Combining attributes and Fisher vectors for efficient image retrieval

Matthijs Douze, Arnau Ramisa, Cordelia Schmid

Combining attributes and Fisher vectors for efficient image retrieval

Matthijs Douze
INRIA, France
matthijs.douze@inria.fr

Arnau Ramisa
Institut de Robòtica i Informàtica Industrial (CSIC-UPC), Spain
aramisa@iri.upc.edu

Cordelia Schmid
INRIA, France
cordelia.schmid@inria.fr

Abstract

Attributes were recently shown to give excellent results for category recognition. In this paper, we demonstrate their performance in the context of image retrieval. First, we show that retrieving images of particular objects based on attribute vectors gives results comparable to the state of the art. Second, we demonstrate that combining attribute and Fisher vectors improves performance for retrieval of particular objects as well as categories. Third, we implement an efficient coding technique for compressing the combined descriptor to very small codes. Experimental results on the Holidays dataset show that our approach significantly outperforms the state of the art, even for a very compact representation of 16 bytes per image. Retrieving category images is evaluated on the "web-queries" dataset. We show that attribute features combined with Fisher vectors improve the performance and that combined image features can supplement text features.

1. Introduction

The problem of retrieving images of particular objects from large datasets has recently received increasing attention [8, 17, 20]. Most approaches build on the bag-of-features (BOF) representation introduced in [22]. This initial approach quantizes local image descriptors into visual words, i.e., for each descriptor the nearest descriptor from a visual vocabulary (learnt by k-means on a training set) is selected. The BOF representation of the image is then the histogram of the number of descriptors assigned to each visual word. Fast access to BOF vectors is obtained by an inverted file.

Recent extensions speed up the assignment of individual descriptors to visual words [14, 19]. They also improve the accuracy by complementing the visual word index for a given descriptor with a binary vector [5] or by learning descriptor projections [20]. All these approaches store one index per local images descriptor, which is prohibitive for very large datasets.

In order to obtain more compact representations, several recent approaches rely on low dimensional representations, such as bag-of-features with a small number of visual words [8] or GIST descriptors [15], and compress them to obtain very compact codes. For example, the approaches [2, 23, 25] compress GIST descriptors by converting them to compact binary vectors. In [6, 8] local descriptors are aggregated into low-dimensional vectors and compressed to small codes. In [17] an image description based on Fisher vectors has been introduced and shown to outperform the bag-of-features representation for the same dimensionality.

Differently from the above approaches, Torresani et al. [24] learn a set of classifiers and use the scores of these classifiers to obtain a low dimensional description of the image. The classifiers are trained on an independent dataset obtained automatically from the Bing image search engine for categories from the LSCOM ontology [13]. This low dimensional representation is shown to speed up image categorization, as it allows training efficient classifiers such as linear support vector machines. Their approach is in spirit similar to image representations based on attributes [3, 10], where the attributes are semantic characteristics defined by humans, such as "is furry", "has head" or "has wheel". Classifiers for these attributes are built based on manually labeled images. Attributes are shown to allow for learning with few or even no example images for a category in transfer learning experiments.

In this paper, we demonstrate that the attributes of [24] give excellent results for retrieval of particular objects. A combination of attribute features with the Fisher vector significantly outperforms the state of the art of particular object retrieval and also improves the retrieval of category images. We also implement an efficient technique for compressing the combined descriptor, based on dimensionality reduction and product quantization [7]. Our compression scheme is shown to improve over the current state of the art for very small codes.

