A Neural Network to Retrieve Images from Text Queries

David Grangier[1,2] and Samy Bengio[1]

[1] IDIAP Research Institute, Martigny, Switzerland,
firstname.lastname@idiap.ch,

[2] Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland

Abstract. This work presents a neural network for the retrieval of images from text queries. The proposed network is composed of two main modules: the first one extracts a global picture representation from local block descriptors while the second one aims at solving the retrieval problem from the extracted representation. Both modules are trained jointly to minimize a loss related to the retrieval performance. This approach is shown to be advantageous when compared to previous models relying on unsupervised feature extraction: average precision over *Corel* queries reaches 26.2% for our model, which should be compared to 21.6% for PAMIR, the best alternative.

1 Introduction

A system for the retrieval of images from text queries is essential to take full benefit from large picture databases such as stock photography catalogs, newspaper archives or website images. A widely used solution to this problem is to manually annotate each image in the targeted database and then use a text search engine over the annotations. However, this approach is time-consuming and hence costly, moreover it often results in incomplete and biased annotations which degrades retrieval performance. Therefore, several approaches to avoid this manual step have been proposed in the literature [1–4]. These approaches are either generative auto-captioning models or discriminative retrieval models. Generative auto-captioning models aims at inferring textual captions from pictures that can then be searched with a text retrieval system [1, 3, 4], while discriminative retrieval models do not introduce an intermediate captioning step and are directly trained to optimize a criterion related to retrieval performance [2].

In this work, a discriminative approach is proposed. This approach relies on a neural network composed of two main modules: the first module extracts global image features from a set of block descriptors, while the second module aims at solving the retrieval task from the extracted features. The training of both modules is performed simultaneously through gradient descent, meaning that image feature extraction and global decision parameters are inferred to optimize a retrieval criterion. This block-based neural network (BBNN) contrasts with previous discriminative models, such as [2], in which the extraction of image representation is chosen prior to training. This difference is shown to yield significant improvement in practice and BBNN is reported to outperform both generative and discriminative alternatives over the benchmark *Corel* dataset [5] (e.g. BBNN reaches 26.2% average precision over evaluation queries which should be compared to 21.6% for PAMIR, the best alternative, see Section 5).

The remainder of this paper is organized as follows: Section 2 briefly describes the related work, Section 3 introduces the proposed approach, Section 4 describes the text and visual features used to represent queries and images. Next, Section 5 presents the experiments performed over the benchmark *Corel* dataset. Finally, Section 6 draws some conclusions.

2 Related Work

As mentioned in introduction, most of the work in image retrieval from text queries focussed on generative models that attempt to solve the image auto-annotation task. These models include Cross-Media Relevance Models (CMRM) [3], Probabilistic Latent Semantic Analysis (PLSA) [4] and Latent Dirichlet Annotation (LDA) [1]. In general, these models introduce different conditional independence assumptions between the observation of text and visual features in an image and the parameters of the model, θ, are selected to maximize the (log) likelihood of some annotated training images, i.e.

$$\theta^* = \text{argmax} \sum_{i=1}^{N} \log P(p_i, c_i|\theta),$$

where $(p_1, \ldots, p_N)$ and $(c_1, \ldots, c_N)$ correspond to the N available training pictures and their captions. The trained models are then applied to associate a caption (or a distribution over text terms) to each of the unannotated test images and a text retrieval system is then applied over these textual outputs.

The training process of these models hence aims at maximizing the training data likelihood, which is not directly related to the targeted retrieval task, i.e. ranking a set of pictures P with respect to a query q such that the picture relevant to q appear above the others. Better performance can be achieved with a more suitable criterion, as recently shown by the discriminative model PAMIR [2]. To the best of our knowledge, the PAMIR approach is the first attempt to train a model to retrieve images from text queries through the optimization of a ranking criterion over a set of training queries. Previous discriminative models have only focussed on categorization ranking problems (e.g. [6, 7]), i.e. the task of ranking unseen images with respect to queries known at training time, which is not a true retrieval task in which an unseen query can be submitted.

