sum_{i=1}^m tf^v_{i,p^{test}}}$, and $v_1, \ldots, v_m$ are the visterms of p^{test}.

The translation table $\{P(t|v), \forall t, v\}$ is built from the training data D_{train} according to the following process. First, each term i (and each visterm j) is represented by a $|D_{train}|$ dimensional vector, t_i (v_j), in which each component k is the weight of term i (visterm j) in the k^{th} training example. As a noise removal step, the matrix $M = [t_1, \ldots, t_T, v_1, \ldots, v_V]$ containing all term and visterm vectors is approximated with a lower rank matrix, $M' = [t'_1, \ldots, t'_T, v'_1, \ldots, v'_V]$, through Singular Value Decomposition, and $P(j|i)$ is finally defined as

$$P(j|i) = \frac{\cos(t'_i, v'_j)}{\sum_{k=1}^{|V|} \cos(t'_i, v'_k)}.$$

Like CMRM, this method has also been evaluated over the *Corel* corpus [35], where it has shown to be effective. The use of Singular Value Decomposition has notably shown to improve noise robustness. However, CMTT has also some limitations. In particular, cosine similarity can only model simple relationships between terms and visual features. Approaches modeling more complex relationships, such as Probabilistic Latent Semantic Analysis [31], have subsequently been introduced.

Probabilistic Latent Semantic Analysis (PLSA) has first been introduced for text retrieval [20], before being extended to image retrieval [31]. This model introduces the following conditional independence assumption: "terms and discrete visual features are independent from pictures conditionally to an unobserved discrete variable $z_k \in \{z_1, \ldots, z_K\}$" ($z_k$ is called an *aspect* variable and the hyperparameter K is referred to as the number of aspects). In this framework, the probability of observing a term t or a visual feature v in a picture p follows

$$P(p, t) = P(p) \cdot \sum_k P(z_k|p)P(t|z_k), \quad (2)$$

$$P(p, v) = P(p) \cdot \sum_k P(z_k|p)P(v|z_k). \quad (3)$$

The different parameters of the model can be estimated relying on a two-step process. First, the probabilities $P(p), P(z_k|p)$ and $P(t|z_k)$ for all $p \in P_{train}$ are estimated to maximize the training caption likelihood through the Expectation Maximization algorithm. Then the probabilities $P(v|z_k), \forall v, k$ are fitted to maximize the training picture likelihood, keeping $P(p)$ and $P(z_k|p)$ fixed. For test pictures without caption, the probabilities $P(p), P(z_k|p)$ are estimated to maximize the picture likelihood, keeping $P(v|z_k), \forall(v, k)$ to the values estimated during training. After this procedure, (2) is applied to infer $P(p, t)$ for any test picture p and any term t. Similarly to CMRM, Bayes rule can then derive $P(t|p)$ from $P(p, t)$.

This model has several strengths: the latent aspect assumption allows one to model more complex dependencies between term and visual features, compared to CMRM or CMTT. Moreover, the two step training procedure biases the latent space toward the text modality, yielding better performance than less constrained latent models [31].

In absence of manual annotations, bi-modal generative models constitute the state-of-the-art for the retrieval of images from multiple-word queries, while, as mentioned above, concept classification is generally preferred for single word queries. However, one could wonder whether it is possible to provide a single solution for both settings. More fundamentally, one can also question the auto-annotation framework on which both types of approaches are based. In both cases, model training aims at solving an auto-annotation problem: for concept classification, the learning objective is to minimize the number of false positives (predicting a word which does not occur in the reference annotation) and false negatives (not predicting a word occurring in the reference annotation), while, for bi-modal generative models, the learning objective is to maximize the likelihood of the training picture/caption pairs. None of those criteria is tightly related to the final retrieval performance and there is hence no guarantee that a model optimizing such annotation objectives also yields good retrieval rankings.

In order to address those issues, we propose a *discriminative ranking model* for the query-by-text problem. The proposed approach is based on recent work on discriminative learning for the retrieval of text documents [8], [17], [25]. It learns a retrieval model with a criterion related to the ranking performance over a set of training queries. To the best of our knowledge, this is the first attempt to address the query-by-text problem directly, without solving an intermediate annotation problem.

III. Passive-Aggressive Model for Image Retrieval

This section introduces our discriminative model for the retrieval of images from text queries, *Passive Aggressive Model for Image Retrieval* (PAMIR). It first formalizes the query-by-text problem before introducing PAMIR parameterization and learning objective. Finally, it explains how the proposed linear model can be applied to infer non-linear decision functions relying on kernels.

A. Formalizing the Query-by-Text Problem

In the query-by-text problem, the retrieval system receives a text query q, from the text space $\mathcal{T}$, and a set of pictures P, from the picture space $\mathcal{P}$. It should then output a picture ranking in which the pictures relevant to q would ideally appear above the others, i.e.

$$\forall p^+ \in R(q,P), \forall p^- \in \overline{R}(q,P),\ rk(q,p^+) < rk(q,p^-) \quad (4)$$

where $R(q,P)$ refers to the set of pictures of P that are relevant to q, $\overline{R}(q,P)$ refers to the set of pictures of P that are not relevant to q and $rk(q,p)$ refers to the position of picture p in the ranking outputted for query q. Our goal is hence to learn a ranking model from training pictures P_{train} and queries Q_{train} such that the constraints of type (4) are likely to be verified over new pictures P_{test} and queries Q_{test}.

Similarly to most text retrieval approaches [2], we address this ranking problem relying on a scoring function F. This function $F : \mathcal{T} \times \mathcal{P} \rightarrow \mathbb{R}$ assigns a real value $F(q,p)$ expressing the match between any query q and any picture p. Our ranking approach is then simple: given a query q, we compute the score of each picture p in the picture set P, $\{F(q,p), \forall p \in P\}$, and order the pictures by decreasing scores. In this context, condition (4) translates to

$$\forall p^+ \in R(q,P), \forall p^- \in \overline{R}(q,P), F(q,p^+) > F(q,p^-), \quad (5)$$

and our objective comes down to learning a function F likely to verify (5) for unseen pictures P_{test} and queries Q_{test}. For that purpose, we introduce a parametric function F_w along with an algorithm to infer the parameter w from (P_{train}, Q_{train}), so that F_w is likely to achieve this objective.

B. Model Parameterization

The parameterization of F_w is inspired from text retrieval,

$$F_w : \mathcal{T} \times \mathcal{P} \rightarrow \mathbb{R}, \quad \text{where} \quad F_w(q,p) = q \cdot f_w(p),$$

f_w refers to a parametric mapping from the picture space $\mathcal{P}$ to the text space $\mathcal{T}$, and $\cdot$ refers to the dot product in the text space, which is commonly used to measure the matching between textual vectors [2]. In other words, our scoring function F_w measures the match between a picture p and a query q by first projecting the picture into the text space according to f_w, before measuring the match between the obtained textual vector $f_w(p)$ and the query q.

In the following, the form of f_w is first limited to linear mappings,

$$f_w : \mathcal{P} \rightarrow \mathcal{T}, \quad \text{where} \quad f_w(p) = (w_1 \cdot p, \ldots, w_T \cdot p) \quad (6)$$

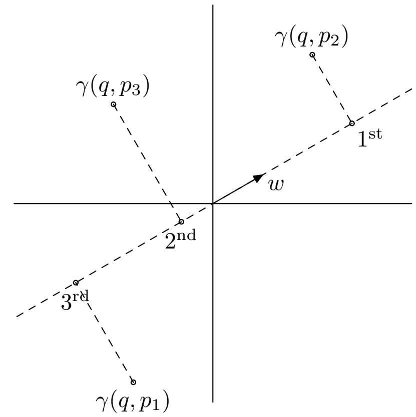

Fig. 1. PAMIR ranking strategy: in this example, the pictures of $\{p_1, p_2, p_3\}$ are ranked p_2, p_3, p_1 in answer to the query q. This figure illustrates that the pictures are ranked according to the order of the projections of $\{\gamma(q,p_1), \gamma(q,p_2), \gamma(q,p_3)\}$ along the direction of w.

and $w = (w_1, \ldots, w_T)$ is a vector of $\mathcal{P}^T$, T being the dimension of the text space $\mathcal{T}$. Section III-E then shows that the training procedure proposed thereafter can be extended to non-linear mappings through the kernel trick.

C. Large Margin Learning for our Ranking Problem

Our goal is to learn the parameter w such that F_w yields high ranking performance over unseen test queries. For that purpose, we first introduce a geometric interpretation of F_w, from which we can derive a margin maximization objective suitable to our ranking task.