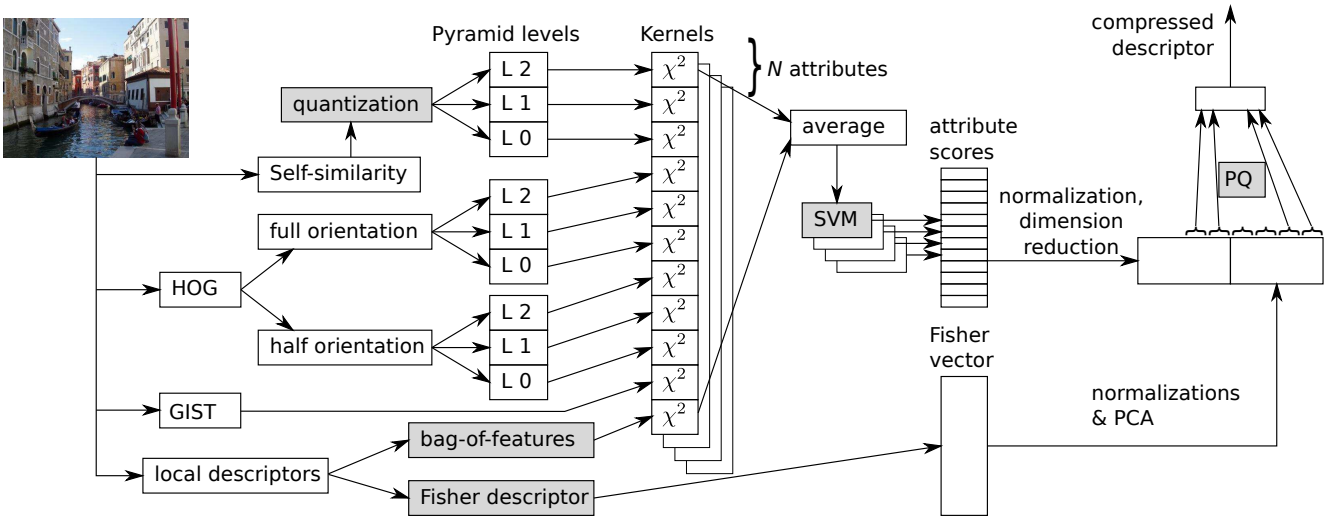

Figure 1. Computation of the attribute + Fisher descriptors for an image. Steps represented in gray require a learning stage.

2. Image description

In the following, we present the three image descriptors used in our experiments: the Fisher vector [17], the attribute features [24] and the text features. Figure 1 illustrates the computation of these features as well as the coding scheme described in section 3.

2.1. Fisher vector

Fisher vectors [16] are a means of aggregating local descriptors into a global descriptor. Local descriptors are computed by extracting orientation- and scale-invariant Hessian-affine interest points [12] and by describing their neighborhoods using the SIFT descriptor [11] (reduced to 64 dimensions by PCA). The position information of the points is not included in the descriptor.

During a preliminary learning stage, a 64-centroid Gaussian mixture model (GMM) was computed to fit the distribution of local descriptors in a dataset of unrelated images. The distribution of local descriptors of an image has a likelihood with respect to this GMM. The Fisher descriptor is the derivative of this likelihood with respect to the GMM parameters. Like [17], we restrict the parameters for which we compute derivatives to the means of the Gaussians, so our descriptor has $64 \times 64 = 4096$ dimensions.

Fisher descriptors were shown to outperform BOF as a global descriptor for image classification [16] and retrieval [17].

2.2. Attribute features

Each attribute corresponds to a term from a vocabulary. For an image, the attribute descriptor encodes how relevant each term is to describe that image. Attribute descriptors are computed from image classifiers built for each of the terms.

The vocabulary and learning set. We use the vocabulary from Torresani et al. [24], which contains $C = 2659$ attributes[1] obtained from the Large Scale Concept Ontology for Multimedia (LSCOM) [13].

The vocabulary includes names for object classes ("warplane", "logo", "hu jintao"), but also terms that are less related to visual representations ("democratic national conventions", "group of tangible things", "indoors isolated from outside") and a few abstract concepts ("attempting", "elevated", "temporal thing").

We also use the images of [24], collected as the top 150 images returned by the bing.com image search engine for each of these terms. Note that there is no manual cleaning of the data. This means that the returned images are noisy, i.e., they do not necessarily correspond to the term (assuming that the term can be represented by an image).

Low-level image features. We use the same features as [24]:

- Color GIST descriptor [15]. Orientation histograms are computed on a 4×4 grid over the entire image. They are extracted at 3 scales with 8, 8 and 4 orientation bins respectively .
- Pyramid of histograms of oriented gradients (PHOG) [1]. The PHOG descriptor first extracts Canny edges. It, then, quantizes the gradient orientation on the edges ($0°$ to $180°$) into 20 bins. Three spatial pyramid levels are used (1×1, 2×2, 4×4). Each level is used in an independent kernel. Differently from [24], we did not use the fourth level

[1] They call the attributes "classemes". The list of vocabulary terms is available at `http://www.cs.dartmouth.edu/~lorenzo/projects/classemes`.

of the PHOG, as it did not improve the classification results.

- PHOG descriptor with oriented edges. It is the same descriptor as the previous one, except that the direction of the gradient is not discarded (orientation in the range 0° to 360°) and there are 40 orientation bins. Again there are three spatial pyramid levels.
- Pyramid of self-similarity descriptors [21]. The self-similarity descriptor builds a log-polar histogram of correlation values between a central pixel and surrounding ones. The descriptor is computed densely every 5 pixels and quantized into 300 words. Frequency histograms are computed on three spatial pyramid levels.
- A bag-of-features descriptor. We use the same local features as in section 2.1 and a vocabulary of dimensionality 4000. The vocabulary was computed on the Bing dataset with k-means.