In this work, we propose to train a neural network with a criterion similar to the one introduced in [2]. This neural network consists of two modules, the first one extracts an image representation from a set of local descriptors and the second one relies on the inferred representation to solve the retrieval problem. The training of both layers is performed jointly through gradient descent (see Section 3). This approach is inspired from convolutional neural networks (CNN) [8] which have been successfully applied to various classification/detection tasks [8, 9]: these models also formulate the identification of a suitable image representation and the classification from this representation as a joint problem. The proposed neural network hence contrasts with the PAMIR model for which the image representation is a-priori chosen. Our experiments over the benchmark *Corel* corpus show that this difference actually yields a significant improvement, e.g. P10 reaches 10.2% for BBNN compared to 8.8% for PAMIR (see Section 5).

3 A Neural Network for Image Retrieval

This section presents the loss function L adopted to discriminatively train an image retrieval model. It then describes the neural network proposed for image retrieval and its training procedure.

3.1 Discriminative Training for Image Retrieval

Before introducing a loss suitable for image retrieval, we should first recall the objective of a retrieval model: given a query q and a set of pictures P, a retrieval model M should ideally rank the pictures of P such that the pictures relevant to q appear above the others, i.e.

$$\forall q, \forall p^+ \in R(q), \forall p^- \notin R(q), rk_M(q, p^+) < rk_M(q, p^-), \tag{1}$$

where $R(q)$ is the set of queries relevant to q and $rk_M(q, p)$ is the rank of picture p in the ranking outputted by M for query q.

In order to achieve such an objective, retrieval models generally introduce a scoring function F that assigns a real value $F(q, p)$ to any query/picture pair (q, p). Given a query q, this function is used to rank the pictures of P by decreasing scores. In this case, the ideal property (1) hence translate to:

$$\forall q, \forall p^+ \in R(q), \forall p^- \notin R(q), F(q, p^+) > F(q, p^-). \tag{2}$$

In order to identify an appropriate function F from a set of training data, the following loss has been introduced [10],

$$\begin{aligned} L(F; D_{train}) &= \sum_{k=1}^{N} l(F; q_k, p_k^+, p_k^-) \\ &= \sum_{k=1}^{N} \max(0, \epsilon_k - F(q_k, p_k^+) + F(q_k, p_k^-)) \end{aligned} \tag{3}$$

where $\forall k, \epsilon_k > 0$ and D_{train} is a set of N triplets $\{(q_k, p_k^+, p_k^-), \forall k = 1, \ldots, N\}$ in which q_k is a text query, p_k^+ is a picture relevant to q and p_k^- is a picture non-relevant to q. This loss L can be referred to as a margin loss since it penalizes the functions F for which there exists training examples (q_k, p_k^+, p_k^-) for which the score $F(q_k, p_k^+)$ is not greater than $F(q_k, p_k^-)$ by at least a margin of ϵ_k. This loss has already been successfully applied to text retrieval problems [10, 11] and to image retrieval problems [2].

Regarding the choice of the margin value ϵ_k, two alternatives have been proposed previously [2]. A first option, *constant*-ϵ, is to set ϵ_k to be the same for all examples, e.g. $\forall k, \epsilon_k = 1$ (the value 1 is chosen arbitrarily here, any positive value would lead to the same optimization problem). Another option, *text*-ϵ, which can be applied only if the training pictures are annotated with textual captions, is to set ϵ_k to be greater than the difference of scores outputted by a text retrieval system F^{text}, i.e.

$$\epsilon_k = \max(\epsilon, F^{text}(q_k, c_k^+) - F^{text}(q_k, c_k^-)), \tag{4}$$

where c_k^+, c_k^- are the captions of the pictures p_k^+, p_k^- and $\epsilon > 0$. This second option has previously shown to be more effective [2] and will hence be used in the following.

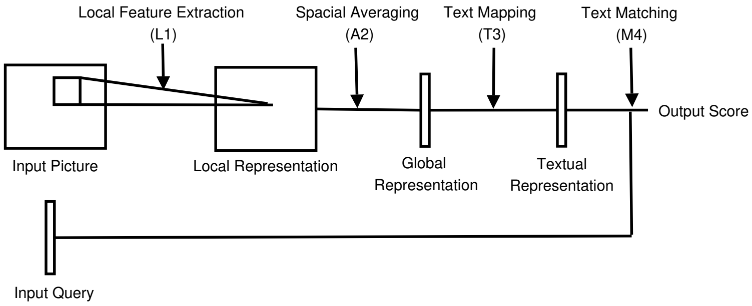

Fig. 1. The 4 successive layers of BBNN: local feature extraction (L1), spacial averaging (A2), text mapping (T3) and text matching (M4).