For any query $q = (q_1, \ldots, q_T) \in \mathcal{T}$ and picture $p \in \mathcal{P}$, we define $\gamma(q,p)$ as the vector $(q_1 p, \ldots, q_T p)$ of $\mathcal{P}^T$ and rewrite $F_w(q,p)$ as $w \cdot \gamma(q,p)$, since

$$\begin{aligned} F_w(q,p) &= q \cdot f_w(p) = q \cdot (w_1 \cdot p, \ldots, w_T \cdot p) \\ &= \sum_{t=1}^{T} w_t \cdot (q_t p) = w \cdot \gamma(q,p). \end{aligned}$$

Hence, we can interpret $F_w(q,p)$ as the projection of $\gamma(q,p)$ onto the vector w. This means that PAMIR ranks the pictures of P according to the order of the projections of $\{\gamma(q,p), \forall p \in P\}$ along the direction of w, see Figure 1. With such an interpretation, one can easily remark that only the direction of w determines whether the constraints of type (5), $\forall q \in \mathcal{T},\ \forall p^+ \in R(q,P)$,

$$\forall p^- \in \overline{R}(q,P),\ w \cdot \gamma(q,p^+) - w \cdot \gamma(q,p^-) > 0,$$

are verified since the norm of w has no influence on the sign of $w \cdot \gamma(q,p^+) - w \cdot \gamma(q,p^-)$.

Hence, we can arbitrarily constrain the weight vector to lie on the unit circle $\mathcal{U}$, and solve our learning problem by finding a vector $u \in \mathcal{U}$ that verifies all training constraints. In other words, we want to select the weight vector in the set

$$\begin{aligned} \mathcal{S} = \{u \in \mathcal{U} \quad \text{s.t.} \quad & \forall (q,p^+,p^-) \in D_{train}, \\ & u \cdot \gamma(q,p^+) - u \cdot \gamma(q,p^-) > 0\} \end{aligned}$$

where D_{train} refers to all triplets (q,p^+,p^-) such that $q \in Q_{train}$, $p^+ \in R(q,P_{train})$, $p^- \in \overline{R}(q,P_{train})$.

When the training constraints are feasible ($\mathcal{S} \neq \emptyset$), any weight vector of $\mathcal{S}$ yields perfect retrieval performance over the training set. However, not all these solutions will yield the same results over some new test data. In order to select a vector of $\mathcal{S}$ likely to yield high generalization performance, we introduce the notion of *margin* for our ranking problem. For any vector $u \in \mathcal{S}$, we define its margin as

$$m(u) = \min_{(q,p^+,p^-)\in D_{train}} u \cdot \gamma(q,p^+) - u \cdot \gamma(q,p^-),$$

which is, by definition of $\mathcal{S}$, a positive quantity. This notion of margin is inspired from the definition introduced in [19] in the context of ranked categorization.

Equipped with this definition, we now explain why large margin solutions are preferable to ensure good generalization performance. Given a test triplet $(q_{test}, p^+_{test}, p^-_{test})$ composed of a query q_{test}, a picture p^+_{test} relevant to q_{test} and a picture p^-_{test} non-relevant to q_{test}, we define $R(q_{test}, p^+_{test}, p^-_{test})$ as the smallest quantity that satisfies $\exists(q_{train}, p^+_{train}, p^-_{train}) \in D_{train}$ s.t.

$$\begin{cases} \|\gamma(q_{train}, p^+_{train}) - \gamma(q_{test}, p^+_{test})\| < R(q_{test}, p^+_{test}, p^-_{test}) \\ \|\gamma(q_{train}, p^-_{train}) - \gamma(q_{test}, p^-_{test})\| < R(q_{test}, p^+_{test}, p^-_{test}). \end{cases}$$

This definition implies that, $\forall u \in \mathcal{S}$,

$$\begin{cases} |u \cdot \gamma(q_{train}, p^+_{train}) - u \cdot \gamma(q_{test}, p^+_{test})| < R(q_{test}, p^+_{test}, p^-_{test}) \\ |u \cdot \gamma(q_{train}, p^-_{train}) - u \cdot \gamma(q_{test}, p^-_{test})| < R(q_{test}, p^+_{test}, p^-_{test}) \end{cases}$$

since $\|u\| = 1$. Therefore,

$$\begin{array}{rl} u \cdot \gamma(q_{test}, p^+_{test}) - u \cdot \gamma(q_{test}, p^-_{test}) & \\ = & (u \cdot \gamma(q_{test}, p^+_{test}) - u \cdot \gamma(q_{train}, p^+_{train})) \\ - & (u \cdot \gamma(q_{test}, p^-_{test}) - u \cdot \gamma(q_{train}, p^-_{train})) \\ + & (u \cdot \gamma(q_{train}, p^+_{train}) - u \cdot \gamma(q_{train}, p^-_{train})) \end{array}$$

can be bounded as,

$$u \cdot \gamma(q, p^+_{test}) - u \cdot \gamma(q, p^-_{test}) > -2R(q_{test}, p^+_{test}, p^-_{test}) + m(u)$$

since $u \cdot \gamma(q, p^+_{train}) - u \cdot \gamma(q, p^-_{train}) > m(u)$ by definition of $m(u)$. Consequently, any solution $u \in \mathcal{S}$ for which the margin $m(u)$ is greater than $2R(q_{test}, p^+_{test}, p^-_{test})$ satisfies the test constraint $u \cdot \gamma(q, p^+_{test}) - u \cdot \gamma(q, p^-_{test}) > 0$.

Therefore, we decide to focus on the selection of the weight vector of $\mathcal{S}$ with the largest margin, as this weight is the most likely to satisfy all the constraints of a given test set,

$$u^* = \operatorname*{argmax}_{u\in\mathcal{S}} \; m(u).$$

This maximization problem is actually equivalent to the following minimization problem

$$\min_{u\in\mathcal{P}^T} \frac{1}{m(u)^2}, \text{s.t.}$$
$$\begin{cases} \|u\| = 1 \\ \forall (q,p^+,p^-) \in D_{train}, u \cdot \gamma(q,p^+) - u \cdot \gamma(q,p^-) > m(u) \end{cases}$$

and the introduction of the vector $w = \frac{1}{m(u)} u$ yields the following formulation of the same problem,

$$\min_{w\in\mathcal{P}^T} \|w\|^2,$$
$$\text{s.t. } \forall (q,p^+,p^-) \in D_{train}, w \cdot \gamma(q,p^+) - w \cdot \gamma(q,p^-) > 1.$$

This formulation of our retrieval problem is similar to the Ranking Support Vector Machine (RSVM) problem [25] introduced in the context of text retrieval, even if the notion of margin was not formalized as such in the case of RSVM.

Like for RSVM, we need to relax the training constraints for the non-feasible case ($\mathcal{S} = \emptyset$), which yields the following optimization problem,

$$\begin{array}{l} \displaystyle\min_{w\in\mathcal{P}^T} \quad \|w\|^2 \quad + \quad C \sum_{(q,p^+,p^-)\in D_{train}} \xi_{q,p^+,p^-}, \\ \text{s.t.} \quad \forall (q,p^+,p^-) \in D_{train}, \\ \qquad \begin{cases} w \cdot \gamma(q,p^+) - w \cdot \gamma(q,p^-) > 1 - \xi_{q,p^+,p^-} \\ \xi_{q,p^+,p^-} \geq 0 \end{cases} \end{array} \tag{7}$$

where the hyperparameter C controls the trade-off between maximizing the margin and satisfying all the training constraints. This problem (7) can equivalently be written as,

$$\min_{w\in\mathcal{P}^T} \|w\|^2 \; + \; C \sum_{(q,p^+,p^-)\in D_{train}} l(w;q,p^+,p^-),$$

where $\forall (q,p^+,p^-) \in D_{train}$,

$$l(w;q,p^+,p^-) = \max(0, 1 - w \cdot \gamma(q,p^+) + w \cdot \gamma(q,p^-)),$$

see [11].

D. An Efficient Learning Algorithm

The resolution of problem (7) involves a costly optimization procedure, if the RSVM approach is adopted. In fact, state-of-the-art techniques to solve this problem have a time-complexity greater than $O(|D_{train}|^2)$ [24], where $|D_{train}|$ denotes the number of training constraints. As we would like to handle large constraint sets, we derive an efficient training procedure by adapting the *Passive-Aggressive* (PA) algorithm, originally introduced for classification and regression problems [12]. For our ranking problem, PA should minimize

$$L(w; D_{train}) = \sum_{(q,p^+,p^-)\in D_{train}} l(w;q,p^+,p^-). \tag{8}$$

while keeping $||w||^2$ small.