We have used the code available on-line for GIST[2], PHOG[3] and self-similarity[4]. To speed up our experimental evaluation, we have reimplemented these codes more efficiently.

Image classifiers. The classifiers for the attributes were obtained based on a standard non-linear binary SVM using LIBSVM with χ^2-RBF averaged kernels instead of the LP-β kernel combiner [4] used in [24]. We use a simple average of the classifiers, as learning the mixing weights was found to make little difference in practice. Similar to the approach of Torresani et al., the negative data includes one random image from each class except from the one that is being trained. The regularization parameters of the SVMs are selected via cross-validation.

The attributes. The image descriptor is a "class coding" of an image, obtained as the concatenation of the scores of the attribute classifiers. The scores can be positive or negative, they compare well using the L2 norm.

2.3. Textual features

Images are often associated with text. For example, on photo sharing sites there are tags and user comments, in photo banks indexing terms are associated with the images, and on random web pages, the text surrounding an image is likely to be relevant.

If a dataset has text associated with the images, we can build a basic text descriptor from these annotations. We remove punctuation and convert all text to lowercase, tokenize it into words and build a dictionary from all the words found in the corpus. We remove stopwords and words that are too rare in the corpus.

We, then, describe each image with a (sparse) histogram of the words appearing in its annotations. The histogram is L2-normalized and we apply TF-IDF weighting to favor infrequent words [22]. Histograms are compared with scalar products.

3. Indexing descriptors

Images are represented by global descriptors, i.e., Fisher vectors and attribute features. Retrieval consists in finding the nearest neighbors in a high-dimensional descriptor space. In this section we, first, describe how to combine descriptors and, then, how to search nearest neighbors efficiently.

3.1. Combining descriptors

To combine Fisher vectors and attribute features, each of them should be normalized. Normalization and comparison of Fisher vectors has been extensively studied in [17, 18]. We use the power normalization ($\alpha = 0.5$) [18] and normalize the vectors with the L2 norm.

Attribute vectors contain SVM classification scores. These scores are approximately Gaussian distributed. We have empirically observed that normalizing the vectors with L2 or L1 norm decreases the retrieval performance. For normalization we rely on the distribution of the descriptors extracted from n training images:

$$A = [a_1 \cdots a_n] \tag{1}$$

The mean description vector is significantly different from 0. We subtract it from the descriptors:

$$a'_i = a_i - \frac{1}{n}\sum_j a_j \tag{2}$$

To normalize attributes we compute the average vector norm on the training set, and normalize all descriptors with this single scalar:

$$\alpha = \frac{1}{n}\sum_j \|a'_j\| \qquad a^*_i = \frac{a'_i}{\alpha} \tag{3}$$

The normalized description matrix is then $A^* = [a^*_1 \cdots a^*_n]$.

Fisher vectors and attribute vectors can now be compared with the L2 distance, and have the same magnitude on average. To combine them, we simply add up the squared L2 distances. As shown in Figure 2, this improves the separation between matching and non-matching images. However, the performance can be improved by using a weighting factor to increase the contribution of the Fisher vector, see Table 1.

[2] http://people.csail.mit.edu/torralba/code/spatialenvelope/

[3] http://www.robots.ox.ac.uk/∼vgg/research/caltech/phog.html

[4] http://www.robots.ox.ac.uk/∼vgg/software/SelfSimilarity/

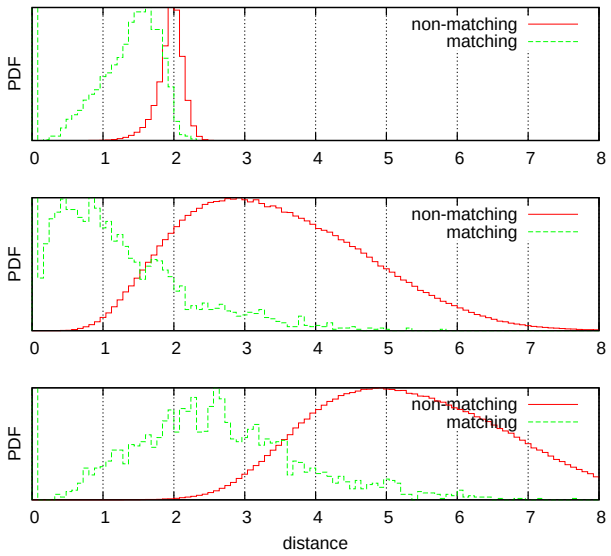

Figure 2. Probability densities for the squared distances between descriptors of matching and non-matching images in the Holidays dataset. Descriptors are: normalized Fisher descriptors (top), normalized attribute descriptors (middle), the A+F combined descriptor, without weighting (bottom).