3.2 Block-Based Neural Network Architecture

As explained above, our goal is to identify a scoring function $q, p \rightarrow F(q, p)$ that minimizes $L(F; D_{train})$. For that purpose, we first introduce the block-based neural network (BBNN), $q, p \rightarrow F_w(q, p)$, and we then explain how the parameters w^* that minimize $w \rightarrow L(F_w; D_{train})$ are identified through stochastic gradient descent.

The proposed neural network is composed of 4 layers (see Figure 1): the local feature extraction layer $L1$, the averaging layer $A2$, the text mapping layer $T3$ and the query matching layer $M4$. The first layer $L1$ extracts local feature descriptors from different positions of the input picture p. The second layer $A2$ computes the average of the local feature vectors extracted by $L1$. The text mapping layer $T3$ then projects the output of $A2$ into the text space. The layer $M4$ finally compares the obtained textual vector with the input query q leading to the output $F(q, p)$. The layers are detailed as follows:

$L1$**: Local Feature Extraction** This layer extracts the *same* type of features at *different* positions of the input picture p through the following process: first, p is divided into B (possibly overlapping) blocks of the same size, $\{b_1, \dots, b_B\}$, and each block is assigned a vector representation, i.e. $b_i \in \mathbb{R}^{N_0}$ (see Section 4). The same parametric function is then applied over each block vector,

$$\forall i, f_i = \tanh(W_1 b_i + B_1),$$

where tanh is the component-wise hyperbolic tangent function, $W_1 \in \mathbb{R}^{N_1 \times N_0}$ and $B_1 \in N_1$ are the model parameters. The output dimension N_1 is a hyper-parameter of the model.

$A2$**: Spacial Averaging** This layer summarizes the B output vectors of $L1$ into a single N_1-dimensional vector through averaging:

$$f = \frac{1}{B} \sum_{i=1}^{B} f_i.$$

The succession of $L1$ and $A2$ is inspired from the bag-of-visterms (BOV) representation which has been widely used in computer vision in the recent years, e.g. [1, 12]. In this case, a first quantization layer maps each vector b_i

to a single discrete value among N_v, which is equivalent to map b_i to a N_v dimensional binary vector in which only one component is 1. In a second step, the input image is represented by a histogram through the averaging of its binary vectors. Here, we replace the quantization step by $L1$, which has two main advantages: first, the vectors f_i are continuous non-sparse vectors which allows to better model correlation between blocks. Second, the parameters of $L1$ are inferred jointly with the next layer parameters to solve the retrieval problem. This contrasts with the BOV approach in which the quantization parameters are generally inferred through generative learning (e.g. k-means clustering).

$T3$: **Text Mapping** This layer takes as input the representation f of picture p as outputted by $A2$. It then outputs a *bag-of-words* (BOW) vector t, i.e. a vocabulary-sized vector in which each component i represents the weight of term i in picture p (see Section 4 for further description on the BOW representation). This mapping from f to t is performed according to the parametric function:

$$t = W_3 \tanh(W_2 f + B_2) + B_3$$

where $W_2 \in \mathbb{R}^{N_2 \times N_1}$, $B_2 \in \mathbb{R}^{N_2}$, $W_3 \in \mathbb{R}^{V \times N_2}$ and $B_3 \in \mathbb{R}^{V}$ are the parameters of layer $T3$, V is the vocabulary size and N_3 is a hyperparameter to tune the capacity of $T3$.

$M4$**: Query Matching** This layer takes two BOW vectors as input: t, the output of $T3$ that represents the input picture p, and q, the input query. It then outputs a real-valued score s. This score is the inner product of t and q,

$$s = \sum_{i=1}^{V} t_i \cdot q_i .$$

This matching layer is inspired from the text retrieval literature in which text documents and text queries are commonly compared according to the inner product of their BOW representation [13].

This neural network approach is inspired from CNN classification models [8] for its first layers ($L1$, $A2$, $T3$) and from text retrieval systems for its last layer (i.e. BOW inner product). Like CNN for classification, our model formulates the problem of image representation and retrieval in a single integrated framework. Moreover like CNN, our parameterization assumes that the final task can be performed through the application of the same local feature extractor at different locations in the image. Our BBNN approach is however not a CNN strictly speaking: the local block descriptors b_i to which the first layer is applied do not simply consist of the gray level of the block pixels like in a CNN. In our case, we extract a N_0 dimensional feature vector summarizing color and texture statistics of the block, as explained in Section 4. This difference is motivated by two main aspects of our task: color information is helpful for image retrieval (see previous works such as [2]) and, moreover, the limited amount of training data prevents us from using a purely data-driven feature extraction technique (see Section 5 which depicts the small number of relevant pictures available for each query).

