Comparing Local Feature Descriptors in pLSA-Based Image Models

Eva Hörster[1], Thomas Greif[1], Rainer Lienhart[1], and Malcolm Slaney[2]

[1] Multimedia Computing Lab, University of Augsburg, Germany
{hoerster,lienhart}@informatik.uni-augsburg.de
[2] Yahoo! Research, Santa Clara, CA, USA
malcolm@ieee.org

Abstract. Probabilistic models with hidden variables such as probabilistic Latent Semantic Analysis (pLSA) and Latent Dirichlet Allocation (LDA) have recently become popular for solving several image content analysis tasks. In this work we will use a pLSA model to represent images for performing scene classification. We evaluate the influence of the type of local feature descriptor in this context and compare three different descriptors. Moreover we also examine three different local interest region detectors with respect to their suitability for this task. Our results show that two examined local descriptors, the geometric blur and the self-similarity feature, outperform the commonly used SIFT descriptor by a large margin.

1 Introduction

Probabilistic models with hidden topic variables, originally developed for statistical text modeling of large document collections such as probabilistic Latent Semantic Analysis (pLSA) [1] and Latent Dirichlet Allocation (LDA) [2], have recently become popular in image content analysis tasks such as scene classification [3,4,5], object recognition [6], automatic segmentation [7] and image annotation [8]. In these approaches documents are modeled as mixtures of hidden topics under the assumption of a bag-of-words document representation. Applied to visual tasks, the mixture of hidden topics refers to the degree to which each object class, i.e. grass, people, sky, is contained in the image. In the ideal case, this gives rise to a low-dimensional image description of the coarse image content, making the description particularly suitable for tasks such as image retrieval [9,10] and scene classification [3,4,5]. Hidden topic model based image representations outperform in both tasks previous approaches [9,4].

When applying topic models in the image domain, the first step is to find an appropriate visual equivalent for words in documents. This is usually done by quantizing local images descriptors computed for each image. A wide variety of types of local descriptors has been proposed [11,12,13,14] and they have become very popular in many computer vision and pattern recognition tasks.

Although a thorough comparison of local descriptors in the context of matching and recognizing the same object or scene is presented elsewhere [15], an

G. Rigoll (Ed.): DAGM 2008, LNCS 5096, pp. 446–455, 2008.

evaluation between advanced local descriptors in the context of pLSA models is still missing. In a matching task, the aim is to find precisely corresponding points of an object or scene in two images under different viewing conditions such as lightning or pose changes. This requires a very distinct region description. However, in a pLSA based scene classification or image retrieval task we would like to pool features describing visually similar regions in order to produce meaningful visual words. Previous works on pLSA based image models only applied and compared the popular SIFT [11] descriptor or simple color/gray scale patches [3,4,5]. Bosch et al.'s work [4] proposes a variation of SIFT, taking color channels into account, in the context of scene recognition with a pLSA model based image representation.

In this work we compare two recently proposed local features descriptors, the geometric blur descriptor [13] and the self-similarity descriptor [14] in a scene classification task using a pLSA-based image representation. Both features have shown promising performance in image analysis tasks and have not been considered in the previous comparison [15]. As the SIFT based descriptors have shown to outperform other features [15], we take results obtained with the SIFT descriptor as a baseline and we use the classification rate on a previously unseen test set as a performance measure. Moreover we also evaluate three different local interest region detectors with respect to their suitability for this task.

2 Approach

In this work we use a pLSA model to represent each image [4]. pLSA [1] was originally derived in the context of text modeling, where words are the elementary parts of documents. The starting point for building a pLSA model is to first represent the entire corpus of documents by a term-document co-occurrence table of size $M \times N$. M indicates the number of documents in the corpus and N the number of different words occurring across the corpus. Each matrix entry stores the number of times a specific word (column index) is observed in a given document (row index). Such a representation ignores the order of words/terms in a document and is commonly called a *bag-of-words* model.

In order to be able to apply this model in the image domain, we first need to define a visual equivalent to words in documents. Visual words are often derived by vector quantizing automatically extracted local region descriptors. This work uses k-means clustering on a subset of local features extracted from training images and the cluster centers become our visual vocabulary.