Combining distances between features in this way is equivalent to concatenating the features into a single vector. We will use this property when compressing the features.

Text features are compared with scalar products. Therefore, they can be combined with the image-based features by subtracting the scalar products from the distances.

3.2. Dimension reduction

To accelerate retrieval, we project the vectors to a lower dimension. For non-sparse vectors up to a few thousand dimensions compared with the L2 distance, a good choice is to apply a PCA transform. In [8], the authors show that the dimensionality of VLAD descriptors (similar to Fisher) can be reduced using PCA with only a small loss of discriminative power.

For attribute vectors, we compute the singular values of A^*. The 64 (resp. 512) highest of the $C = 2659$ singular values account for 85 % (resp. 97 %) of the energy of the matrix. This implies that the vector components have a strong linear dependence, and, thus, the L2 distance between vectors is conserved after dimensionality reduction.

Dimensionality reduction of the attribute vector is also possible by selecting a subset of the dimensions. We have evaluated random selection and selection based on the cross-validation error of the classifiers as suggested by Torresani et al. [24]. Reducing the number of dimensions has the advantage over PCA that we do not need to evaluate all classifiers, which reduces the computational cost. We evaluate the different techniques for dimensionality reduction in the next section, see Figure 4.

3.3. Coding and searching

An additional improvement of efficiency and compactness can be obtained by encoding the image descriptors. To encode the dimensionality reduced vectors, we use the product quantization method of Jégou & al. [7]. The underlying principles are the following:

- The vectors of the database are encoded by vector quantization. They are represented by the index of the nearest centroid from a fixed vocabulary. The original vectors are not stored.
- Product quantization is used to obtain a very large vocabulary, which allows for a better distance approximation. Product quantization splits vectors into subvectors and quantizes them separately. The cost of the nearest-neighbor search decreases exponentially with the number of subvectors.
- The distance between a query vector and a vector of the database is approximated by the distance of the query vector to the centroid corresponding to the database vector. Thus, the distance computation is asymmetric: the query vector is not quantized, but the database vector is.

This method was shown to be very efficient for approximate nearest neighbor search with the L2 distance in high dimensional spaces for large datasets [8].

4. Experimental results

The experimental evaluation is performed in sections 4.1 and 4.2 for retrieval of particular objects (instance search), and in section 4.3 for retrieval of object categories.

4.1. Image retrieval of particular objects

We evaluate the retrieval of particular objects on the INRIA Holidays dataset [5]. This is a collection of 1491 holiday images, 500 of them being used as queries. The accuracy is measured by mean Average Precision (mAP).

Table 1 shows the performance of our attribute features on this dataset. We can observe that they obtain a performance on par with the state-of-the-art Fisher and VLAD descriptors with a somewhat lower dimensionality. Note that our implementation of the Fisher descriptor performs similarly to the authors' implementation [17].

The combination of the attribute features with the Fisher descriptor improves the performance by 10 percent over using the Fisher descriptor alone. For a similar dimensionality our descriptor significantly outperforms the state of the art. To obtain comparable results with a Fisher descriptor the dimension must be increased to 200k dimensions, i.e., 4096 components in the Gaussian mixture model.

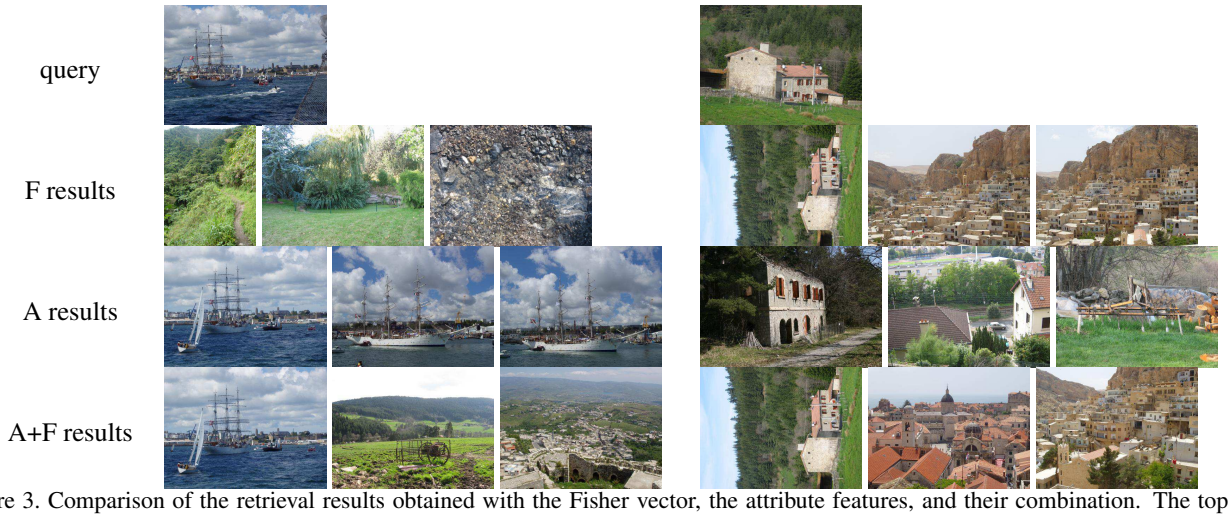

Figure 3. Comparison of the retrieval results obtained with the Fisher vector, the attribute features, and their combination. The top row shows the query image, the remaining rows the first three retrieved images for the different descriptors.

