Learning Distance Functions for Automatic Annotation of Images

Josip Krapac, Frédéric Jurie

Learning Distance Functions for Automatic Annotation of Images

Josip Krapac and Frédéric Jurie

INRIA Rhône-Alpes, 655, Avenue de l'Europe, 38334 Saint Ismier Cedex, France
{josip.krapac,frederic.jurie}@inrialpes.fr

Abstract. This paper gives an overview of recent approaches towards image representation and image similarity computation for content-based image retrieval and automatic image annotation (*category tagging*). Additionaly, a new similarity function between an image and an object class is proposed. This similarity function combines various aspects of object class appearance through use of representative images of the class. Similarity to a representative image is determined by weighting local image similarities, where weights are learned from training image pairs, labeled "same" and "different", using linear SVM. The proposed approach is validated on a challenging dataset where it performed favorably.

1 Introduction

In the last decade we have witnessed a rapid increase in the number of digital images. However, the access to this content is hindered by the availability of methods to search and organize it. Available systems that perform these tasks rely on textual information that describes the image's semantics and use text search algorithms to search image collections. Major drawback of this approach is that it limits search only to images for which textual labels are available and since labeling of images is usually performed by humans, it is a slow and inherently subjective process.

Methods that search and organize images by their visual content are known as *content-based image retrieval* (CBIR) methods. This way the problem of absence, incorrectness or incompleteness of textual labels is circumvented.

Applications of CBIR [7] include search of image databases by content, automatic annotation of image collections as well as user localization in real environments and building virtual worlds from real images.

Questions that need to be answered in solving these tasks are: (1) how to represent the images to allow search and organization by their content, (2) how to define similarity function between the image representations so that it reflects human perceptual similarity.

The main contribution of this paper is a new distance function between an image and a category. This distance function is used for propagation of category tags to unannotated images. The distance function to a category is obtained by combining the distance functions to typical representatives of the categories – *focal*

N. Boujemaa, M. Detyniecki, and A. Nürnberger (Eds.): AMR 2007, LNCS 4918, pp. 1–16, 2008.

images. The similarity function between focal image and unannotated image combines local similarities from detected, represented and matched image *patches* to obtain global image similarity. The combination of local (patch) similarities which defines global similarity is learned from a few training image pairs labelled as "same" or "different" which labelling presents equvivalence constraints.

The paper is organized as follows: we first give in the next section an overview of recent advances in this very active field, and then describe in Section 3 the proposed approach. Finally, Section 4 gives a few experimental results obtained on a very challenging problem.

2 Related Work

2.1 Global vs. Local Approaches

The first question to ask is what features to extract that will help perform meaningful retrieval. Ideally, features should be related to image semantics, but that is very difficult to obtain because of the semantic gap. However, noticing that there is a statistical dependency between real world raw images' pixels and semantics, it is possible to show that only small amount of information contained in the raw image data is necessary for the image retrieval and classification. Therefore the first step is to choose a suitable low dimensional representation for the image in which information important for discrimination is retained.

First developed methods used global image histograms of color and shape [9] or texture [17] features. The distance between representations is generally computed using L_2 distance, or in case of normalized histograms using standard distance measures between probability distributions: χ^2 distance or KL divergence. While global representations have advantage to be easy to build and to be invariant to position of objects in the image, they provide only a very rough representation of images allowing to deal only with global contexts (like "forest" images) or with objects covering the whole image.

To overcome limitations of global approaches *relevance feedback* has been introduced. The idea is to take the results provided by an algorithm and to use information about whether or not those results are relevant before performing a new query. The feedback, given by the user, allows to weight the image representation before re-computing the distance.

Recently developed image representations overcome problems of global approaches the other way: by representing the image as collection of image parts – patches. These representations are thus called *local representation*.

Most popular way to select patches is by means of interest point detectors [28]. Interest point detector detect characteristic structures (corners [8], blobs [15] or ridges [30]) in the image. One of the main reasons for their popularity is the ability to adapt to scale variation of the characteristic structures, detecting characteristic structure at their *intrinsic scale*. In this case patches are defined as local neighborhoods of detected interest points at detected intrinsic scale of characteristic structure.

Patches can also be sampled randomly [24] or even densely [32] from the image.

Selected patches are described by signatures or *patch descriptor*. Important condition on design of patch descriptor is their robustness to various photomotric and geometric transformations in the sense that patches that belong to *same* object parts should have *same* descriptions. To achieve invariance to these changes, patch descriptors either model the distribution or characterize the properties of distribution of patch's color [31], grayscale values [25,12] or response of filter banks which describe local texture (SIFT [16], textons [14], geometric blur [2], PCA-SIFT [13]). In the case when patches are obtained by segmentation they can be described by the shape of segment boundary [1].