Given the vocabulary, we extract local features from each image in the database and replace each detected feature vector with its most similar visual word, defined as the closest word in the high-dimensional feature space. The word occurrences are counted, resulting in a term-frequency vector for each image document. These term-frequency vectors for each image then constitute the co-occurrence matrix. Since the order of terms in a document is ignored, any geometric relationship between the occurrences of different visual words in images is disregarded.

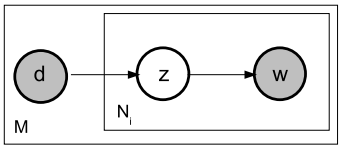

Fig. 1. Graphical representation of pLSA model: $M = \#$ of images in database, $N_i = \#$ of visual words in image d_i, observable random variable (shaded) w for the occurrence of a visual word and d for the respective image document, $z =$ hidden topic variable

Given the co-occurrence matrix, the pLSA uses a finite number of hidden topics to model the co-occurrence of visual words inside and across images. Each image is explained as a mixture of hidden topics and these hidden topics refer to objects or object parts. Thus we model an image as consisting of one or multiple objects: e.g., an image of a beach scene consists of water, sand and people. Assuming that every word w_j occurring in a document d_i in the corpus is associated with a hidden, unobservable topic variable z_k, we describe the probability of seeing word w_j in document d_i by the following model:

$$P(w_j, d_i) = P(d_i) \sum_k P(w_j|z_k)P(z_k|d_i) \tag{1}$$

where $P(d_i)$ is the prior probability of picking document d_i and $P(z_k|d_i)$ the probability of selecting a hidden topic depending on the current document, also referred to as the topic vector. Figure 1 shows the graphical representation of the pLSA model.

We learn the probability distributions of visual words given a hidden topic, as well as the probability distributions of hidden topics given a document, completely unsupervised using the Expectation Maximization (EM) algorithm [1,16]. Probability distributions of new images that are not contained in the original training corpus are estimated by a fold-in technique [1]. Here the EM algorithm is applied to the unseen images to compute its topic distribution while keeping the word distributions conditioned on the topic $P(w_j|z_k)$ fixed. In our work, we compute the parameters of a pLSA model on the training data and then apply this model to the test data using the fold-in technique. Finally, we represent each image by its associated topic vector $P(z|d)$ which gives us a very low-dimensional image representation.

For scene recognition the topic vectors of each unlabeled test image are classified by simple k-Nearest Neighbor (kNN) search through the labeled training images using the L2-norm as distance metric. We could apply more sophisticated distance metrics and/or machine learning algorithms such as SVMs to improve the classification results. As our main goal in this work is to compare different local feature descriptors and not machine learning algorithms, we have chosen the simple kNN approach.

3 Local Feature Descriptors

The image recognition system we describe in this paper starts building the pLSA based image representation by describing each image with a number of feature vectors of one kind. Then a vocabulary for that feature is computed as described in the previous section and a bag-of words representation is derived for each image. Local image features are often used in this context as they have the advantage of being more flexible than global image characterizations, while at the same time capturing more meaningful patterns than individual pixel values. Given a predefined interest point at a specified scale (i.e., size of local neighborhood), they describe the local image region surrounding the interest point compactly by a feature vector. There exists a large number of different types of local features, e.g. [11,12,13,14], each capturing a different property of a local image region and being more or less invariant to illumination, changes in viewpoint and other image transformations. In the following we will use the term *feature* and *descriptor* interchangeably.

We investigate the performance of the following three local feature descriptors in the context of the pLSA model:

SIFT [11]**:** A SIFT feature for a detected interest point is computed by first calculating the orientation of the most dominant gradient. Then, relative to this orientation the gradient-based feature vector entries are computed from the local gray-scale neighborhood. This is done by dividing the local neighborhood into subregions and subsequently accumulating the gradient magnitudes of each pixel into a local orientation histograms. The gradients are then weighted with a Gaussian window centered at the interest point location. The entries of the local orientation histograms form the entries of the, in our case, 128-dimensional feature vector. The vector is normalized to ensure invariance to illumination conditions. SIFT features are also invariant to small geometric distortions and translations due to location quantization. They are widely used in several computer vision and pattern recognition tasks. Thus the results obtained with SIFT features serve us as a baseline here.