Descriptor	dimension	mAP
BOF k=1000 [6]	1000	41.1
Fisher k=64 [17]	4096	≈ 60
Fisher k=4096 [17]	262144	70.5
VLAD k=64 [8]	8192	52.6
Fisher (F), k=64, L2 dist.	4096	59.5
Attributes (A), L2 dist.	2659	55.0
A + F, F-weight $\times 1$	6755	64.5
A + F, F-weight $\times 2$	6755	69.5
A + F, F-weight $\times 2.3$	6755	69.9

Table 1. Comparison of the different descriptors and their combination on the Holidays dataset.

We also observe that the influence of the weighting factor is not critical. Values in the range between 1.5 and 2.5 produce very similar results. The Fisher vector is assigned a higher weight, which can be explained by the fact that we search for images of a particular object. In the following, we always use a weight of 2.3.

Figure 3 shows retrieval results for two example queries with the Fisher vector, the attribute features, and their combination. We can observe that the images retrieved with attribute features are more likely to represent similar categories.

4.2. Compression and indexing

This section evaluates the impact of dimensionality reduction and compression.

Dimension reduction. Figure 4 evaluates the performance of attribute features for different dimensionality reduction methods. The curve for PCA saturates rapidly. This indicates that the components have a strong linear dependence. This also implies that an arbitrary selection is possible, in particular if higher dimensional vectors are used. Note that in this case not all attributes need to be computed, which reduces computation time. We can also observe that selection with cross-validation does not improve over random selection. All methods obtain excellent performance if a dimension of 256 or higher is used.

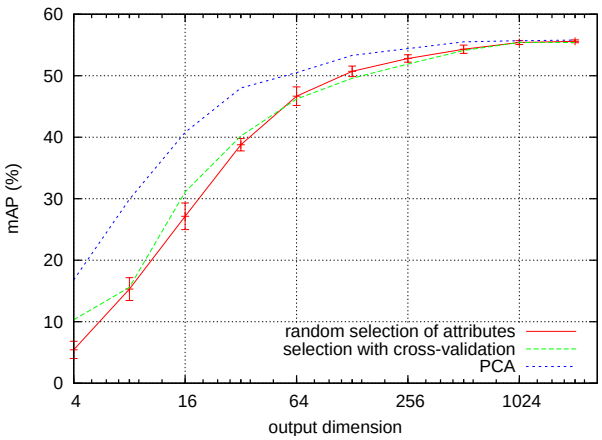

Figure 4. Comparison of different dimensionality reduction techniques for the attribute features on the Holidays dataset. mAP performance is displayed as a function of the dimension.

Table 2 evaluates the impact of the dimension on the combined A + F descriptor. Reducing the number of dimensions to 1024, i.e., by a factor 6, with PCA has almost no impact on the results. The performance with a vector of 128 dimensions (mAP=63.3) compares very favorably with the results reported in [8] for the same dimension (mAP=51.0). As observed in Figure 4, the reduction with

Dimension reduction			
Attributes	Fisher	dimension	mAP
PCA 512	PCA 512	1024	69.3
PCA 256	PCA 256	512	68.2
PCA 64	PCA 64	128	63.3
PCA 16	PCA 16	32	54.0
select random 256	PCA 256	512	67.9

Table 2. Dimensionality reduction for the combined descriptor on the Holidays dataset.

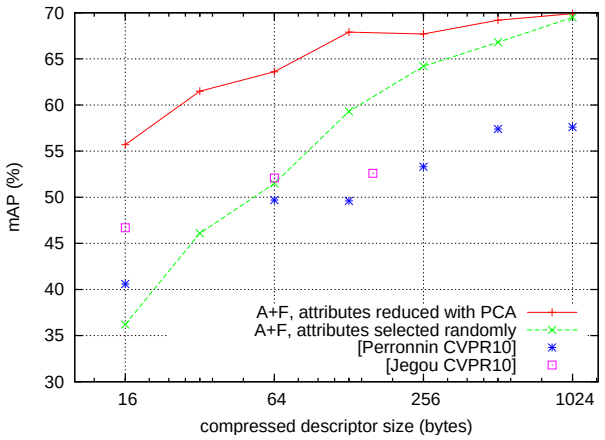

Figure 5. Performance of the A+F descriptor after dimension reduction and descriptor encoding on the Holidays dataset. The Fisher vectors are always reduced with PCA.

randomly selected features gives almost as good results as PCA for 256 dimensions.