2.2 Query-by-Example Image Search

Local image representation maps image into a set of patches described by visual descriptors. This image representation scheme can be used for building image search algorithms because images can be compared by comparing two sets of visual descriptors. Schmid and Mohr [28] propose to match the visual descriptors extracted from the query image with those extracted from the images in the image database and to vote for the image from database that accumulates the largest number of matched keypoints.

This very efficient yet powerful approach have been used by many authors for different tasks such as the localization of the camera [5,10], the reconstruction of 3D scenes by assembling images searched over the Internet [5,10] or the navigation of autonomous robots [22].

2.3 Category Tagging

These approaches allow to search images by presenting an example image as query. To broaden the application of image search methods beyond these *query-by-example* approaches, a small set of images is semantically annotated by humans with category tags that semanticaly describe image's content, so that images can be searched both by content *and* semantics. The problem is then how to propagate category tags from annotated to unannotated images.

This enriching images with tags related to their content, termed here as *category tagging*, can be casted as a binary classification problem: if the image belongs to the category – the corresponding tag is added. Generally, *recognition algorithms* which assign an object identifier to an image possibly containing the object, are used for tagging images.

One strong limitation of these early patch based approaches for category tagging is their restriction to the recognition of images that are strictly identical to the query. This is because the comparison is based on the matching of local structures, which are assumed to be strictly identical.

We are here more interested in methods working at the *category level* instead of the *instance level*. The difference between *object instance recognition* and *object class recognition* is in the definition of the visual class denoted by object identifier. In the former case all images that contain the exactly same object

instance (eg. same car) are tagged as belonging to same visual category, so images containing different object instances belong to different visual categories (eg. different car models). In later case all images that contain the same object class are tagged as belonging to the same visual category (eg. cars are one visual category, cats are other visual category...).

In object instance recognition only variability between the images of same object is due to changes in object pose (changes of observer's viewpoint with respect to the object), object configuration (changes between parts of the object with respect to each other) and scene illumination. Object surface can locally be approximated by a plane so local pose changes can be approximated as rigid, and hence majority of research efforts are concentrated on capturing the variation due to object pose [19] in local descriptions. In object class recognition there is additional variability due to intra-class (or within-class) differences between objects of the same visual class. This variability is complex and hard to characterize as it stems from assigment of semantic category tags, and thus makes the problem of object class recognition significantly harder then problem of object instance recognition.

2.4 Visual Vocabularies

Because of large intra class variation of object appearances and because of the small number of training data generally available, building geometric models of classes is difficult, not to say impossible. Similar problem occurs in texture classification, which can be considered as a source of inspiration for object class recognition problems. Interestingly, Malik et al. [14], in their work on texture classification, suggest to vector quantize local descriptors of images (called *textons*) and then to compute texton statistics.

This idea of building a distance between images based on the comparison of texton statistics have then been widely used for image classification. Indeed, instead of comparing the two sets directly (each patch with each patch) an indirection step in introduced – description of patch by *visual word*. Visual words are quantized patch description vectors, where quantization bins can be obtained by clustering (K-means [29], hierarchical k-means [18], mean shift [11]) of training description vectors. Visual word is representative of the bin (cluster), and set of all visual words is called *visual vocabulary*. Analogously to descriptions of text documents, here an image is described by occurrences of visual words. This representation relies on local descriptions and is invariant geometrical structure (layout) of detected patches and hence is termed *bag-of-(visual) words representation*. Each image is described by histogram of occurrences of visual words that are present in the image, so that each object class is characterized by a distribution of visual words.

Visual vocabulary influences the classification performance, so creation of visual vocabulary is an important step. Majority of clustering techniques used for creation of visual vocabulary are computationally expensive, and more importantly, don't use the information about patch class so created vocabularies are not discriminative.

In [20] extremely randomized forests were used for vocabulary creation. Creation of random forest is computationally much less complex then other methods of vocabulary creation, while attaining comparable results. Also, patch class labels are used to guide the cluster splitting towards creation of discriminative clusters.

The very recent Pascal VOC2007 image categorization challenge[1] has demonstrated the superiority of these vocabulary based approaches over all other competitive ones.