For that purpose, the algorithm constructs a sequence of weight vectors $(w^0, \ldots, w^n)$ according to the following iterative procedure: the first vector is set to be zero, $w^0 = 0$ and, at the i^{th} iteration, the weight w^i is selected according to the i^{th} training example (q^i, p^{i+}, p^{i-}) and the previous weight w^{i-1},

$$w^i = \operatorname*{argmin}_{w} \; \frac{1}{2}\|w - w^{i-1}\|^2 + c \; l(w; (q^i, p^{i+}, p^{i-})). \tag{9}$$

Hence, at each iteration, we select the weight w^i as a trade-off between minimizing the loss on the current example $l(w; (q^i, p^{i+}, p^{i-}))$ and remaining close to the previous weight vector w^{i-1}. The *aggressiveness* parameter c controls this trade-off. Based on [12], it can be shown that the solution of (9) is

$$\begin{array}{rl} & w^i = w^{i-1} + \tau_i v^i, \\ \text{where} & \tau_i = \min\left\{c, \dfrac{l(w^{i-1}; (q^i, p^{i+}, p^{i-}))}{\|v_i\|^2}\right\} \\ \text{and} & v^i = \gamma(q^i, p^{i+}) - \gamma(q^i, p^{i-}). \end{array} \tag{10}$$

The hyperparameter c is selected to maximize the performance over some validation data D_{valid}. The number of iterations n is also validated: training is stopped as soon as the validation performance stops improving. This *early stopping* procedure actually allows one to select a good trade-off between satisfying all training constraints (i.e. minimizing the training loss $L(w; D_{train})$) and maximizing the margin (i.e. minimizing $\|w\|^2$). During the training process, it can be shown that, while the training error is decreasing [12], $\|w\|^2$ tends to increase, see Appendix. Hence, the number of iterations n plays a role similar to C in RSVM (7), setting the trade-off between margin maximization and training error minimization. The introduced PA algorithm therefore solves our learning problem with a time-complexity growing linearly with the number of iterations n. The observed complexity, reported later in Section V, actually shows that n grows much slower than $|D_{train}|^2$, a lower bound on RSVM time-complexity, enabling PAMIR to address much larger constraint sets.

E. Non-Linear Extension

Our model parameterization is based on a linear mapping f_w from the picture space $\mathcal{P}$ to the text space $\mathcal{T}$, see Eq. (6). This parameterization can be extended to non-linear mappings through the kernel trick, which allows PAMIR to benefit from effective picture kernels recently introduced in the computer vision literature, e.g. [48], [27], [30]. To kernelize PAMIR, we show that its parameterization solely requires the evaluation of dot products between picture vectors. For that purpose, we prove that, in the weight vector $w = (w_1, \ldots, w_T)$, each subvector $w_t,\ \forall t$, is a linear combination of training pictures. This then implies that the evaluation of

$$f_w(p) = (w_1 \cdot p, \ldots, w_T \cdot p),\ \forall p \in \mathcal{P},$$

only requires to compute the dot product between p and any training picture. The proof that, $\forall t$, the vector w_t is a linear combination of training pictures is performed by induction over the iterations of our training procedure: at the first iteration, the property is obviously verified since $w_t^0 = 0$, then the update preserves the property since, $w_t^i = w_t^{i-1} + \tau_i v_t^i$, where v_t^i is itself a linear combination of training pictures, $v_t^i = q_t^i\ (p^{i+} - p^{i-})$, see Eq. (10). Hence, at the last iteration n, $w_t = w_t^n$ verifies the property. This means that we can rewrite w_t as $w_t = \sum_{j=1}^{|P_{train}|} \alpha_{t,j} p_j$, where $\forall j,\ \alpha_{t,j} \in \mathbb{R}$. Consequently, we can introduce any kernel function $k : \mathcal{P} \times \mathcal{P} \to \mathbb{R}$, and rewrite f_w as,

$$\forall p \in \mathcal{P},\ [f_w(p)]_t = \sum_{j=1}^{|P_{train}|} \alpha_{t,j} k(p_j, p),$$

where $[f_w(p)]_t$ denotes the t^{th} component of the $f_w(p)$ vector. Practically, in this kernelized case, each w_t is stored as as a support set, consisting of pairs $(\alpha_{t,j}, p_j)$. The following section notably discusses different types of kernels suitable for our task.

This section has introduced PAMIR, a model suitable for image retrieval from text queries. This model has several advantages compared to the previous approaches presented in Section II: unlike SVM for concept classification, PAMIR can natively handle multiple-word queries, without requiring any fusion strategy; unlike bi-modal generative models, it relies on margin maximization training and hence enjoys good generalization properties [45]. More importantly, unlike both SVM for concept classification and bi-modal generative models, PAMIR training relies on a ranking criterion related to the final retrieval performance of the model. This criterion yields a discriminative retrieval model, which does not learn from textual annotations, but directly from training queries with pictures assessed for relevance.

IV. Text and Visual Features

This section introduces both the representation of text queries, and the representation of pictures, along with kernel functions suitable for picture comparison.

A. Query Representation

The *bag-of-words* framework is borrowed from text retrieval [2] for query representation. In this context, a vocabulary V is given prior to training to define the set of allowed words. Then, the *bag-of-words* representation neglects word ordering and assigns each query as a vector $q \in \mathbb{R}^T$, where T denotes the vocabulary size. The i^{th} component q_i of this vector is referred to as the weight of term i in the query q. In our case, it is defined as the *normalized idf* weighting scheme [2],

$$q_i = \frac{b_{i,q}\ idf_i}{\sqrt{\sum_{j=1}^{T} (b_{j,q}\ idf_j)^2}}$$

where $b_{i,q}$ is a binary weight, denoting the presence ($b_{i,q} = 1$) or absence ($b_{i,q} = 0$) of i in q, and idf_i is the inverse document frequency of i. This latter quantity is defined based on a reference corpus, such as an encyclopedia, and corresponds to $idf_i = -log(r_i)$, where r_i refers to fraction of corpus documents containing term i. This weighting hypothesizes that, among the terms present in q, the terms appearing rarely in the reference corpus are more discriminant and should be assigned higher weights.

B. Picture Representation

The representation of pictures for image retrieval is a research topic in itself, and different approaches have been proposed in the recent years, e.g. [16], [42], [43]. Contrary to the well-established bag-of-words representation for text data, there is not yet a single image representation that would be adequate for a wide variety of retrieval problems. However, among the proposed representations, a consensus is emerging on using *local descriptors* for various tasks, e.g. [29], [36]. This type of representation segments the picture into *regions of interest*, and extracts visual features from each region. The segmentation algorithm as well as the region features vary among approaches, but, in all cases, the image is then represented as a set of feature vectors describing the regions of interest. Such a set is often called a *bag-of-local-descriptors*.

This study also adopts the local descriptor framework. Our features are extracted by dividing each picture into overlapping

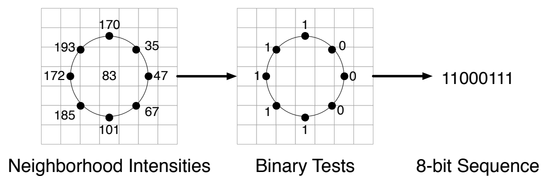

Fig. 2. An example of Local Binary Pattern ($LBP_{8,2}$). For a given pixel, the Local Binary Pattern is a 8-bit code obtained by verifying whether the intensity of the pixel is greater or lower than its 8 neighbors.

square blocks, and each block is then described with edge and color histograms. For edge histograms, we rely on *uniform Local Binary Patterns* [34]. These texture descriptors have shown to be effective on various tasks in the computer vision literature [34], [42], certainly due to their robustness with respect to changes in illumination and other photometric transformations [34]. Local Binary Patterns assign the texture histogram of a block by considering differences in intensity at circular neighborhoods centered on each pixel. Precisely, we use $LBP_{8,2}$ patterns, which means that a circle of radius 2 is considered centered on each block. For each circle, the intensity of the center pixel is compared to the interpolated intensities located at 8 equally-spaced locations on the circle, as shown on Figure 2, left. These eight binary tests (lower or greater intensity) result in an 8-bit sequence, see Figure 2, right. Hence, each block pixel is mapped to a sequence among $2^8 = 256$ possible sequences and each block can therefore be represented as a 256-bin histogram. In fact, it has been observed that the bins corresponding to non-uniform sequences (sequences with more than 2 transitions $1 \to 0$ or $0 \to 1$) can be merged, yielding more compact 59-bin histograms without performance loss [34].

Color histograms are obtained by k-means clustering. The color codebook is learned from the Red-Green-Blue pixels of the training pictures, and the histogram of a block is obtained by mapping each block pixel to the closest codebook color.

Finally, the histograms describing color and edge statistics of each block are concatenated, which yields a single vector descriptor per block. Our local descriptor representation is therefore simple, relying on both a basic segmentation approach and simple features. Of course, alternative representations could have been used, e.g. [16], [13], [43]. However, this paper focuses on the learning model, and a benchmark of picture representations is beyond the topic of this research.

C. Picture Kernels

Our model relies on a kernel function $k : \mathcal{P} \times \mathcal{P} \to \mathbb{R}$ over the picture space $\mathcal{P}$, as explained in Section III. Given our picture representation, we hence need a kernel to compare bags of local descriptors. Fortunately, several kernels comparing sets of feature vectors have been proposed along with the development of local descriptors [48], [27], [30].

Distribution Kernel approaches fit a distribution $p(v|p)$ over the space of local descriptors for each picture p, and then apply a kernel between distributions to compare pictures. Such kernels includes the Bhattacharya kernel or the expected likelihood kernel [21].

In this study, we fit a Gaussian Mixture Model for each picture p through Expectation-Maximization, as proposed in [30]. Motivated by scalability issues, we fit standard Gaussians on the input space, not kernelized Gaussian mixtures like [30]. The learned distributions are then compared with the Expected Likelihood Kernel (ELK),

$$k^{\text{ELK}}(p, p') = \int_v p(v|p)\ p(v|p')dv,$$

which can be computed in closed form for Gaussian mixtures [21], [30].