Encoding. Figure 5 evaluates the encoding of descriptors in addition to dimension reduction. We can observe that our approach significantly outperforms that state of the art. For example, at 64 bytes, the results are more than 10 mAP points above the state of the art. For a code of 16 bytes, the mAP is at 56, which is excellent given the size of the code. Again it outperforms the state of the art by 10 mAP points. We can also observe that for small codes, the encoding increases the performance gap between dimension reduction by PCA and random attribute selection.

Large-scale experiments. To evaluate large-scale search, we merge the Holidays dataset with up to 1 million images downloaded from Flickr. For the attributes, we select a random subset of 256 attributes, in order to speed up computation. We project Fisher vectors to 256 dimensions with PCA.

Figure 6 shows that the combination improves significantly over the individual descriptors and that it degrades gracefully for growing datasets. We can also observe that a compression to 256 bytes results in almost no loss in performance and that results for a compression to 64 bytes are excellent.

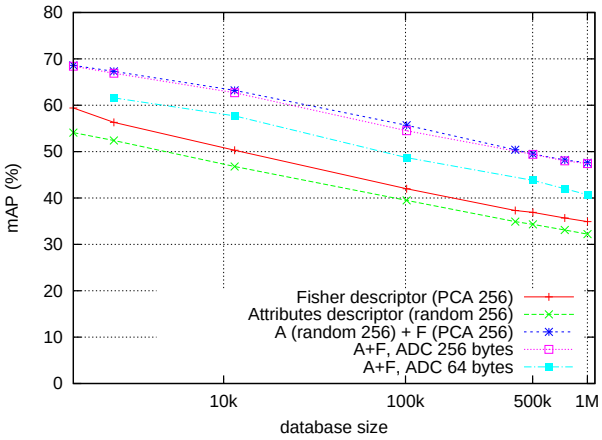

Figure 6. Performance on the Holidays dataset combined with one million distractor images from Flickr.

4.3. Image retrieval of categories

Dataset. We evaluate retrieval of categories on the "web-queries" dataset [9]. This dataset contains 67585 images retrieved by querying an image search engine with 353 short text queries ("concepts"). The query terms are diverse and representative of user requests (Table 3).

group	concepts		nb.
	examples	nb.	images
Person/people	Madonna, Spiderman	104	8721
Object/animal	violin, shark	64	4892
Landmark	Machu Picchu, Big Ben	55	5087
Specific image	Tux Linux, Guernica	54	3086
Cartoon	B. Simpson, Snoopy	18	1817
Scene	tennis court, forest	16	982
Type of image	painting, clipart	16	1543
Event	Cannes festival, race	11	242
Other	meal, family, GTA	15	1586
Total		353	27956

Table 3. The "web-queries" dataset. The 353 concepts are split in 9 groups. The number of concepts as well as relevant/positive images are indicated for each group.

The images of the dataset were retrieved from web pages that also contain text. The annotations provided with the dataset include: the page title, the text surrounding the image, and the alternate text for the image. After removal of stopwords and infrequent words, this results in a 24066-word vocabulary from which text descriptors can be computed (section 2.3).

The images were manually annotated as relevant (positive) or not to the text query. For our evaluation we use each

of the positive images as a query (27956 queries) in the entire dataset and evaluate the precision@10, i.e., the average percentage of true positives (same label) among the first 10 returned results. This measure is preferred to mAP, as precision is more important than recall if a user wants to browse a limited number of results.

concept	3 strongest attributes
george clooney	actor on TV, celebrity, actor in musicals
spider	insect, pit, invertebrate
dolphin	warplane, submarine, rowboat
forest	forest, garden, broadleaf forest
tower	bell tower, observation tower, minaret
mont blanc	glacier, snow skier, mountain
opera sydney	boat ship, warplane, patrol boat

Table 4. Relationship between web-queries concepts and attributes.