2.5 Similarity Functions and Focal Images

In [23] similar approach was used to learn the similarity function between images from train image pairs labeled "same" or "different". Image pair is represented by the set of patch pairs, which are formed in process of *patch matching*. Instead of quantization of patch descriptions, *pairs* of patch descriptions were quantized discriminatively using extremely randomized forests. Each visual word in this case represents the characteristic local (patch) (dis)similarity. The patch pairs are weighted towards "same" or "different" prediction by linear classifier, so global similarity is obtained by weighting and integrating over local (patch) similarities. We use this algorithm to build the visual vocabularies and measure image distance to focal images.

Regarding the class models, our work is inspired by the recent work of Frome *et al.* In [6] classification of query image is performed by combining the similarities to representative images of the object class – *focal images*, and choosing the class for the query image with biggest combined similarity. Similarity is learned from training image pairs, for each focal image separately. This is an example of divide-and-conquer approach, since calculating the similarity to an image is easier task then calculating the similarity to a class, because class has more complex feature distribution then an image.

Our work is inspired by both approaches: we use focal images to divide problem into simpler problems, but we learn similarity measure as in [23].

3 Method

As mentioned before, we cast the *category tagging* problem as the problem of object class recognition, i.e. we determine the classes a *query image* belongs to.

In general, the decision of object class membership is brought by calculating a measure of similarity to each learned object class for given query image and comparing that similarity measure with a threshold. If the similarity is high enough, the tag is added.

Here, inspired by [6], we experiment with a different approach. We calculate similarity between query image and an object class by combining similarity measures calculated between query image and representative images of an object class. Representative images of an object class are called *focal images*.

[1] http://www.pascal-network.org/challenges/VOC/voc2007/

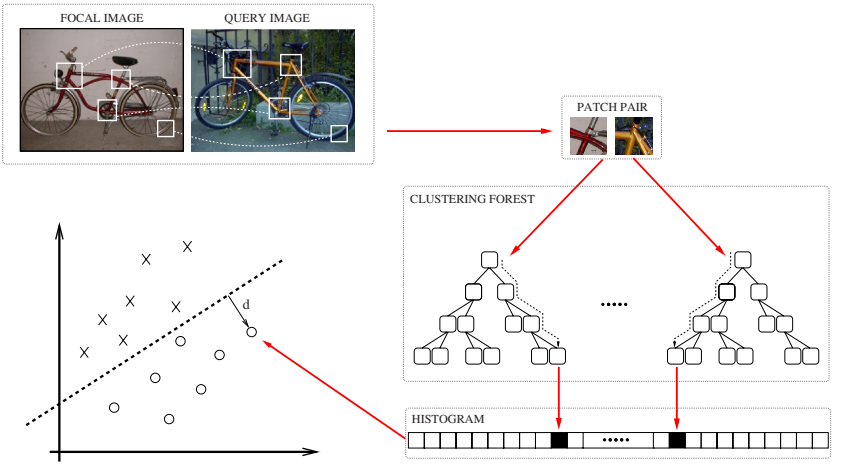

Fig. 1. Similarity between focal image and query image: 1) Matching patch pairs 2) Clustering patch pairs by clustering forest 3) Representation of image pair by histogram of cluster occurrences 4) Weighting clusters to calculate similarity d between focal image and query image

The overview of similarity the measure calculation for an image is outlined in the Fig. 1, while in Fig. 2 we show how we bring the final decision by combining the results of similarities to focal images.

3.1 Similarity to the Focal Image

For each focal image we construct a classifier that gives the measure of similarity between the selected focal image and the query image. Training image pairs (focal,query) are formed and labelled "same" and "different" by choosing the query images of a known class from training set. The calculation of similarity measure between the focal image and the query image, as illustrated Fig. 1, can be divided in 4 steps.

Step 1: Selection, Description and Matching of Patches. We represent the image as a collection of patches and use an interest point detector to select the patches.

Using interest point detector ensures that local image structures detected in one image will be also detected in other images, an important requirement for creating patch pairs by matching of selected patches. Although one could question if the patches selected this way are discriminative enough to be used in challenging task of object class recognition, we believe that by selecting a large number of patches we will be able to form patch pairs which are discriminative for an object class.

Motivated by results of a recent comparative study [18] of interest point detectors and descriptors for object class recognition we use scale-invariant Hessian

point detector, which enables the patch description to be invariant to structure's scale, and consequently makes our methods invariant to the scale of the object.

We choose SIFT [16] descriptor to describe patches, because it has been shown to be sufficiently robust to capture the intra-class variance of patches [18]. Patch pairs are formed by selecting similar patches in two images, where similarity between patches relates inversely to L_2 distance between patches' SIFT descriptors.