3.3 Stochastic Gradient Training Procedure

Stochastic gradient descent is the most widely used training technique for neural networks applied to large corpora. Its main advantages are its robustness with respect to local minima, and its fast convergence. We therefore decided to apply this optimization technique to identify the weight vector $w = [W_1; W_2; W_3; B_1; B_2; B3]$ that minimizes the loss $w \rightarrow L(F_w; D_{train})$, which yields the following algorithm:

> Initialize w.
> **Repeat**
> Pick $(q, p^+, p^-) \in D_{train}$ randomly with replacement.
> Compute the gradient $\frac{\partial l}{\partial w}(F_w; q, p^+, p^-)$.
> Update weights $w \leftarrow w - \lambda \frac{\partial l}{\partial w}(F_w; q, p^+, p^-)$.
> **Until** termination criterion.

It should be noted that this version of stochastic gradient training differs from the most used implementation in its sampling process [14]: we choose to sample a training triplet with replacement at each iteration rather than processing the samples sequentially in a shuffled version of the training set. While having no impact on the distribution of the examples seen during training, this difference avoids the costly shuffle for large triplet sets (e.g. there are $\sim 10^8$ triplets for the Corel dataset presented in Section 5).

The other aspects of the training process are more classical: the weight initialization is performed according to the methodology defined in [14] and early stopping is used as the termination criterion [14], i.e. training is stopped when performance over a held-out validation set D_{valid} stops improving. The learning rate λ is selected through cross-validation, as are the other hyperparameters of the model (i.e. N_1, N_2).

4 Text and Visual Features

In this section, we describe the bag-of-words representation used to represent text queries and the edge and color statistics used to represent image blocks.

4.1 Text Features

The text queries are assigned a bag-of-words representation [13]. This representation assigns a vector to each query q, i.e. $q = (q_1, \ldots, q_V)$ where V is the vocabulary size and q_i is the weight of term i in query q. In our case, this weight is assigned according to the well known *normalized tf idf* weighting, i.e.

$$q_i = tf_{q,i} \cdot idf_i,$$

where the term frequency $tf_{q,i}$ is the number of occurrences of i in q and the inverse document frequency idf_i is defined as $idf_i = -log(r_i)$, r_i referring to the fraction of training picture captions containing term i. It should be noted that this definition of *idf* hypothesizes that each training picture is labeled with a caption. This is the case for the *Corel* data used in our experiments (see Section 5). However, were such captions to be unavailable, it would still be possible to compute *idf* over another textual corpus, such as an encyclopedia.

Table 1. Query Set Statistics.

	Q_{train}	Q_{valid}	Q_{test}
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

4.2 Image Block Features

The image block descriptors b_i, on which the first layer of our model relies (see Section 3), summarizes edges and color statistics in the following manner.

Color information is represented through a N_C-bin histogram. This histogram relies on a codebook inferred from k-means clustering of the RGB pixels of the training pictures.

Edge information is represented through uniform Local Binary Pattern (uLBP) histograms. These histograms summarize texture information through the binary comparison of pixel intensities between each pixel and its eight neighbors. These features have shown to be effective over various computer vision tasks, including retrieval [15].

Color and edge histograms are then concatenated into a single block vector. Furthermore, a log-scale is adopted in the histograms, i.e. each pixels count c is replaced by $log(1 + c)$, since such non-linear scalings have already shown to be advantageous in previous work [16, 13].

5 Experiments and Results

This section first describes the experimental setup and then discusses the results.

5.1 Experimental Setup

The experiments presented in this section have been performed over the *Corel* dataset according to the setup defined in [5]. This setup has been widely used in the image retrieval community [3, 2, 4] and has become a kind of benchmark protocol for image retrieval. The data used consist of $4,500$ development pictures and 500 test pictures. The size of each picture is either 384×256 or 256×384. We further split the development set into a $4,000$-picture training set and a 500-picture validation set. This hence leads to three picture sets, P_{train}, P_{valid} and P_{test}. Each picture is further labeled with a caption relying on a 179-word vocabulary. These captions have been used for two purposes: for the definition of relevance assessments (i.e. we considered a picture to be relevant to a query q if its caption contained all query terms as explained in [2]) and for $text - \epsilon$ training (in this case, we used inner product of BOW vector as F^{text} function, see equation (4)).