Concepts vs. attributes. Table 4 shows which attributes yield the highest classification scores for a few concepts of the web-queries dataset. The attribute scores are averaged over all images corresponding to a concept, including the ones that are annotated as not relevant. The dataset for training the classifiers (Bing dataset) and the web-queries dataset are retrieved in similar ways, i.e., text queries in an image search engine, but with different query terms and including false positives. The table shows that despite these limitations, the attributes give relevant semantic information about the concepts.

group	Fisher	Att.	Combination		Text
	(F)	(A)	A+F	A+F+T	(T)
Person/people	11.9	8.5	13.2	61.1	51.8
Object/animal	4.1	6.6	6.6	45.7	37.8
Landmark	18.2	20.8	27.8	72.4	62.6
Specific image	35.4	33.5	37.4	53.9	33.2
Cartoon	16.3	14.0	19.4	64.4	54.8
Scene	4.3	6.1	6.9	37.9	28.3
Type of image	9.0	10.2	12.2	45.7	31.7
Event	20.5	13.8	21.1	30.2	18.0
Other	30.1	27.3	32.5	65.6	50.2
Total	15.2	14.6	18.7	58.2	47.1

Table 5. Precision@10 for different descriptors (Fisher, attribute and text as well their combination) on the web-queries dataset.

Image results. Table 5 presents the precision@10 results for the web-queries dataset. The results for an image-based description are relatively low. This can be explained by the variety of the images and also underlines that today's image descriptors are far from sufficient for category search on a web-scale. We can observe that the method is good at recognizing images of a particular instance (logo, famous painting, etc.) and landmarks with more-or-less rigid structure, i.e., the scores are relatively high for the "Specific image" and "Landmark" groups. Figure 7 shows a few retrieval examples for the web-queries dataset with our A+F descriptor.

Combination with text. The precision@10 is significantly higher if querying with text descriptors. Interestingly, for "specific images" the image-based score outperforms the one obtained by text. Furthermore, we can observe that the combination of our A+F image features with text features improves significantly over the text only results, i.e., by more than 10%.

5. Conclusion

This paper has shown that attribute features, a high-level classification-based image representation, contribute to the task of image retrieval. Combining state-of-the-art Fisher vectors with attribute features improves the performance significantly, and this even with very compact codes. We can obtain a mAP of 56% with only 16 bytes on the Holidays dataset.

On a web queries dataset we again observe that the combination of attribute features with Fisher vectors helps, but the overall performance is rather low. This illustrates the difficulty of searching for category images given only one image. We have also observed that the combination of image and text improves significantly over text and image-based search alone.

The current selection and training procedure for attribute features is somewhat ad hoc. Future work includes the evaluation of different types of attribute features and training procedures. It would be interesting to see if "semantic" attributes as defined in [3] increase the performance, in particular for web queries.

Acknowledgements. This work was partially funded by the QUAERO project supported by OSEO, the European integrated project AXES, the ANR project GAIA, and by MICINN under project MIPRCV Consolider Ingenio CSD2007-00018. We would like to thank Florent Perronnin for providing the parameters of the GMM used to compute Fisher descriptors.

References

[1] A. Bosch, A. Zisserman, and X. Munoz. Representing shape with a spatial pyramid kernel. In *CIVR*, 2007.

[2] M. Douze, H. Jégou, H. Singh, L. Amsaleg, and C. Schmid. Evaluation of GIST descriptors for web-scale image search. In *CIVR*, 2009.

[3] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *CVPR*, 2009.

[4] P. Gehler and S. Nowozin. On feature combination for multiclass object classification. In *CVPR*, 2010.

[5] H. Jégou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *ECCV*, 2008.