Step 2: Clustering Patch Pairs. Once we have formed patch pairs that describe local similarities between focal and query image we can combine these local similarities to form global similarity. Instead of using only the distance between visual descriptors to describe local similarities (as in [6]) we introduce clustering step to additionally characterize the similarity between patches. We choose to cluster patch pairs using EXTRA trees, motivated by their good performance in the similar task of learning of the distance between images [23]. Each cluster describes characteristic patch (dis)similarity.

EXTRA trees are binary trees constructed by recursively splitting the nodes on the basis of labelled data in the node. The process starts from the root node, which contains all training patch pairs from focal image to query images. At each node, we randomly generate a number of splitting criteria and select the optimal splitting criterion for the node.

Splitting criterion $c(\cdot,\cdot|i,s,t)$ is binary predicate that as input has a pair of patches, represented as pair of vectors $\mathbf{f}^{\text{focal}}$ and $\mathbf{f}^{\text{query}}$ and is determined by parameters i,s and t. Parameter i selects the dimension of vector which represents the patches in the pair, t is threshold and $s \in \{-1, 1\}$ determines sign of comparison:

$$c(\mathbf{f}^{\text{focal}}, \mathbf{f}^{\text{query}}|i,s,t) = (s \cdot (\mathbf{f}_i^{\text{focal}} - t) \geq 0) \wedge (s \cdot (\mathbf{f}_i^{\text{query}} - t) \geq 0) \qquad (1)$$

We can interpret the above equation as: "If i^{th} dimension of *both* focal image patch vector and query image patch vector are above (or below, depending on s) threshold t then splitting criterion is true." Random splitting criterion is created by selecting parameters i, s and t at random.

When splitting criterion is applied to all patch pairs in the current node of the tree it splits the patch pairs into two disjunctive sets: a set of patch pairs for which the splitting criterion is true, and a set of patch pairs for which the splitting criterion is false. These sets correspond to two child nodes. The optimality of splitting criterion is evaluated by its information gain IG which is calculated from the two sets obtained by applying the splitting criterion: $IG = H - \frac{n_0 H_0 + n_1 H_1}{n}$, where H,H_0 and H_1 denote, respectively, the entropy of patch pairs in current (parent) node, entropy of patch pairs for which splitting criterion is false (entropy of first child) and entropy of patch pairs for which splitting criterion is true (entropy of second child node). Similarly n,n_0 and n_1 denote, respectively, numbers of patch pairs in the parent node, first and second child. The criterion with highest information gain is selected for the node.

By selecting this optimality measure of splitting criteria we supervise the clustering process to create discriminative clusters. Each leaf of the tree is a

cluster which defines similarity between patches of the match: patch similarity is determined by the path the patch pair took from root to leaf node.

The main advantage of random trees is computational simplicity of construction because, in contrast to methods that search the optimal values of splitting criterion parameters, we just select them at random and keep the one with highest information gain. On the other hand, randomness responsible for computational simplicity of construction increases variance of resulting clusterer.

To reduce the variance introduced by random sampling of parameters of splitting criteria we use a number of independently grown trees – a clustering forest [23]. Since every tree defines a different clustering of patch pair feature space, every pair of patches reaches several clusters (leaves of the trees in the forest) – one cluster per tree in the forest.

The number of patch pairs to create the clusterer is question of bias/variance trade-off. If we construct a tree from too small number of patch pairs the variance will be large and if we use all available matches we will over-fit the training data so clusters will be specific to training data and will not generalize to test data. On the other hand,as always in vision, the available data is scarce and we want to use information from all available patch pairs. Intermediate solution is to create tree using all available matches, but stop the splitting of the node if number of patch pairs remaining in the node is too small, to prevent over-fitting. This procedure is known as *pruning*.

Step 3: Representing a Pair of Images by Histogram. Global similarity between focal image and query image is represented by a set of patch pairs. Using the clusterer we represent global similarity between focal and query image by a histogram of occurrences of clusters that describe local (patch) (dis)similarities.

Having constructed clusterer we can represent every pair of images (focal, query) by a histogram of occurrences of clusters – leaves in the forest which are reached by patch pairs from this pair of images.

The histogram of pair of images is interpreted as a vector in a high dimensional space where each dimension of a vector (each histogram bin) corresponds to a cluster (leaf of the forest).

We choose to binarize histograms, as in [20], to avoid the problem of non-binary normalized histograms where the influence of small (relative) number of discriminative patch pairs can be reduced by a large (relative) number of non-discriminative ones.