Matching Kernel approaches [48] rely on a *minor* kernel, k^l, that compares local descriptors. The kernel between two sets of local descriptors, $p = \{d_{p,i}\}_{i=1}^{|p|}$ and $p' = \{d_{p',i}\}_{i=1}^{|p'|}$, is defined as the average of the best-match-score between the descriptors of p and p',

$$k^{\text{match}}(p, p') = \frac{1}{2}\left[\hat{k}(p, p') + \hat{k}(p', p)\right],$$

$$\text{where} \quad \hat{k}(p, p') = \frac{1}{|p|}\sum_{i=1}^{|p|} \max_j k^l(d_{p,i}, d_{p',j}).$$

Formally, this function k^{match} is not a true Mercer kernel, since its Gram matrix is not always positive definite [6]. However, in practice, it can be used with SVM or PAMIR, without enjoying the same theoretical guarantee as a true kernel [6]. Empirically, SVMs relying on this kernel have shown to be effective over several object categorization tasks [6], [15], [48].

Visterm Kernel approaches explicitly represent the pictures in a high dimensional vector space, where the linear kernel is applied. For that purpose, each local descriptor of a picture p is represented as a discrete index, called *visual term* or *visterm*, and, like for text data, the picture is represented as a *bag-of-visterms* vector, in which each component p_i is related to the presence or absence of visterm i in p.

The mapping of the descriptors to discrete indexes is performed according to a codebook C, which is typically learned from the local descriptors of the training pictures through the k-means algorithm [14], [23], [36]. This study also applies this standard strategy. The assignment of the weight p_i of visterm i in picture p is classical as well,

$$p_i = \frac{tf_{i,p}^v\ idf_i^v}{\sqrt{\sum_{j=1}^{|C|}(tf_i^v\ idf_i^v)^2}},$$

where tf_i^v, the term frequency of i in p, refers to the number of occurrences of i in p, while idf_i^v, the inverse document frequency of i, is defined as $-log(r_i^v)$, r_i^v being the fraction of training pictures containing at least one occurrence of i.

Each of the presented kernels proposes a different technique to compare bags of local descriptors, whose effectiveness highly depends on the application context. For our task, we selected the most appropriate kernel through validation, as explained in the next section.

V. Experiments and Results

In this section, we present the experiments performed to evaluate PAMIR. We first describe our experimental setup, and then discuss the various issues related to hyperparameter

coast water lighthouse cliff   *people market food parasol*

Fig. 3. Examples of Corel pictures along with the associated captions.

TABLE I
Corel^Small Statistics

	train	valid	test
Number of pictures	4,000	500	500
Picture size		384x256 or 256x384	
Number of queries	7,221	1,962	2,241
Avg. # of rel. pic. per q.	5.33	2.44	2.37
Vocabulary size		179	
Avg. # of words per query	2.78	2.51	2.51

TABLE II
Corel^Large Statistics

	train	valid	test
Number of pictures	14,861	10,259	10,259
Picture size		384x256 or 256x384	
Number of queries	55,442	39,690	39,613
Avg. # of rel. pic. per q.	3.79	3.51	3.52
Vocabulary size		1,892	
Avg. # of words per query	2.75	2.72	2.72

selection, including the choice of a suitable kernel. Finally, we report the experimental results comparing PAMIR to the alternative models presented in Section II.

A. Experimental Setup

The datasets used for evaluation originate from stock photography, one of the application context of query-by-text image retrieval. Data from other domains, such as web search engine or newspaper archive, could also have been used. However, we decided to focus on stock photography, since the annotations associated with such pictures are generally produced by professional assessors with well defined procedures, which guarantees a reliable evaluation.

Two datasets are used in our experiments, `Corel`^`Small` and `Corel`^`Large`. Both sets originate from the *Corel* stock photography collection[1], which offers a large variety of pictures, ranging from wilderness scenes to architectural building pictures or sport photographs. Each picture is associated with a textual caption that depicts the main objects present in the picture, see Figure 3.

`Corel`^`Small` corresponds to the $5,000$-picture set presented in [14]. This set, along with the provided split between development and test data, has been used extensively in the query-by-text literature, e.g. [3], [23], [31]. It is composed of a $4,500$-picture development set P^s_{dev} and a 500-picture test set P^s_{test}. For model training and hyperparameter selection, we further divided the development set into a $4,000$-picture train set P^s_{train} and a 500-picture validation set P^s_{valid} (see Table I).

The queries needed to train and evaluate our model originate from the caption data. For that purpose, we first defined the relevance assessments considering that a picture p is relevant to a query q if and only if the caption of p contains all query words. Then, we defined the query set, Q^s_{train}, Q^s_{valid}, or Q^s_{test}, as the set containing all the queries for which there is at least one relevant picture in the picture set, P^s_{train}, P^s_{valid}, or P^s_{test}. This strategy defining queries and relevance assessments is hence not identical to a labeling in which a human assessor issues queries and labels pictures. However, it is based on manually produced captions and the resulting relevance information can be considered as reliable. In fact, there is no doubt that the pictures marked as relevant according to the definition above are indeed relevant, e.g. if the words *beach, sky* are present in a caption, it can confidently be claimed that the corresponding picture is relevant to the queries *"beach", "sky"* and *"beach sky"*. The only problem that could affect our relevance data is due to the possible incompleteness of some captions. If a word is missing from a caption, the corresponding picture will wrongly be marked as non-relevant for all queries containing this word. This weakness is however not specific to our labeling process. For instance, *system pooling*, the semi-automatic technique used for labeling data in retrieval benchmarks, also underestimates the number of relevant documents [2].

`Corel`^`Small` statistics are summarized in Table I. The datasets are used as follows: the parameter vector w is learned over $(P^s_{\text{train}}, Q^s_{\text{train}})$ through the training procedure defined in Section III. Hyperparameters, such as the number of training iterations, or the type of kernel used, are selected over $(P^s_{\text{valid}}, Q^s_{\text{valid}})$. Final evaluation is conducted over $(P^s_{\text{test}}, Q^s_{\text{test}})$. The training and evaluation of the alternative models is also performed over to the exact same data split, as it is the only way to conduct a fair comparison between the models [32].

The second dataset, `Corel`^`Large`, contains $35,379$ images and hence corresponds to a more challenging retrieval problem than `Corel`^`Small`. Like for the smaller set, `Corel`^`Large` pictures originate from the *Corel* collection and `Corel`^`Large` queries have been defined relying on the picture captions as explained above. The statistics of the training, validation and test sets of `Corel`^`Large` are reported in Table II.

For both datasets, performance evaluation has been conducted relying on standard information retrieval measures: average precision, precision at top 10, and break-even point [2]. For any query q, these measures evaluate the picture ranking outputted by the retrieval system as follows.

Precision at top 10 pictures ($\mathbf{P10}$) measures the percentage $Pr(10)$ of relevant pictures within the top 10 positions of the ranking. $\mathbf{P10}$ hence evaluates the percentage of relevant material a user would encounter on the first 10–result page of a search engine. Although it is easy to interpret, this measure tends to overweight simple queries with many relevant pictures when averaging over a query set. For such queries, it is easier

[1] Corel data are distributed through `http://www.emsps.com/photocd/corelcds.htm`.

to rank some relevant pictures within the top 10, simply because the relevance set is larger and not because of any property of the ranking approach.

Break-Even Point (**BEP**), often called R-Precision, measures the percentage $Pr(|R(q)|)$ of relevant pictures within the top $|R(q)|$ ranking positions, where $|R(q)|$ is the number of relevant pictures for the evaluated query q. Contrary to **P10**, this measure does not overweight queries with many relevant pictures.

Average Precision (**AvgP**) is the standard measure used for retrieval benchmark [2], and it corresponds to the average of the precision at each position where a relevant picture appears, $\mathbf{AvgP} = \frac{1}{|R(q)|} \sum_{p \in R(q)} Pr(rk(q,p))$, where $rk(q,p)$ is the rank of picture p for query q.

In the following, we report the performance of PAMIR and the alternative models as the average of these measures over the sets of test queries Q^s_{test} and Q^l_{test}.

B. Hyperparameter Selection

This section studies the enfluence of the hyperparameters on PAMIR performance. The feature extractor parameters, the type of kernel used, and the learning algorithm parameters are selected through validation: the model is trained with different parameter values over the training set and the parameters achieving the highest average precision over the validation set. For $\texttt{Corel}^{\texttt{Small}}$, all types of parameters are validated. For $\texttt{Corel}^{\texttt{Large}}$, only the learning parameters are validated for efficiency reasons, keeping the feature extractor and kernel parameters to the value selected over $\texttt{Corel}^{\texttt{Small}}$.