Geometric blur [13]**:** The geometric blur feature vector computation is based on oriented edge channels, which in our work are computed by the boundary edge detector proposed by Martin et al. [17]. A sub-descriptor is determined for each edge channel; the concatenation of all sub-descriptors forms the final geometric blur descriptor. In order to compute a sub-descriptor we collect the values of sample points in the neighborhood of the interest point. Sample points lie on concentric circles around the interest point. The outmost circle in this work has a radius of 20 pixels. The distance between the 6 concentric circles decreases in a quadratic manner. As twelve equally distributed sample values are taken from each circle the size of each sub-descriptor is 72 and thus the dimensionality of the entire feature vector is 288 when using four oriented edge channels. The value of each sample point is taken from a blurred version of the respective edge channel; blurring is performed using a Gaussian kernel whose standard deviation is defined by the distance of the sample point from the interest point.

Self similarity [14]**:** To derive the self-similarity feature for an interest point, first a so called correlation surface is computed for the surrounding neighborhood. We compare a small image patch of size $x_1 \times x_1$ around the interest point with the larger surrounding image region of size $x_2 \times x_2$. In this work we choose $x_1 = 5$ and $x_2 = 41$. Comparison is based on the sum of square differences between the gray values. The distance surface itself is then normalized and transformed into a correlation surface, which in turn is transformed into a log-polar coordinate system and partitioned into 80 bins (20 angles, 4 radial intervals). The maximum values in each bin constitute the local self-similarity descriptor. Normalizing the descriptor vector ensures some invariance to color and illumination changes. Invariance against small local affine and non-rigid deformations is achieved by the log-polar representation; by choosing the maximal correlation value in each bin, the descriptor becomes insensitive to small translations.

All investigated feature descriptors are purely based on gray-scale images. The performance of scene classification, as considered in this work, is likely to improve by taking color into account (e.g. color SIFT [4]). As this may not be true for other content analysis tasks using probabilistic topic models such as object recognition or image retrieval (because here categories might be defined by shape rather than color), we do not consider color in this work.

We compute local features as described above at predefined interest points with an associated scale factor defining the size of the supporting image region around the interest point. In order to be able to compare the different local descriptors, we will also analyze the behavior of the most common feature, the SIFT feature, for three different interest point detectors. We will pick the best performing detector for feature evaluation. The considered detectors are:

- *Difference of Gaussian (DoG) detector* [11]: Here a DoG pyramid is computed. Interest points are defined as scale space extrema in the DoG pyramid and are associated with its respective scale. Thus the DoG detector facilitates scale invariant computation of the subsequent local feature descriptor if the supporting region size takes the scale factor into account. Note that in this approach the number of interest points per image varies as it depends on the structure and texture in each image.
- *Dense grid over several scales*: We compute interest points on a dense grid with spacing d between grid points in x- and y-directions and over several scales. As all images in our experiments are of the same size, the same number of interest points is computed for each image.
- *Edge sampling* [13]: In this approach we require interest points to be located at positions of high edge energy. First we compute oriented edge channels by using a boundary detector [17]. Then all edge channels are thresholded keeping only locations of high edge energy. Interest points are computed by randomly sampling those locations. For random sampling all edge channels are considered, nevertheless every position is selected at most once. Note that in this approach we predefine the number of features per image; features are computed at one scale only.

Table 1. Categories and number of images per category in the OT dataset

category	1	2	3	4	5	6	7	8
scene type	coast	forest	highway	inside city	mountain	open country	street	tall building
nb. images	360	328	260	308	374	410	292	356

Fig. 2. Sample images for each category in the OT dataset

4 Experimental Evaluation

Experimental Setup: We use the OT dataset [18] to evaluate the three different interest region detectors and descriptors in the context of a scene classification task. The database consists of a total of 2688 images from 8 different scene categories. The number of images as well as examples for each category are shown in Table 1 and Fig. 2, respectively. On this dataset we perform image classification by assigning each test image automatically to one of the eight categories.

We divide the images randomly into 1344 training and 1344 test images. We further subdivide the 1344 training images into a training and a validation set, of size 1238 and 106 respectively. We used the validation set to find the best parameter configuration for the pLSA model. In the model we fix the number of topics to 25 and optimize only the number of distinct visual words for the different detectors/descriptors. A number of 25 topics has been shown to give a good performance for this dataset [4].