Step 4: Weighing the Clusters. We believe that clusters are specific to an object class and that some clusters are more significant for the similarities of objects within the class (and dissimilarities of objects between the classes) then the others. This relative relevance of clusters for classification is acknowledged by assigning weights to clusters.

Since for every query image in the training set we know the true class we use this information to learn the linear classifier to find a hyper-plane that separates pairs of images of the same class as focal image from pairs of images which have different class from focal image. The normal to the obtained hyper-plane is a

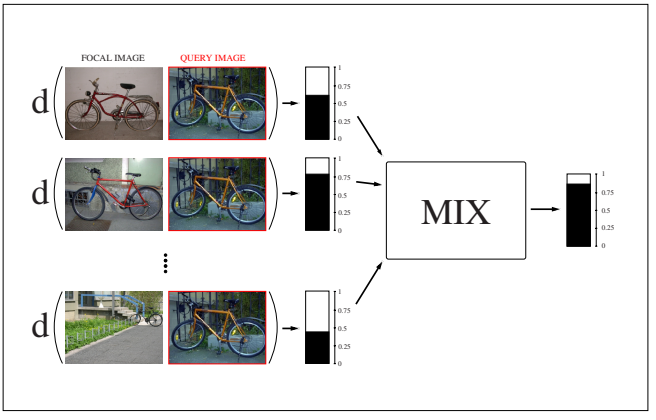

Fig. 2. Combining distances from focal images to query image by mixture of experts

vector of weights that reflect relative importance of clusters (local similarities) for classification (global similarity).

We use linear support vector machine as a classifier because for a large number of applications it has shown good generalization performance when using high dimensional data and small training set, as it is our case.

3.2 Similarity to an Object Category

Focal images represent the different aspects of object class appearance. To obtain similarity of a query image to an object category we need to combine all learned similarities to focal images. We call the classifiers that give similarity measures to the focal images *base classifiers*, and their scores *similarities*.

In general, to combine base classifiers their decisions must be comparable. Standard way is to convert similarities to probabilities, so for each query image classifier reports probability that query image belongs to the same categories as focal image.

To this end we use Platt's scaling [27], a procedure that finds the parameters of the sigmoid function that performs mapping of base classifier's scores (similarities) to probabilities by maximum likelihood fit to the data. It has been shown that this procedure outperforms others in case of small learning sets [21]. To find these parameters we have used the validation set. In the rest of the paper we will assume that similarities are mapped to probabilities and will use terms *similarities* and *probability estimates* interchangeably.

To combine the probability estimates of base classifiers, in absence of any additional information about the data that would guide the combination process, fixed rules are usually used. The most common fixed rules are: sum, product, median, maximum and minimum of all scores of base classifiers [4].

Most commonly used fixed rule is the "sum of scores" rule, also used by [6] to combine the similarities from focal images. This rule assumes that all classifiers have the same influence on the combined decision.

Each of these fixed rules will perform well if some assumptions are satisfied [4], but since these assumptions are never fully satisfied we can always find a better way to combine classifiers if we use specific information about the data we want to combine.

This is why we train linear SVM to learn the classifier relative weights on the validation data. Since category tags are available for the images in the validation set we can use probability estimates of base classifiers on validation data as training data for *mixture-of-experts* classifier that learns classifier weights. Learned weight vector can have negative components, which means that we reverse decisions of some classifiers. To avoid this anomaly, we map negative components of weight vector $\mathbf{w}^c$ by taking their absolute value and dividing it by L_1 norm of the weight vector:

$$w_i^{c'} = \frac{|w_i^c|}{\sum_{i=1}^{N_c} |w_i^c|}$$

The decision is then brought by *weighted sum* rule, where weights are learned by combining linear classifier and mapped to be positive.

4 Experimental Results

4.1 Dataset

We have tested the proposed method on Graz-02 database [26]. This database has been designed for object recognition or object categorization, containing images with objects of high complexity, high intra-class variability and highly cluttered backgrounds.

We used only two classes: bicycles and background. Bicycles are selected to have same orientation which leaves 102 images of bicycles and 300 images of background.

The data-set is divided in 4 subsets: set of focal images, training set, validation set and testing set. Half of randomly selected images are selected for testing, and another half is divided between testing and validation set. Images in the training set are used to learn the clusterer (EXTRA trees) and classifier (SVM), and images in the validation set are used to find the parameters of sigmoid function that performs mapping of similarities to probabilities and to find the mixture-of-experts classifier weights.

We have selected 8 focal images from the class of bicycles: in first four focal images different types of bicycles (corresponding to different aspects of bicycle class distribution) are prominent object in the image; in the remaining focal images bicycles are not prominent due to various degradations (small scale of the object, occlusions, overlapping and low contrast).