Feature extraction requires to select the block segmentation parameters (block size and block overlap) and the number of clusters used for color quantization. The block size determines the trade-off between obtaining local information (with small blocks) and extracting reliable statistics for each block (with large blocks), this parameter is selected through validation. Block overlap is set to half the block size such that all pixels belong to the same number of blocks, to avoid the predominance of pixels located at the block borders. The number of color bins is set to 50, as a trade-off between extracting a compact block representation and obtaining a perceptually good image reconstruction. Table III reports the validation performance for different block sizes. These results show that large blocks (> 128 pixels) are not suitable for our retrieval problem. In fact, it seems that considering less than 15 local descriptors per image does not provide PAMIR with enough statistics to address the retrieval task. The performance is stable for small blocks, between 32 and 96 pixels, with a slight advantage for 64 pixel blocks. We therefore pick this latter value for evaluation.

The selection of the kernel is also performed through validation. In fact, the different kernels comparing bag-of-local descriptors have been proposed recently and few studies focused on the empirical comparison of these approaches [15]. Table IV reports the best validation performance for each kernel, along with its parameters. Among the three kernels evaluated, the visterm kernel is clearly yielding the best performance, followed by the match kernel and then the Expected Likelihood Kernel. These results yields several remarks.

TABLE III
SELECTING THE BLOCK SIZE OVER $(Q^s_{\text{VALID}}, P^s_{\text{VALID}})$.

The other hyperparameters (kernel and learning parameters) are set to their optimal validation value.

block size	32	48	64	96	128	192	256
blocks per pic.	345	135	77	28	15	3	2
AvgP (valid.)	26.1	25.3	27.3	25.3	22.3	17.8	18.3

The Expected Likelihood Kernel (ELK) over Gaussian mixtures surprisingly yields its best results with only a single Gaussian per picture. This observation is not in line with the handwritten digit recognition experiments reported in [30]. Even if the differences in the datasets and the tasks performed might explain this difference, we further investigated on this point. In fact, the non-convex Expectation-Maximization procedure seems to explain the failure of ELK over Gaussian mixtures. The fitting of a mixture over the same picture with different initializations yield similar distributions in terms of data likelihood. However, these distributions are not equivalent for ELK evaluations and large relative variations are observed for a given pair of pictures, depending on the initialization of the Expectation-Maximization procedure for these pictures. This effect could possibly be reduced through averaging, if one fits multiple mixtures per picture. However, such a solution would be too costly for large datasets.

The performance of the match kernel is reported to be higher than the ELK. The match kernel relies on a *minor* kernel to compare pairs of local descriptors. In our experiments, the linear kernel, the Radial Basis Function (RBF) kernel, and the polynomial kernel have been tried as minor kernels. Table IV reports results only for the RBF kernel, which yielded the highest validation performance. Regarding efficiency, the match kernel is computationally demanding as it needs to compare all pairs of local descriptors between two pictures.

The visterm kernel is reported to yield the highest validation performance and optimal performance is reached with a codebook of $10,000$ prototypes. Moreover, the visterm approach also yields a more efficient model, compared to the other kernels. In fact, the visterm framework represents the pictures as bag-of-visterms vectors, where the linear kernel is applied. This means that the picture vectors can be pre-computed, as soon as the pictures are available. Then, model training and testing only require the evaluations of the linear kernel between sparse vectors. Such an operation can be performed efficiently as its complexity only depends on the number of non-zero components of the vectors (bounded by 77, the number of blocks per image), not on the data dimension ($10,000$, the codebook size) [2]. Furthermore, the linear kernel allows for handling w explicitly, which involves much less computation than handling support sets.

The training parameters of PAMIR are the number of iterations n and the aggressiveness c. Both of them sets the trade-off between the two learning objectives, i.e. minimizing the training loss and identifying a large margin model. Table V reports the selected values. For both $\texttt{Corel}^{\texttt{Large}}$ and $\texttt{Corel}^{\texttt{Small}}$, the number of iterations is significantly lower than the number of training constraints (e.g. for $\texttt{Corel}^{\texttt{Small}}$,

TABLE IV

SELECTING THE KERNEL OVER $(Q^s_{\text{VALID}}, P^s_{\text{VALID}})$.

The other hyperparameters (feature extractor and learning parameters) are set to their optimal validation value.

Kernel	$\mathbf{AvgP}$	Parameters
Exp. Likelihood	23.1	num. of Gaussians per picture (1)
Match	25.6	stdv of the local RBF kernel (5)
Visterm-Linear	27.3	codebook size $(10,000)$

TABLE V

SELECTING THE PARAMETERS OF THE LEARNING PROCEDURE.

The other hyperparameters (feature extractor and kernel parameters) are set to their optimal $\texttt{Corel}^{\texttt{Small}}$ validation value.

Dataset	Aggressiveness c	Num. of iter. n
$\texttt{Corel}^{\texttt{Small}}$	0.1	2.53×10^6
$\texttt{Corel}^{\texttt{Large}}$	0.1	1.55×10^7

TABLE VII

INDEXING TIMES FOR PAMIR AND THE ALTERNATIVES MODELS

Execution times have all been measured in seconds on the same machine (AMD Athlon64, 2.4Ghz, 2GB RAM).

	CMRM	CMTT	PLSA	SVM	PAMIR
$\texttt{Corel}^{\texttt{Small}}$	3	9	240	687	17
$\texttt{Corel}^{\texttt{Large}}$	849	4,099	1,025	24,650	450

TABLE VIII

RETRIEVAL TIMES FOR PAMIR AND THE ALTERNATIVES MODELS

All models have the same retrieval complexity. Execution times have all been measured on the same machine (AMD Athlon64, 2.4Ghz, 2GB RAM).

	CMRM	CMTT	PLSA	SVM	PAMIR
$\texttt{Corel}^{\texttt{Small}}$			0.34 ms per que		
$\texttt{Corel}^{\texttt{Large}}$			7.94 ms per que		

2.53×10^6 iterations should be compared to 1.45×10^8 training constraints). The algorithm hence converges before examining all the training set, which is certainly due to some redundancy in the training data. This highlights the efficiency of the PA approach, compared to other optimization techniques for SVM-like problems, as discussed in Section III.

To conduct a fair comparison, the alternative models have been trained over the same local descriptors and their hyperparameters have been selected with the same validation procedure. Namely, we selected the block size (for all models), the visual codebook size (for CMRM, CMTT and PLSA), and the kernel along with the corresponding parameters (for concept classification SVM) based solely on the validation set of $\texttt{Corel}^{\texttt{Small}}$, while all other parameters have been validated for both $\texttt{Corel}^{\texttt{Small}}$ and $\texttt{Corel}^{\texttt{Large}}$, see Table VI. Note that Table VI does not report the regularization parameter (C) for the SVM as it has been individually tuned for each term.

Before presenting the generalization performance, we briefly compare the computational time required by the different models, for both indexing and retrieval. Table VII reports the indexing times needed by PAMIR and the alternative models. *Indexing* corresponds to all the computations performed prior to the submission of the test queries, once the test pictures are available, excluding the operations related to feature extraction, such as visterm quantization. Indexing can hence be performed *off-line*, before the user can interact with the system. In the case of PAMIR, it includes the training step, plus the mapping of each test picture to the text space. For bi-modal generative models (CMRM, CMTT and PLSA), it corresponds to model training, plus the inference of $p(t|p)$ for each vocabulary term t and each test picture p. In the case of concept classification SVM, it corresponds to the training of an SVM for each vocabulary term, and the classification of each test image according to each of the trained SVMs. Table VII shows that our efficient training procedure yields an indexing time of the same order as the most efficient model, CMRM. This table also shows that SVM for concept classification is especially costly: this approach involves training a model for each vocabulary term, and each model training has a complexity that grows at least quadratically with the training set size [24]. This makes the application of this technique challenging for large datasets such as $\texttt{Corel}^{\texttt{Large}}$. Of course, the reported times highly depend on implementation details and optimization tricks[2], and should be considered carefully. It should also be noted that the reported times correspond to a single run of training, while, in a real-world usage scenario, a variable number of runs might be required depending on the number of hyperparameter values selected for validation. However, the results clearly indicate that indexing a corpus with PAMIR is not more costly than indexing a corpus with the other models. After indexing, all models then need to compute the dot-product matching between the submitted query and the textual representations inferred from the text pictures, before ranking the obtained scores. All models hence yield the *same* retrieval time, 0.34 msec per query for $\texttt{Corel}^{\texttt{Small}}$ and 7.94 msec per query for $\texttt{Corel}^{\texttt{Large}}$, on our reference machine, see Table VIII. This hence means that all models can be used interactively by the user, without any perceived delay.

C. *Experimental Results*

This section evaluates PAMIR and the alternative models over the test parts of $\texttt{Corel}^{\texttt{Small}}$ and $\texttt{Corel}^{\texttt{Large}}$.