Having determined the optimal number of visual words for the current detector/descriptor combination we re-train the pLSA model with the entire training set by merging training and validation set. Final results are then computed on the test set and detector/descriptor performances are compared.

In our experiments, we will first analyze the suitability of three feature detectors in the scene classification task while holding the feature descriptor fixed. Then we pick the best performing detector to evaluate the local descriptors.

Interest Point Detectors: We select the frequently used SIFT descriptor for the comparison of the three detectors. Their parameters are set as follows: the spacing d between grid points is 5 pixels, resulting in about 5250 features per image when using a factor of $2^{\frac{1}{4}}$ between different scales. The number of randomly sampled edge locations per image is set to 5000. In average a number of 559 features is extracted per image with the DoG detector.

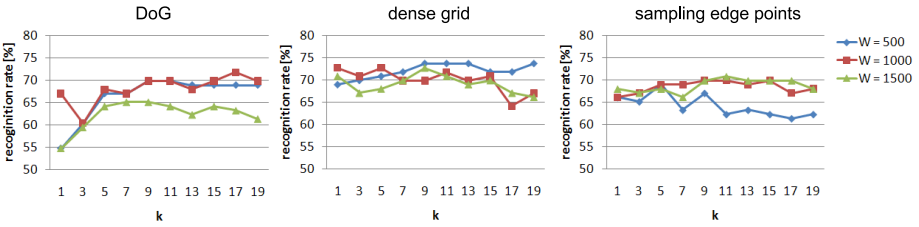

Fig. 3. Recognition rates on the validation set for the three different detectors over parameter k of kNN for different numbers of visual words

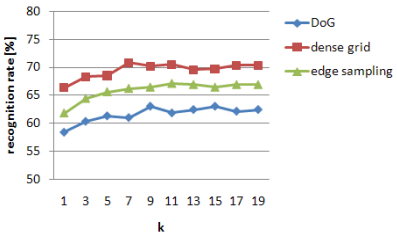

Fig. 4. Recognition rates on the test set for three different detectors over k for kNN

Figure 3 displays the resulting recognition rates on the validation set for different numbers of visual words W for all three detectors over the parameter k of the kNN algorithm. We observe that for the DoG detector, the dense grid detector over several scales, and the edge sampling detector $W = 1000$, $W = 500$ and $W = 1000$ gives the best recognition results, respectively.

Using these parameter settings we train a pLSA model on the entire training set for each detector type and fit the test set images to this model in order to compute a topic vector representation for all images. The comparison of the recognition results on the test set can be seen in Fig. 4. The dense grid detector outperforms the other detectors followed by random edge sampling.

This may be due to several reasons: Firstly, both the dense grid detector and the random edge sampling algorithm compute more features per image than the DoG detector and also, they compute an equal number of features for each image. This may enable a better fitting of the pLSA model to the scene recognition problem. Secondly, the interest points and regions computed by the dense grid cover the entire image and thus the bag-of-words image representation also covers the entire image and not only regions close to edge pixels or scale-space extrema. A further reason might be that in a scene recognition task the repeatability of exact positions and scales, as provided by the DoG detector, may be not as important as in other tasks such as object recognition where one would like to match only the exact subpart. The DoG detector offers in contrast to the other detectors scale invariance. Nevertheless this is also not as important in scene classification as, e.g., in object detection. All images of one category are

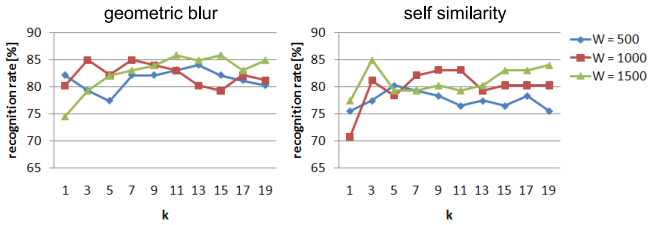

Fig. 5. Recognition rates on the validation set for the two descriptors for different numbers of visual words and k in kNN

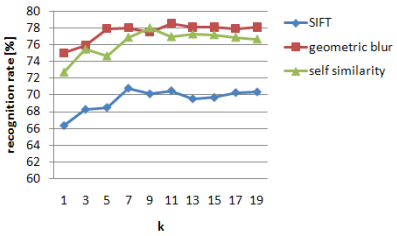

Fig. 6. Recognition rates on the test set over k for kNN for different local feature types

taken at approximately the same scale. Note that the results are consistent with previous results [4], where a dense representation performed best, too.