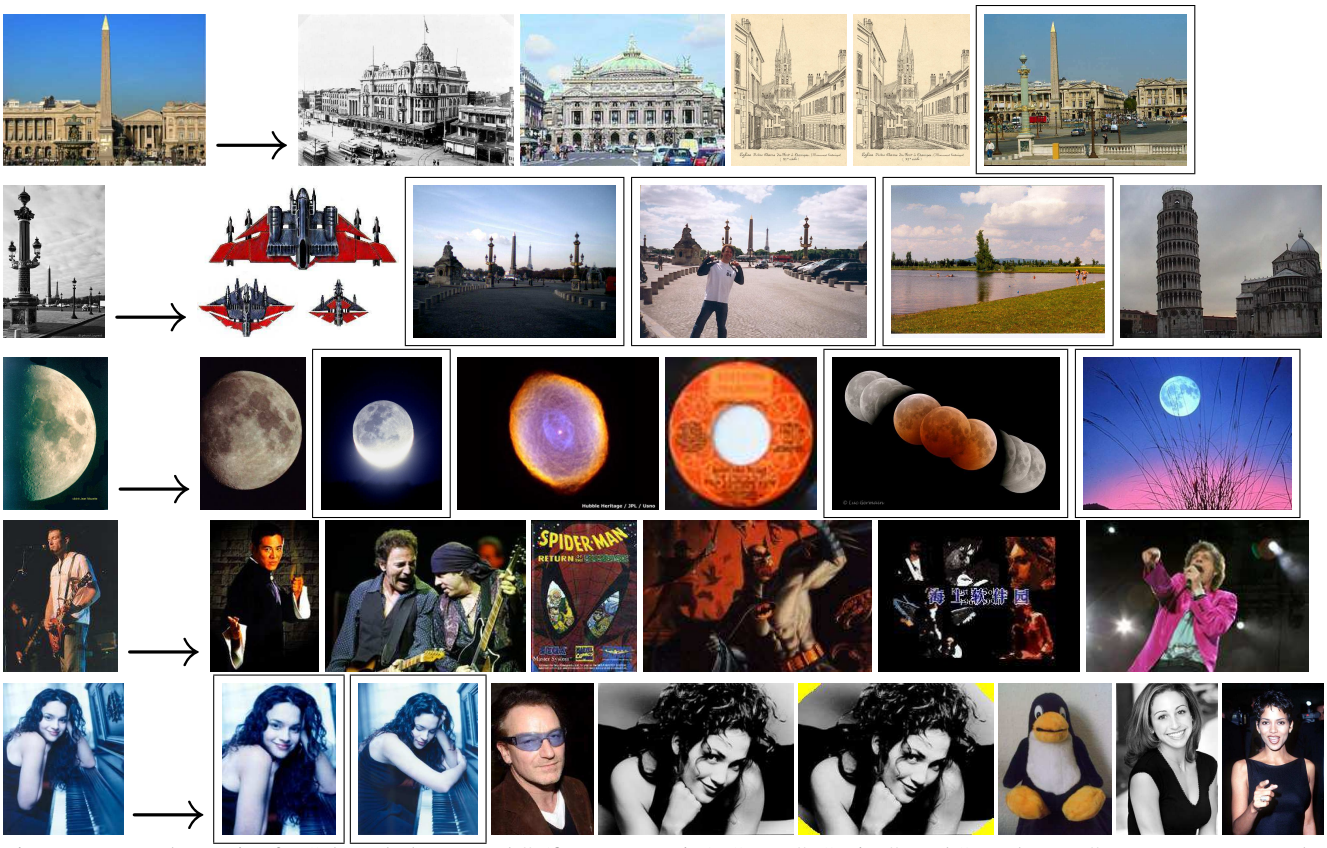

Figure 7. Example queries for "place de la concorde" (first two queries), "moon", "guitar", and "Norah Jones" concepts. We use the image-only A+F descriptor. Retrieval results are displayed in order. True positives are marked with a box.

[6] H. Jégou, M. Douze, and C. Schmid. Packing bag-of-features. In *ICCV*, 2009.

[7] H. Jégou, M. Douze, and C. Schmid. Product quantization for nearest neighbor search. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 33(1):117–128, jan 2011.

[8] H. Jégou, M. Douze, C. Schmid, and P. Perez. Aggregating local descriptors into a compact image representation. In *CVPR*, 2010.

[9] J. Krapac, M. Allan, J. Verbeek, and F. Jurie. Improving web-image search results using query-relative classifiers. In *CVPR*, 2010.

[10] C. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, 2009.

[11] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[12] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004.

[13] M. Naphade, J. Smith, J. Tesic, S. Chang, W. Hsu, L. Kennedy, A. Hauptmann, and J. Curtis. Large-scale concept ontology for multimedia. *IEEE Multimedia*, 13(3):86–91, 2006.

[14] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *CVPR*, 2006.

[15] A. Oliva and A. Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. *IJCV*, 42(3):145–175, 2001.

[16] F. Perronnin and C. Dance. Fisher kernels on visual vocabularies for image categorization. In *CVPR*, 2006.

[17] F. Perronnin, Y. Liu, J. Sanchez, and H. Poirier. Large-scale image retrieval with compressed Fisher vectors. In *CVPR*, 2010.

[18] F. Perronnin, J. Sanchez, and T. Mensink. Improving the Fisher kernel for large-scale image classification. In *ECCV*, 2010.

[19] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, 2007.

[20] J. Philbin, M. Isard, J. Sivic, and A. Zisserman. Descriptor learning for efficient retrieval. In *ECCV*, 2010.

[21] E. Shechtman and M. Irani. Matching local self-similarities across images and videos. In *CVPR*, 2007.

[22] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *ICCV*, 2003.

[23] A. Torralba, R. Fergus, and Y. Weiss. Small codes and large databases for recognition. In *CVPR*, 2008.

[24] L. Torresani, M. Summer, and A. Fitzgibbon. Efficient object category recognition using classemes. In *ECCV*, 2010.

[25] Y. Weiss, A. Torralba, and R. Fergus. Spectral hashing. In *NIPS*, 2008.