Table IX, which reports the results over $\texttt{Corel}^{\texttt{Small}}$, shows that PAMIR outperforms all the alternative evaluated models. Compared to the best alternative, SVM, a relative improvement of 21% is reported for $\mathbf{AvgP}$ (26.3% for PAMIR versus 22.0% for SVM). Improvements are also observed for the other measures, $\mathbf{P10}$ and $\mathbf{BEP}$, which means that the use of PAMIR is advantageous for both users focussing on the first positions of the ranking (as shown by $\mathbf{P10}$ results) or users focussing on the whole ranking (as shown by $\mathbf{AvgP}$ results). One should note that the relatively low values reported for the $\mathbf{P10}$ results does not indicate a failure of the models but reflects the difficulty of the task: in fact, the optimal value for $\mathbf{P10}$ is 20.2% due to the low number of relevant pictures per query. This therefore means that the PAMIR user focussing only on the first ten results will retrieve about half the pictures he would have retrieved using the ideal ranker. In order to verify whether the observed advantage on the

[2] Our implementation of PAMIR is available at `www.idiap.ch/pamir/`.

TABLE VI

HYPERPARAMETERS FOR CMRM, CMTT, PLSA AND SVM

Model	Dataset	Hyperparameters
CMRM	$\texttt{Corel}^{\texttt{Small}}$	block size (192), visual codebook size (3,000), smoothing parameters ($\alpha = 0.5$, $\beta = 0.1$)
	$\texttt{Corel}^{\texttt{Large}}$	block size (192), visual codebook size (3,000), smoothing parameters ($\alpha = 0.2$, $\beta = 0.1$)
CMTT	$\texttt{Corel}^{\texttt{Small}}$	block size (256), visual codebook size (2,000), number of singular values kept (50)
	$\texttt{Corel}^{\texttt{Large}}$	block size (256), visual codebook size (2,000), number of singular values kept (1,000)
PLSA	$\texttt{Corel}^{\texttt{Small}}$	block size (32), visual codebook size (50,000), number of aspects (400)
	$\texttt{Corel}^{\texttt{Large}}$	block size (32), visual codebook size (50,000), number of aspects (600)
SVM	$\texttt{Corel}^{\texttt{Small}}$	block size (48), kernel (visterm kernel with a 20,000-visterm codebook)
	$\texttt{Corel}^{\texttt{Large}}$	block size (48), kernel (visterm kernel with a 20,000-visterm codebook)

TABLE IX

AVERAGED PERFORMANCE ON $\texttt{COREL}^{\texttt{SMALL}}$ TEST QUERIES

Bold numbers report when a model outperforms all others according to the Wilcoxon test at the 95% confidence level.

	CMRM	CMTT	PLSA	SVM	PAMIR
AvgP (%)	19.2	19.8	20.7	22.0	**26.3**
BEP (%)	13.1	13.7	12.8	13.8	**17.4**
P10 (%)	7.8	7.6	8.7	9.3	**10.0**

TABLE X

AvgP (%) FOR EASY AND DIFFICULT QUERIES OF $\texttt{COREL}^{\texttt{SMALL}}$

The 'easy' query set contains the 421 test queries with 3 or more relevant pictures, while the 'difficult' query set contains the 1,820 test queries with only 1 or 2 relevant pictures. Bold numbers report when a model outperforms all others according to the Wilcoxon test at the 95% confidence level.

	CMRM	CMTT	PLSA	SVM	PAMIR
Easy Queries	34.0	31.3	38.0	41.9	**43.3**
Difficult Queries	15.8	17.2	16.7	17.3	**22.4**

TABLE XI

AvgP (%) ON SINGLE & MULTI-WORD QUERIES OF $\texttt{COREL}^{\texttt{SMALL}}$

$\texttt{Corel}^{\texttt{Small}}$ contains 179 test queries with a single word and 2,062 queries with more than one word. Bold numbers report when a model outperforms all others according to the Wilcoxon test at the 95% confidence level.

	CMRM	CMTT	PLSA	SVM	PAMIR
Single-Word Que.	25.8	26.4	31.7	32.7	**34.0**
Multi-Word Que.	18.6	19.3	19.7	21.0	**25.7**

average results could be due to a few queries, we further ran the Wilcoxon signed rank test to compare PAMIR and each alternative model [38]. This test examines the distribution of the differences in the score obtained for each query and verifies whether it is symmetric around zero, which would mean that PAMIR has actually no advantage over the alternative approach. The test rejected this hypothesis at the 95% confidence level for all alternative models and all measures, as indicated by the bold numbers in the tables.

In order to compare the models over difficult and easy queries, we split the set of test queries into an 'easy' set, containing the queries with 3 or more relevant pictures in P^s_{test}, and a 'difficult' set, containing the queries with only one or two relevant pictures in P^s_{test}. Table X reports the average precision obtained over the two sets. PAMIR is shown to be the best model over both sets and its advantage is reported to be greater over the 'difficult' set (on this set, the relative **AvgP** improvement compared to SVM, the second best model, is +29%, as compared to +3.2% over the 'easy' set). This outcome is certainly due to PAMIR ranking criterion, since previous work showed that similar criteria for classification are especially adapted to unbalanced problems, i.e. classification tasks with a low percentage of positive examples [37].

As a further comparison, Table XI reports the average precision obtained over single and multiple-word queries separately. Several previous papers focused on single-word queries only, e.g. [23], [31], [35], and reporting those results allows for direct comparison with this literature. The single-word queries correspond to an easier task since the average number of relevant pictures per query is 9.4 for the single-word queries, compared to 1.8 for the multiple-word queries. The results reported in Table XI agree with this observation and all models are reported to reach higher performance on the single-word queries compared to multiple-word queries. On both query subsets, the advantage of PAMIR is confirmed. The PAMIR improvement is shown to be greater for multiple-word queries (+22.3% relative improvement in **AvgP** compared to the second best model, SVM) than for single-word queries (+4.0% relative improvement in **AvgP** compared to SVM). Two characteristics of PAMIR might explain this outcome: PAMIR training criterion has shown to be adapted to retrieval problems with few relevant pictures, which is the case of multiple-word queries. Moreover, PAMIR is the only model trained over multiple-word queries, which certainly helps achieving better performance over such queries. In fact, we observed that, for multiple-word queries, the other models often favor one of the query terms at the expense of the others. Figure 4 shows, for instance, that SVM favors the term 'car' at the expense of 'building' for the query 'building car'. On this example, the SVM ranking provides only one picture containing both cars and buildings, while PAMIR succeed in retrieving all the 3 relevant pictures in the top 5 positions. The PAMIR results even provide a non-relevant picture that could have been labeled relevant with looser labeling instructions (see the fifth picture of the ranking). The other example on Figure 4 is a single word query, 'petals'. It yields good results for both models, which retrieve 3 relevant pictures out of 4 in the top 5 positions. One can note a slight advantage for PAMIR that returns only flower-related pictures. Of course, these examples have limited statistical values but they give an idea on the type of ranking the user is facing.

With our setup, some queries appear in both the test and train sets (for instance, single-word queries are common to both sets). In order to verify the ability of PAMIR to generalize to new queries, we evaluated our model on the 601 test queries,

TABLE XII
RESULTS OVER TEST-ONLY QUERIES OF $\text{Corel}^{\text{Small}}$ QUERIES

Among the 2,241 test queries of $\text{Corel}^{\text{Small}}$, 601 queries are not appearing in the training or in the validation set. Bold numbers report when a model outperforms all others according to the Wilcoxon test at the 95% confidence level.

	CMRM	CMTT	PLSA	SVM	PAMIR
AvgP (%)	12.7	12.9	11.1	10.1	**16.1**
BEP (%)	7.1	6.3	4.1	3.1	**7.7**
P10 (%)	2.5	2.7	2.9	2.5	**3.5**

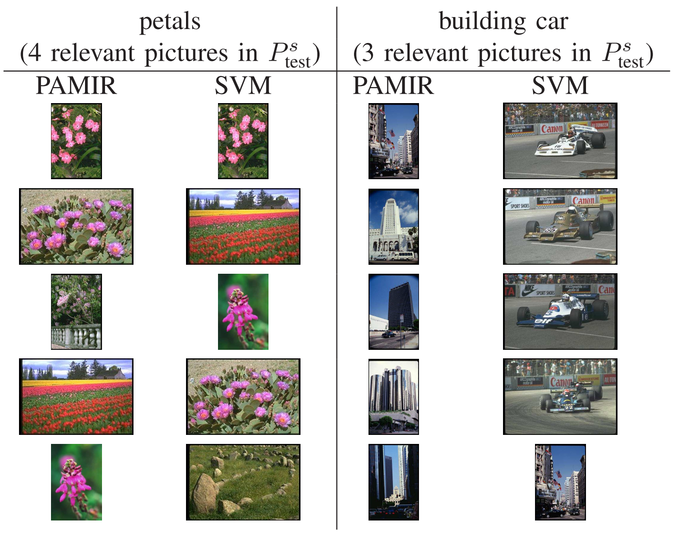

Fig. 4. Example: the top 5 pictures obtained with PAMIR and SVM, for two queries over $\text{Corel}^{\text{Small}}$. Higher resolution images, as well as other examples, are available at `www.idiap.ch/pamir/`.

which are not present in the training set. These queries can be considered as difficult, not only because the model has not seen pictures relevant to them during training, but also because they have very few relevant documents (1.03 on average). This second aspect can easily by explained if one remark that test queries with many relevant test pictures are also likely to have at least one relevant picture within the training data, which means that such queries are likely to belong to the training set as well. The results over this set of queries confirm the results observed on the whole set (see Table XII) and PAMIR is reported to outperform the alternative according to all measures. Moreover, for all models, the performance is much lower than for the 'difficult' query set (see Table X), which indicates that generalization to new queries deserves to be investigated further in the future.