Feature Descriptors: The dense grid detector showed the best recognition performance in the evaluation above, thus we use this interest point detector in the subsequent comparison of local feature descriptors. First we determine the appropriate number of visual words in the pLSA model for each descriptor. This has already been done for the SIFT feature (see Fig. 3). Figure 5 depicts the recognition rates for different k in the kNN and different numbers of visual words, for the geometric blur descriptor and the self-similarity descriptor. The best results for both features are obtained using 1500 visual words.

For both descriptors we train a novel pLSA model on the entire training set and compute a topic vector representation for all training and test images. Then we compare the results of all local features, including SIFT, in Fig. 6.

It can be seen that both, geometric blur and self-similarity features outperform the commonly used SIFT feature by more than 5%. Moreover the geometric blur feature has a slightly better recognition rate, about 1% better, than the self-similarity feature, and the best recognition is achieved for $k = 11$ with 78.05%. It should be noted in this context, that a performance difference of 1% is not statistically significant given the small OT dataset. Nevertheless, the self-similarity descriptor is of lower dimensionality compared to the geometric blur features: 80 vs. 288 dimensions. This lower dimensionality makes computations such as clustering and visual word assignment much faster. Moreover, the self-similarity feature is computed without performing segmentation or edge detection as has to be done to compute the oriented edge channels for the geometric

SIFT (W=500, k=7)

	1	2	3	4	5	6	7	8
1	66,11	1,67	13,33	0,00	5,56	9,44	2,78	1,11
2	0,61	92,07	0,61	0,00	3,66	3,05	0,00	0,00
3	5,38	0,00	76,15	2,31	1,54	5,38	6,92	2,31
4	0,65	0,00	1,95	81,82	0,00	0,65	10,39	4,55
5	8,02	4,28	2,14	0,00	66,31	14,44	4,28	0,53
6	13,66	10,24	8,78	0,00	12,20	50,73	3,41	0,98
7	0,00	0,68	1,37	8,22	4,11	0,00	82,88	2,74
8	1,12	0,56	6,18	12,92	1,69	1,69	15,17	60,67

geometric blur (W=1500, k=11)

	1	2	3	4	5	6	7	8
1	78,33	0,56	7,78	0,00	0,56	12,78	0,00	0,00
2	0,00	93,29	0,00	0,00	1,83	1,83	3,05	0,00
3	16,15	0,00	70,77	2,31	1,54	4,62	4,62	0,00
4	4,55	0,00	0,00	81,82	0,00	1,30	3,90	8,44
5	0,53	4,81	4,28	0,00	79,14	7,49	3,74	0,00
6	21,46	4,39	5,85	0,00	6,83	59,02	2,44	0,00
7	0,00	0,00	2,05	7,53	1,37	0,00	85,62	3,42
8	0,00	1,69	1,12	8,43	0,56	0,00	3,93	84,27

self similarity (W=1500, k=11)

	1	2	3	4	5	6	7	8
1	66,11	1,11	9,44	0,56	2,78	17,78	2,22	0,00
2	0,00	90,24	0,00	0,00	2,44	3,66	3,66	0,00
3	9,23	0,00	73,08	0,77	6,92	3,85	6,15	0,00
4	1,30	0,00	0,65	82,47	0,00	0,00	5,84	9,74
5	4,28	6,42	3,74	0,53	70,05	6,95	8,02	0,00
6	9,27	5,37	3,90	0,49	4,88	74,63	1,46	0,00
7	0,00	2,05	1,37	3,42	1,37	0,68	91,10	0,00
8	0,00	0,56	0,00	10,11	0,00	0,00	8,99	80,34

Fig. 7. Confusion tables for results on the test set for different descriptor types and a dense grid region detector. The numbers 1,2,...8 refer to the categories listed in Table 1.

blur feature. Thus, given the similar performance and the more than a magnitude lower computational complexity over geometric blur, the self-similarity feature is the prefered feature[1].