4.2 Experimental Setup

We use scale-invariant Hessian point detector to select the patches from imagesm and set low detection threshold to get enough patches for construction of patch

pairs. The scaling factor between neighboring scales of image pyramid, used in the process of scale selection, is set to 1.2. We discard the patches detected at small scales.

For patch description we use SIFT descriptor. SIFT descriptors are calculated at scale detected by scale-invariant Hessian interest point detector. For SIFT descriptor we use 4 bins for spatial distribution of gradients (both in x and y direction) and 8 bins to describe gradient orientation distribution, giving a 128-dimensional patch descriptor. Descriptors are normalized to have L_2 norm equal to 1 so they are invariant to affine illumination changes.

We use 5 trees in the forest, motivated by results of [23]. The dimension of histogram that represents the pair of images is approximately 20000. Average depth of the trees is 39 ± 4.

For weighting the clusters we use C-SVM, with parameter $C = 1$.

We have used Platt's scaling implementation from LIBSVM library [3] to perform mapping of similarities to probabilities.

Performance is expressed as equal error rate (EER) of precision and recall. For each focal image, due to random nature of classifier construction, we construct 5 classifiers and report mean and standard deviation.

4.3 Matching Patch Pairs

We constrain matching procedure by forming matches for each patch in the focal with just the most similar patch in query image. We call this kind of matching *unidirectional matching*.

Bidirectional matching is the matching procedure which additionally constrains unidirectional matches to keep only the patch pairs which are most similar in other direction also (from query image to focal image).

Using the unidirectional matching the number of patch pairs is equal to number of patches in focal image, while with bidirectional matching limits the number of patch pairs to $\min(N_{\text{focal}}, N_{\text{query}})$, where N denotes the number of patches detected in the image. This is why in case of unidirectional matching we use binary histograms, while in case of bidirectional matching the use of binary histograms would introduce bias due to different number of patch pairs per query image. In latter case we therefore use non-binary histograms and normalize them (to have L_1 norm 1) to make the description vector of image pair invariant to number of formed patch pairs.

We conducted experiments to test the effectiveness of described matching strategies. The results are presented in Table 1.

Table 1. Influence of matching patch pairs strategy to classification results

Focal image ID	1	2	3	4	5	6	7	8	all
unidirectional matching	0.88	0.85	0.88	0.87	0.89	0.87	0.90	0.86	$\mathbf{0.87 \pm 0.02}$
bidirectional matching	0.82	0.83	0.82	0.82	0.83	0.75	0.80	0.82	$\mathbf{0.81 \pm 0.03}$

This results show that unidirectional matching performs better. This results can be explained by several reasons:

– Number of matches to describe the image is smaller in case of bidirectional matching, which means that clusters will be formed on the basis of smaller amount of data and hence may not be representative characterization of local (dis)similarities.
– Representation of image pair by non-binary histogram causes the suppression of small, but important discriminative clusters
– Bidirectional matching limits the patch matching to patches from region of feature space where feature distributions of both images overlap. If in this region most patches belong to background we will not be able to learn from patches that belong to object.

4.4 Creation of the Clustering Forest

We have performed the experiments to determine what is the best way to create the clusterer with given number of matches. We have created the clusterer using only 5% and 20% of available training patch pairs, selected randomly from all available training patch pairs. In this case we do not need to prune the trees. When we use all training patch pairs we prune the trees, when node contains less then 20 patch pairs, by turning the node into leaf. Table 2. summarizes the experiment's results.

Results of experiments show that our assumption was correct: the trees created from only 5% of patch pairs perform worse than the ones created with 20%. Although the results of experiments in last two columns of Table 2. are the same, we choose to use the trees created from all available patch pairs because we do not introduce randomness due to nature of patch pairs selection to create the clustering forest.

4.5 Influence of Context

To investigate the influence of background patches we have cropped images of the bicycles to contain only the bicycles (object). We have lowered the detection threshold of interest point detector to 0 to get approximately same number of patch detections in cropped image as in original, uncropped one.