The queries, Q_{train}, Q_{valid} and Q_{test}, used for training, validation and evaluation correspond to all subsets of the 179 vocabulary words for which there is at least one relevant picture within the training, validation or test pictures respectively. Table 1 summarizes query set statistics. The three query/picture datasets (Q_{train}, P_{train}), (Q_{valid}, P_{valid}) and (Q_{test}, P_{test}) have been respectively used to

Table 2. P10 and mean average precision (%) over test queries.

	CMRM	PLSA	PAMIR	BBNN
P10	5.8	7.1	8.8	**10.2**
AvgP	14.7	16.7	21.6	**26.2**

train the model (i.e. select the parameters that minimize the loss L), to select the model hyperparameters (i.e. the learning rate λ and the number of hidden units N_1, N_2) and to perform evaluation. For this evaluation, BBNN performance is measured with precision at top 10 (P10) and average precision (AvgP), the standard measures for information retrieval benchmarks [13]. These measures are complementary and evaluate different retrieval scenarios: P10 focuses on the first positions of the ranking, as the user of a web search engine would do, while AvgP focuses on the whole ranking, as an illustrator requiring all pictures about a specific theme would do. For any query, P10 measures the precision within top 10 positions (i.e. the percentage of relevant pictures within the 10 top-ranked pictures), while AvgP corresponds to the average of precision measured at each position where a relevant picture appears. Both measures have been averaged over the whole query set. BBNN has then been compared with the alternative models CMRM, PLSA and PAMIR which have been evaluated according to the same setup, as explained in [2].

Regarding picture preprocessing, 64×64 square blocks have been extracted every 32 pixels horizontally and vertically, leading to 77 blocks per picture. The size has been chosen as a trade-off between obtaining rich block statistics (i.e. having large blocks with many pixels) and extracting local patterns from the image (i.e. having many small blocks). The overlap of 32 pixels has been selected such that all pixels belong to the same number of blocks, which avoids the predominance of pixels located at the block borders. Concerning the color codebook size, we defined $N_C = 50$ which allows a perceptually good picture reconstruction while keeping the block histogram size reasonable. Although it would be more appropriate to select all these parameters through cross-validation, these a-priori choices already led to promising results, as reported in the next section.

5.2 Results

Table 2 reports the results obtained over the test queries. BBNN outperforms all other evaluated techniques for both measures. For AvgP, the relative improvement over CMRM, PLSA and PAMIR is respectively +78%, +57% and +21%. For P10, BBNN reaches 10.2%, which means that, on average, ~ 1 relevant picture appears within the top 10 positions. This number corresponds to good performance considering the low number of relevant pictures per query (2.37 on average, see Table 1). It fact, P10 cannot exceed 20.2% over *Corel* evaluation queries. In order to check whether the improvements observed for P10 and AvgP on the whole query set could be due to a few queries, we further compared BBNN results to those of the other models according to the Wilcoxon signed rank test [17]. The test rejected this hypothesis with 95% confidence for all models and both measures, which is indicated by bold numbers in the table. This means that BBNN consistently outperforms the alternative approaches on the test query set.

Table 3. P10 and mean average precision (%) over single-word test queries.

	CMRM	PLSA	PAMIR	BBNN
P10	17.8	21.3	25.3	**28.5**
AvgP	19.2	24.5	30.7	**35.0**

The results reported in Table 2 outline the effectiveness of discriminative approaches (PAMIR and BBNN) which both outperform the generative alternative (CMRM and PLSA). This shows the appropriateness of the selected loss function (3) for image retrieval problems. This outcome is in agreement with the text retrieval literature that recently reported good results with models relying on similar criteria [10, 16, 11].

As mentioned above, a difference in performance is also observed between the two discriminative models: BBNN is reported to outperform PAMIR (26.2% vs 21.6% AvgP). Since both models rely on the optimization of the same loss function, the observed difference is certainly due to the parameterization of the models. On one hand, PAMIR takes as input a bag-of-visterms representation of images, this representation being inferred from local descriptor through unsupervised clustering [2]. On the other hand, BBNN formulates the problem of representing images from local descriptors and the image retrieval task in a single integrated framework (see Section 3). This joint formulation allows the identification of a problem-specific image representation, which seems more effective than the bag-of-visterms representation.