Overall, the results over $\text{Corel}^{\text{Small}}$ outline the advantage of PAMIR over the alternative solutions. This outcome is certainly due to our discriminative learning strategy. The training of the other models either maximizes the joint picture/caption likelihood (CMTT, CMRM and PLSA) or minimizes the error rate of the per-term classification problems (SVM for concept classification), while our model relies on a ranking criterion, related to the final retrieval performance. This difference has shown to be especially helpful for both difficult queries (queries with few relevant pictures) and multiple-word queries.

TABLE XIII
AVERAGED PERFORMANCE ON $\text{Corel}^{\text{Large}}$ QUERIES

Bold numbers report when a model outperforms all others according to the Wilcoxon test at the 95% confidence level.

	CMRM	CMTT	PLSA	SVM	PAMIR
AvgP (%)	2.11	2.23	2.61	3.60	3.65
BEP (%)	1.26	1.46	1.69	1.81	**1.90**
P10 (%)	1.44	1.49	1.79	2.26	**2.53**

Table XIII reports the results of the experiments performed over $\text{Corel}^{\text{Large}}$. The reported performance over this set are much lower than for $\text{Corel}^{\text{Small}}$, which is not surprising considering the difficulty of the task. In $\text{Corel}^{\text{Large}}$, the relevant pictures account for 0.27 per thousand on average, which should be compared to 4.7 per thousand on average for $\text{Corel}^{\text{Small}}$. Moreover, the limited amount of relevant material present in the training set of $\text{Corel}^{\text{Large}}$ also makes this task more difficult: in $\text{Corel}^{\text{Large}}$, the average number of relevant pictures per training query is 3.79, which should be compared to 5.33 for $\text{Corel}^{\text{Small}}$ (see Table I and II). Hence, the models trained over $\text{Corel}^{\text{Large}}$ should address a more difficult ranking problem, while having seen less relevant pictures to generalize from. In fact, the statistics of $\text{Corel}^{\text{Large}}$ make this task closer to real world applications, such as image search for stock photography or news wire services, and the results over $\text{Corel}^{\text{Large}}$ are hence of a greater interest from a user perspective.

Although low, the results over $\text{Corel}^{\text{Large}}$ are much higher than random performance for all models (e.g. random performance is $\sim 0.03\%$ for $\mathbf{P10}$ which is much lower than 1.44%, the worst $\mathbf{P10}$ results, obtained with CMRM). All approaches can hence leverage from the training data. In fact, even if the models are far from optimal performance, they can still be useful to the user, as illustrated by the two queries shown on Figure 5. The first example 'tree snow people' corresponds to a relatively easy query with 13 relevant pictures in the test set. Like for the 'building car' example on $\text{Corel}^{\text{Small}}$, the SVM solution is dominated by one of the concepts, 'snow', at the expense of the others, and does not retrieve any relevant picture in the top 5. On the contrary, PAMIR, which is directly trained from multiple-word queries, yields high performance with 3 relevant pictures within the top 5 positions. The second query 'zebra herd' has less relevant pictures (4 in the test set). The results show a slight advantage for PAMIR: our model retrieves two relevant pictures at the third and fourth positions, while the SVM retrieves one relevant picture at the fifth position. This example illustrates that both models are often confused by similar pictures (savannah scenes in this case) for concepts with few training instances (only 22 pictures contain zebras among the 14,861 pictures of P^l_{train}).

Like for $\text{Corel}^{\text{Small}}$, the results in Table XIII clearly show the advantage of PAMIR over the other approaches. In fact, model comparison yields similar conclusions over $\text{Corel}^{\text{Small}}$ and $\text{Corel}^{\text{Large}}$: CMTT and CMRM reach comparable performance levels, PLSA performs better than the other generative models, but not as well as the SVM. Again, PAMIR yields the best results. Furthermore, the Wilcoxon

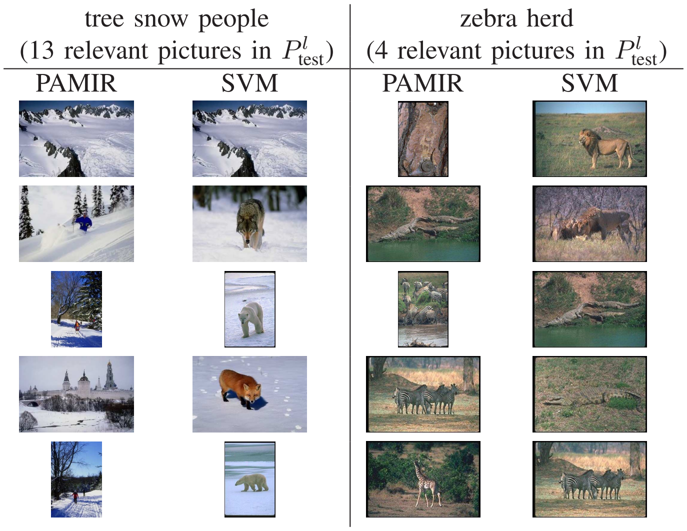

Fig. 5. Example: the top 5 pictures obtained with PAMIR and SVM for two queries over $\texttt{Corel}^{\texttt{Large}}$. Higher resolution images, as well as other examples, are available at `www.idiap.ch/pamir/`.

test over $\texttt{Corel}^{\texttt{Large}}$ concludes that PAMIR significantly outperforms each alternative, at the 95% confidence level, for $\mathbf{P10}$ and $\mathbf{BEP}$. For $\mathbf{AvgP}$, the test concludes that PAMIR outperforms all generative models (CMTT, CMRM and PLSA), and yields an $\mathbf{AvgP}$ similar to the SVM's. Overall, the results over both sets are consistent and show the advantage of our discriminative model over the alternatives.

VI. Conclusions

We have proposed a discriminative approach to the retrieval of images from text queries. In such a task, the model receives a picture corpus P and a text query q. It should then rank the pictures of P such that the pictures relevant to q appear above the others. Contrary to previous approaches that generally rely on an image auto-annotation framework, our learning procedure aims at selecting the model parameters likely to yield a high ranking performance over the unseen test data. For that purpose, we introduced a loss inspired from ranking SVM [25] and formalized the notion of margin for our retrieval problem. We then introduced a learning algorithm building upon Passive-Aggressive (PA) minimization [12]. The resulting model, Passive-Aggressive Model for Image Retrieval (PAMIR), has several advantages: its learning objective is related to the final retrieval performance, its training procedure is efficient for learning over large datasets, and the model parameterization can benefit from effective picture kernels recently introduced in the computer vision literature [27], [30], [48]. These advantages actually yield a model effective in practice, as shown by our experiments over stock photography data. For instance, over the standard *Corel* benchmark [14], PAMIR yields 26.3% average precision, which should be compared to 22.0% for SVM for concept classification, the best alternative. Our model has notably shown to be especially advantageous over multiple-word queries and difficult queries with few relevant pictures.

Although it outperforms the alternative models, PAMIR is far from reaching perfect performance, especially over the challenging $\texttt{Corel}^{\texttt{Large}}$ data. Therefore, we plan to investigate several directions to improve our model. First, we plan to modify PAMIR loss function to focus mainly on the top of the ranking, as most users examine only the first results. An approach derived from [26] could be applied to optimize measures like $\mathbf{P10}$. The loss could also be modified to optimize measures considering relevance assessments with gradual relevance levels, such as Discounted Cumulative Gain [47]. Another useful extension would be the prediction of a cut-off rank, that is, a ranking position below which the user is unlikely to encounter any relevant documents. Solutions inspired from [7] could help solving this problem. Finally, we also plan to investigate further on the use of kernels for local features. We want to model the spacial relationships between local features [39], and adopt a multi-resolution approach [18].

The proposed model, along with the reported results, hence advocate for addressing the image retrieval problem through a discriminative ranking approach, and open several possible directions of research to fully benefit from this formalism.

Appendix

This appendix shows that an upper bound on the norm $\|w^i\|$ grows with the number of iterations i of the Passive Aggressive algorithm. Precisely, it shows that $\|w^i\| \leq 2\ c\ \rho\ i$, where ρ corresponds to the radius of the training data $\rho = \max_{(q,p)\in D_{train}} \gamma(q,p)$.

The proof, inspired from [11], is conducted by induction over the iteration i. At the first iteration, the property is satisfied, since $w^0 = 0$. The update rule of w^t also preserves the property. If we assume the property to be verified at iteration $i-1$, i.e. $\|w^{i-1}\| \leq 2\ c\ \rho\ (i-1)$, we have $\|w^i\| \leq 2\ c\ \rho\ (i-1) + \|\tau_i v^i\|$, according to the update rule (10). By definition, τ_i is positive and smaller than c and hence $\|w^i\| \leq 2\ c\ \rho\ (i-1) + c\|v^i\|$. Furthermore, v^i is defined as $\gamma(q^i,p^{i+}) - \gamma(q^i,p^{i-})$, which implies that $\|v^i\| \leq 2\rho$. Consequently, $\|w^i\| \leq 2\ c\ \rho\ i$. This concludes the proof.