The results in Table 3. show that for some focal images context (background) shows important role, especially in focal image 5 where bicycle is not prominent object in the image due to small scale. For focal image 7 results are also significantly worse when using cropped images. In this image bicycles are overlapping (stacked one behind another), and therefore, because of the overlap, parts of the bicycles locally do not look like bicycles. The good result of the method on these focal images in uncropped case can be explained by influence of patch pairs

Table 2. Influence of percentage of matches to create the clusterer

% matches to create clusterer	5%	20%	100%
Pruned	no	no	yes
EER of PR	0.80 ± 0.03	0.83 ± 0.03	0.83 ± 0.03

Table 3. Influence of background of matches to classification results

Focal image ID	1	2	3	4	5	6	7	8	all
object + background	0.88	0.84	0.88	0.87	0.88	0.86	0.90	0.86	0.87 ± 0.02
object	0.87	0.83	0.85	0.86	0.76	0.85	0.83	0.86	0.84 ± 0.03

formed from background patches in focal image. Nevertheless, for majority of focal images results are not significantly influenced by patch pairs formed from background patches, so we can conclude that method learns to discriminate the object dominantly from object part of image.

4.6 Combination of Classifiers

We have performed experiment with use of fixed rules to combine the probability estimates of base classifiers and compared with results of a linear support vector machine (weighted sum). The results are presented in Table 4.

4.7 Advantages and Limitations

The results are presented for one focal image, because the results for other focal images are similar, so advantages and limitations of the method can be empathized on one example. From results of classification for first focal image presented in the Fig. 3 we can conclude that similarity is correctly determined as high for images where object (bicycle) is dominant in the image. Since the bicycles are not compact objects there is large amount of background present even in the patches which contain parts of the bicycle. For bicycles which have simple, almost uniform background the performance is good. Majority of misclassified bicycles (detected as background) displays has complex, textured background or/and object of the small size compared to size of the background. The low performance in this case can be explained by the small fraction of detected patches that belong to the object.

Background images misclassified as object all display parts that are locally similar to bicycle parts due to significant texture whose rich structure ensures existence of these local similarities. The misclassification can be explained by observing the patches that contribute to misclassification (patches that are weighted towards prediction "same as object class"). Our matching procedure is not constrained to one-to-one matches and hence it is possible to have *popular patches* in query image – patches that are matched to many patches in focal image (e.g. in the case of misclassifier motorbike brake handle is matched to parts of bycicle's frame). If this patch pair is similar in the way the object patch pairs are then it will be assigned to cluster that is weighted for object class

Table 4. Results of classifier combination

Combination rule	max	min	median	sum	product	weighted sum (linear SVM)
EER of PR	0.84	0.88	0.86	0.86	0.86	0.88

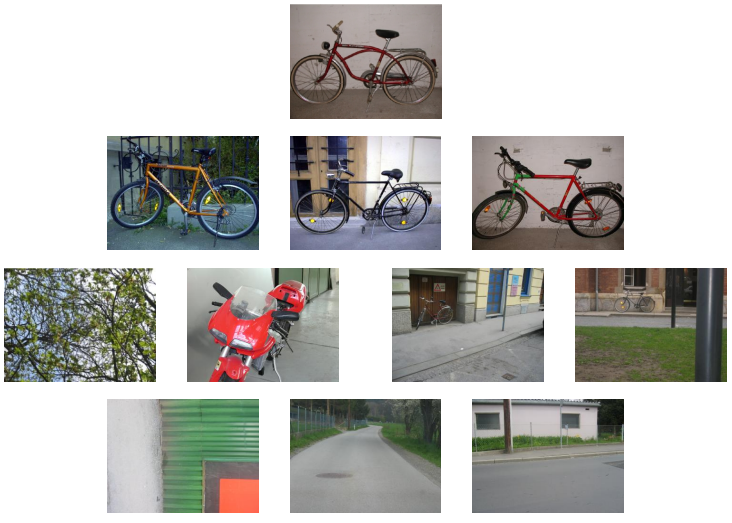

Fig. 3. Results for first focal image. First row: focal image, second row: first 3 most similar images, third row: two false positive followed by two false negative images, fourth row: last 3 most similar images.

prediction and wrong decision will be brought. To solve this problem a better matching procedure has to be employed, perhaps also including simple geometric constraints.

As expected, images which are misclassified for all focal images are misclassified in combination, which means that we could get better results using combination of classifiers only if we improve the performance of base classifiers.

5 Conclusion

We have given an on overview of approaches for image description and image similarity measures used for content-based image retrieval.

The matching in the case of object classes is not well defined problem since definition of "corresponding part" is ambiguous. It has been shown that matching strategy, which determines "corresponding parts", severely influences the results, and we believe that better matching strategy could improve the results.

We employed divide-and-conquer strategy through use of focal images, and we have shown that in the case of complex problem of object class recognition (used here for category tagging) it has shown to be beneficial.