For a more detailed analysis of the results, the confusion tables for the best performing parameter settings for each descriptor are depicted in Fig. 7. In the confusion tables it can be seen that there are some categories, such as *forest*, *inside city* and *street*, where all descriptors work almost equally well, showing a performance of over 80% and in the *forest* category achieving over 90% accuracy. We also noticed some confusions occur between closely related categories with similar visual appearance, e.g. *open country* and *coast*, *tall building* and *inside city* as well as *mountain* and *open country*. In this cases, results might be further improved by including color.

The largest differences can be noticed in the category *tall building* where SIFT has an about 20% smaller recognition rate than both other features. The geometric blur descriptor significantly outperforms SIFT and self similarity in the categories *coast* and *mountain*, whereas the self-similarity feature performs best in the *open country* category.

Finally we would like to examine the variance in performance due to random initialization in both, the k-means clustering algorithm and the pLSA implementation. Therefore we choose the parameter and feature setting of the best performing configuration so far (geometric blur descriptor, $W = 1500$, $k = 11$) and repeat the scene classification experiment on the test set ten times, each time computing the visual vocabulary and pLSA model with different random initializations. The recognition rates range between 77.75% and 79.69% with an average value of 78.93% and a standard deviation of 0.58%. It can be seen that there are no large variations between different runs of the same experiment.

In summary it can be stated that for scene classification geometric blur outperforms the other features. In cases where fast computation is needed one should also consider using the lower dimensional and faster-to-compute self-similarity feature which only performs slightly worse than the geometric blur feature.

[1] We have experimented with simpler oriented-edge channel computations than the one presented by [17], however performance dropped drastically indicating that sophisticated edge channel computations are important.

5 Conclusion

In this work we have studied the influence of the type of local feature descriptors in the context of pLSA based image models and a scene recognition task. We compare three different local feature descriptors. Our results show that the commonly used SIFT descriptor is outperformed by the two other feature descriptors: the geometric blur feature and the self-similarity features. Moreover we also evaluate three different local interest region detectors with respect to their suitability in this task and we found that a dense grid detector over several scales performs best. Future work could consist in adopting the best performing descriptors to color.

References

1. Hofmann, T.: Unsupervised learning by probabilistic Latent Semantic Analysis. Mach. Learn. 42(1-2), 177–196 (2001)
2. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. J. Mach. Learn. Res. 3, 993–1022 (2003)
3. Fei-Fei, L., Perona, P.: A Bayesian hierarchical model for learning natural scene categories. In: CVPR, pp. 524–531 (2005)
4. Bosch, A., Zisserman, A., Munoz, X.: Scene classification via pLSA. In: ECCV (2006)
5. Quelhas, P., Monay, F., Odobez, J.-M., Gatica-Perez, D., Tuytelaars, T., Gool, L.V.: Modeling scenes with local descriptors and latent aspects. In: ICCV, pp. 883–890 (2005)
6. Sivic, J., Russell, B.C., Efros, A.A., Zisserman, A., Freeman, W.T.: Discovering objects and their location in images. In: ICCV (2005)
7. Cao, L., Fei-Fei, L.: Spatially coherent latent topic models for concurrent object segmentation and classification. In: ICCV (2007)
8. Barnard, K., Duygulu, P., de Freitas, N., Forsyth, D., Blei, D., Jordan, M.: Matching words and pictures. J. Mach. Learn. Res. 3, 1107–1135 (2003)
9. Lienhart, R., Slaney, M.: pLSA on large scale image databases. In: ICASSP (2007)
10. Hörster, E., Lienhart, R., Slaney, M.: Image retrieval on large-scale image databases. ACM CIVR, 17-24 (2007)
11. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV 60(2), 91–110 (2004)
12. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape context. PAMI 2(4), 509–522 (2002)
13. Berg, C.A., Berg, T.L., Malik, J.: Shape matching and object recognition using low distortion correspondences. In: CVPR (2005)
14. Shechtman, E., Irani, M.: Matching local self-similarities across images and videos. In: CVPR (2007)
15. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. PAMI 27(10), 1615–1630
16. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. Journal of the Loyal Statistical Society B.39 (1977)
17. Martin, D., Fowlkes, C., Malik, J.: Learning to detect natural image boundaries using local brightness, color, and texture cues. PAMI 26(5), 530–549 (2004)
18. Oliva, A., Torralba, A.: Modeling the shape of the scene: a holistic representation of the spatial envelope. IJCV 42, 145–175