Acknowledgments

The authors thank Florent Monay for his advice. This work has been supported by the Swiss NSF through the MULTI project and the Swiss OFES through the PASCAL Network.

References

[1] A. Amir, G. Iyengar, J. Argillander, M. Campbell, A. Haubold, S. Ebadollahi, F. Kang, M. R. Naphade, A. Natsev, J. R. Smith, J. Tesic, and T. Volkmer. IBM research TRECVID-2005 video retrieval system. In *TREC Video Workshop*, 2005.

[2] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, Harlow, England, 1999.

[3] K. Barnard, P. Duygulu, D. Forsyth, N. de Freitas, D. M. Blei, and M. I. Jordan. Matching words and pictures. *Journal of Machine Learning Research (JMLR)*, 3, 2003.

[4] K. Barnard and D. Forsyth. Learning the semantics of words and pictures. In *International Conference on Computer Vision (ICCV)*, 2001.

[5] D.M. Blei, A.Y. Ng, and M.I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3, 2003.

[6] S. Boughorbel, J.P. Tarel, and F. Fleuret. Non-mercer kernels for svm object recognition. In *British Machine Vision Conference*, 2004.

[7] K. Brinker and E. Huellermeier. Calibrated label-ranking. In *NIPS Workshop on Learning to Rank*, 2005.

[8] C.J.C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *International Conference on Machine Learning (ICML)*, 2005.

[9] G. Carneiro and N. Vasconcelos. Formulating semantic image annotation as a supervised learning problem. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[10] S.-F. Chang, W. Hsu, W. Jiang, L. Kennedy, D. Xu, A. Yanagawa, and E. Zavesky. Trecvid-2006 video search and high-level feature extraction. In *TREC Video Workshop*, 2006.

[11] R. Collobert and S. Bengio. Links between perceptrons, MLPs and SVMs. In *International Conference on Machine Learning (ICML)*, 2004.

[12] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. Online passive-aggressive algorithms. *Journal of Machine Learning Research (JMLR)*, 7, 2006.

[13] Florent Monay D. Grangier and S. Bengio. Learning to retrieve images from text queries with a discriminative model. In *International Conference on Adaptive Multimedia Retrieval (AMR)*, 2006.

[14] P. Duygulu, K. Barnard, N. de Freitas, and D. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *European Conference on Computer Vision (ECCV)*, 2002.

[15] J. Eichhorn and O. Chapelle. Object categorization with SVM: Kernels for local features. Technical Report 137, Max Planck Institute, 2004.

[16] S.L. Feng and V. Lavrenko R. Manmatha. Multiple bernoulli relevance models for image and video annotation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004.

[17] D. Grangier and S. Bengio. Exploiting hyperlinks to learn a retrieval model. In *NIPS Workshop on Learning to Rank*, 2005.

[18] K. Grauman and T. Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *International Conference on Computer Vision (ICCV)*, 2005.

[19] R. Herbrich, T. Graepel, and K. Obermayer. Large margin rank boundaries for ordinal regression. In Peter J. Bartlett, Bernhard Schoelkopf, and Dale Schuurmans, editors, *Advances in Large Margin Classifiers*. MIT Press, Cambridge, MA, 2000.

[20] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42, 2001.

[21] T. Jebara and R. Kondor. Bhattacharyya and expected likelihood kernels. In *Conference on Learning Theory*, 2003.

[22] J. Jeon, V. Lavrenko, and R. Manmatha. Automatic image annotation and retrieval using cross-media relevance models. In *ACM Special Interest Group on Information Retrieval (SIGIR)*, 2003.

[23] J. Jeon and R. Manmatha. Using maximum entropy for automatic image annotation. In *International Conference on Image and Video Retrieval*, 2004.

[24] T. Joachims. Making large-scale support vector machine learning practical. In A. Smola B. Scholkopf, C. Burges, editor, *Advances in Kernel Methods: Support Vector Machines*. MIT Press, Cambridge, MA, 1998.

[25] T. Joachims. Optimizing search engines using clickthrough data. In *International Conference on Knowledge Discovery and Data Mining (KDD)*, 2002.

[26] T. Joachims. A support vector method for multivariate performance measures. In *International Conference on Machine Learning (ICML)*, 2005.

[27] R. Kondor and T. Jebara. A kernel between bags of vectors. In *International Conference on Machine Learning (ICML)*, 2003.

[28] V. Lavrenko, M. Choquette, and W. B. Croft. Cross-lingual relevance models. In *ACM Special Interest Group on Information Retrieval (SIGIR)*, 2002.

[29] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*, 60(2), 2004.

[30] Siwei Lyu. Kernels for unordered sets: the gaussian mixture approach. In *European Conference on Machine Learning (ECML)*, 2005.

[31] F. Monay and D. Gatica-Perez. PLSA-based image auto-annotation: constraining the latent space. In *ACM Multimedia*, 2004.

[32] H. Mueller, S. Marchand-Maillet, and T. Pun. The truth about corel: Evaluation in image retrieval. In *International Conference on Image and Video Retrieval*, 2002.

[33] M.R. Naphade. On supervision and statistical learning for semantic multimedia analysis. *Journal of Visual Communication and Image Representation*, 15(3), 2004.

[34] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 24(7), 2002.

[35] J. Y. Pan, H. J. Yang, P. Duygulu, and C. Faloutsos. Automatic image captioning. In *International Conference on Multimedia and Expo (ICME)*, 2004.

[36] P. Quelhas, F. Monay, J. M. Odobez, D. Gatica-Perez, T. Tuytelaars, and L. J. Van Gool. Modeling scenes with local descriptors and latent aspects. In *International Conference on Computer Vision (ICCV)*, 2005.

[37] A. Rakotomamonjy. Optimizing auc with support vector machine. In *European Conference on Artificial Intelligence Workshop on ROC Curve*, 2004.

[38] J.A. Rice. *Rice, Mathematical Statistics and Data Analysis*. Duxbury Press, 1995.

[39] J. Sivic, B.C. Russell, A.A. Efros, A. Zisserman, and W.T. Freeman. Discovering objects and their location in images. In *International Conference on Computer Vision (ICCV)*, 2005.

[40] A.F. Smeaton, P. Over, and W. Kraaij. Evaluation campaigns and TRECVid. In *ACM Workshop on Multimedia Information Retrieval (MIR)*, 2006.

[41] M. Szummer and R. W. Picard. Indoor-outdoor image classication. In *Workshop on Content-based Access of Image and Video Databases*, 1998.

[42] V. Takala, T. Ahonen, and M. Pietikainen. Block-based methods for image retrieval using local binary patterns. In *Scandinavian Conference on Image Analysis (SCIA)*, 2005.

[43] K. Tieu and P. Viola. Boosting image retrieval. *International Journal of Computer Vision (IJCV)*, 56(1), 2004.

[44] A. Vailaya and H. J. Zhang A. Jain. On image classification: city vs. landscape. In *Workshop on Content-based Access of Image and Video Libraries*, 1998.

[45] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, Berlin, 1995.

[46] J. Vogel and B. Schiele. Natural scene retrieval based on a semantic modeling step. In *International Conference on Image and Video Retrieval*, 2004.

[47] E. M. Voorhees. Evaluation by highly relevant documents. In *ACM Special Interest Group on Information Retrieval (SIGIR)*, 2001.

[48] C. Wallraven and B. Caputo. Recognition with local features: the kernel recipe. In *International Conference on Computer Vision (ICCV)*, 2003.

David Grangier received a master degree from the Eurecom Institute, and from the Ecole Nationale des Telecommunications de Bretagne (2003). He is currently a PhD candidate at the Ecole Polytechnique Federale de Lausanne, Switzerland. Since 2003, he also works as a research assistant at the IDIAP Research Institute, Switzerland. David Grangier's research focuses on various aspects of statistical machine learning, including learning to rank, learning distance measures and online learning. He has also strong interrests in pattern recognition algorithms for speech and vision problems.

Samy Bengio (PhD in computer science, University of Montreal, 1993) is a research scientist at Google since early 2007. Before that, he was a senior researcher in statistical machine learning at the IDIAP Research Institute, where he supervised PhD students and postdoctoral fellows working on many areas of machine learning such as support vector machines, time series prediction, mixture models, large-scale problems, speech recognition, multi channel and asynchronous sequence processing, multi-modal person authentication, brain computer interfaces, text mining, and many more. He is Associate Editor of the Journal of Computational Statistics, has been general chair of the Workshops on Machine Learning for Multimodal Interactions (MLMI'2004, 2005 and 2006), program chair of the IEEE Workshop on Neural Networks for Signal Processing (NNSP'2002), and on the program committee of several international conferences such as NIPS and ICML.