Since several studies report results only for single word queries (e.g. [4, 5]), we also trained and evaluated the model over the subset of our train and test queries containing only 1 word. The results of this experiments are reported in Table 3. This evaluation further confirms the advantage of BBNN which yields a significant improvement is this case also. It should be noted that the difference observed between Table 2 and Table 3 does not mean that the retrieval models are more adapted to single-word queries: it only reflects the fact that single-word queries correspond to an easier retrieval problem (the average number of relevant documents per query is 2.4 for the whole Q_{test} set and 9.4 for its single-word query subset).

Overall, the results of both retrieval experiments confirm the advantage of supervised feature extraction that has already been observed with CNN over other tasks, such as classification or detection [8, 9].

6 Conclusions

We have introduced a discriminative model for the retrieval of images from text queries. This model relies on a neural network architecture inspired from convolutional neural networks [8]. The proposed network, Block-Based Neural Network (BBNN), formulates the identification of global image features from local block descriptors and the retrieval of images from such features as a joint problem. This approach is shown to be effective over the benchmark *Corel* dataset [5]. In particular, BBNN is reported to outperform both generative and discriminative state-of-the-art alternatives. For instance, the mean average precision over *Corel* test queries has been improved by 21% relative compared to the second best model PAMIR [2] (26.2% vs 21.6%). These results are promising and need to be confirmed over other datasets. It could also be interesting to extend the BBNN approach such that it could be applied to other retrieval problems, such as video retrieval.

Acknowledgments: This work has been performed with the support of the Swiss NSF through the NCCR–IM2 project. It was also supported by the PASCAL European Network of Excellence, funded by the Swiss OFES.

References

1. Barnard, K., Duygulu, P., Forsyth, D., de Freitas, N., Blei, D.M., Jordan, M.I.: Matching words and pictures. J. of Machine Learning Research **3** (2003)
2. Grangier, D., Bengio, S.: A discriminative approach for the retrieval of images from text queries. Technical report, IDIAP Research Institute (2006)
3. Jeon, J., Lavrenko, V., Manmatha, R.: Automatic image annotation and retrieval using cross-media relevance models. In: ACM Special Interest Group on Information Retrieval. (2003)
4. Monay, F., Gatica-Perez, D.: PLSA-based image auto-annotation: constraining the latent space. In: ACM Multimedia. (2004)
5. Duygulu, P., Barnard, K., de Freitas, N., Forsyth, D.: Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In: European Conf. on Computer Vision. (2002)
6. Tieu, K., Viola, P.: Boosting image retrieval. Intl. J. of Computer Vision **56**(1) (2004)
7. Wu, H., LuE, H., Ma, S.: A practical SVM-based algorithm for ordinal regression in image retrieval. In: ACM Multimedia. (2003)
8. LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D.: Handwritten digit recognition with a back-propagation network. In: Conf. on Advances in Neural Information Processing Systems. (1989)
9. Garcia, C., Delakis, M.: Convolutional face finder: A neural architecture for fast and robust face detection. T. on Pattern Analysis and Machine Intelligence **26**(11) (2004)
10. Joachims, T.: Optimizing search engines using clickthrough data. In: Intl. Conf. on Knowledge Discovery and Data Mining. (2002)
11. Grangier, D., Bengio, S.: Exploiting hyperlinks to learn a retrieval model. In: NIPS Workshop on Learning to Rank. (2005)
12. Quelhas, P., Monay, F., Odobez, J.M., Gatica-Perez, D., Tuytelaars, T., Gool, L.J.V.: Modeling scenes with local descriptors and latent aspects. In: Intl. Conf. on Computer Vision. (2005)
13. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. Addison Wesley, Harlow, England (1999)
14. LeCun, Y., Bottou, L., Orr, G.B., Mueller, K.R.: Efficient backprop. In Orr, G.B., Mueller, K.R., eds.: Neural Networks: Trick of the Trade. Springer (1998)
15. Takala, V., Ahonen, T., Pietikainen, M.: Block-based methods for image retrieval using local binary patterns. In: Scandinavian Conf. on Image Analysis. (2005)
16. Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., Hullender, G.: Learning to rank using gradient descent. In: Intl. Conf. on Machine Learning. (2005)
17. Rice, J.: Rice, Mathematical Statistics and Data Analysis. Duxbury Press, Belmont, California (1995)