Additional work is required to fully validate these preliminary results. First, it would be interesting to experiment the approach with more categories and evaluate the performance on large scale problems. Second, as the question of finding best correspondences between categories is the most critical part of the proposed algorithm, it would be interesting to try to learn to find these correspondences using labeled sets of corresponding parts.

References

1. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. IEEE Trans. Pattern Anal. Mach. Intell. 24(4), 509–522 (2002)
2. Berg, A.C., Malik, J.: Geometric blur for template matching. In: CVPR, vol. 1, pp. 607–614 (2001)
3. Chang, C., Lin, C.: LIBSVM: A library for support vector machines (2001), `http://www.csie.ntu.edu.tw/~cjlin/libsvm`
4. Duin, R.P.W.: The combining classifier: To train or not to train? In: ICPR (2002)
5. Fritz, G., Seifert, C., Paletta, L.: A mobile vision system for urban detection with informative local descriptors. In: Computer Vision Systems (2006)
6. Frome, A., Singer, Y., Malik, J.: Image retrieval and classification using local distance functions. In: NIPS, pp. 417–424. MIT Press, Cambridge, MA (2007)
7. Gudivada, V.N., Raghavan, V.V.: Content-based image retrieval-systems. Computer 28(9), 18–22 (1995)
8. Harris, C., Stephens, M.: A combined corner and edge detector. In: Proc. of Fourth Alvey Vision Conf., pp. 147–151 (1988)
9. Jain, A.K., Vailaya, A.: Image retrieval using color and shape. Pattern Recognition (1996)
10. Johansson, B., Cipolla, R.: A system for automatic pose-estimation from a single image in a city scene. In: Int. Conf. Signal Proc. Pattern Rec. and Analysis (2002)
11. Jurie, F., Triggs, B.: Creating efficient codebooks for visual recognition. In: International Conference on Computer Vision (2005)
12. Kadir, T., Brady, M.: Saliency, scale and image description. International Journal of Computer Vision V45(2), 83–105 (2001)
13. Ke, Y., Sukthankar, R.: Pca-sift: a more distinctive representation for local image descriptors. In: CVPR 2004, pp. II: 506–513 (2004)
14. Leung, T., Malik, J.: Recognizing surfaces using three-dimensional textons. In: ICCV, vol. 2, pp. 1010–1017. IEEE, Los Alamitos, CA (1999)
15. Lindeberg, T.: Feature detection with automatic scale selection. Int. J. Comput. Vision 30(2), 79–116 (1998)
16. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vision 60(2), 91–110 (2004)
17. Manjunath, B.S., Ma, W.Y.: Texture features for browsing and retrieval of image data. IEEE Trans. Pattern Anal. Mach. Intell. 18(8), 837–842 (1996)
18. Mikolajczyk, K., Leibe, B., Schiele, B.: Local features for object class recognition. In: ICCV, vol. 2, pp. 1792–1799. IEEE, Los Alamitos, CA (2005)
19. Mikolajczyk, K., Schmid, C.: Scale and affine invariant interest point detectors. International Journal of Computer Vision 60(1), 63–86 (2004)
20. Moosmann, F., Triggs, B., Jurie, F.: Fast discriminative visual codebooks using randomized clustering forests. In: NIPS, pp. 985–992 (2007)
21. Niculescu-Mizil, A., Caruana, R.: Predicting good probabilities with supervised learning. In: ICML, pp. 625–632. ACM, New York, NY, USA (2005)
22. Nistér, D., Naroditsky, O., Bergen, J.: Visual odometry for ground vehicle applications. Journal of Field Robotics 23(1), 3–20 (2006)
23. Nowak, E., Jurie, F.: Learning visual similarity measures for comparing never seen objects. In: CVPR. IEEE, Los Alamitos, CA (2007)
24. Nowak, E., Jurie, F., Triggs, B.: Sampling strategies for bag-of-features image classification. In: European Conference on Computer Vision. Springer, Heidelberg (2006)

25. Obdrzalek, S., Matas, J.: Object recognition using local affine frames on distinguished regions. In: BMVA 2002, vol. 1, pp. 113–122 (2002)
26. Opelt, A., Pinz, A., Fussenegger, M., Auer, P.: Generic object recognition with boosting. IEEE Trans. Pattern Anal. Mach. Intell. 28(3), 416–431 (2006)
27. Platt, J.: Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. Technical report, Microsoft Research (1999)
28. Schmid, C., Mohr, R.: Local grayvalue invariants for image retrieval. IEEE Transactions on Pattern Analysis and Machine Intelligence 19(5), 530–535 (1997)
29. Sivic, J., Zisserman, A.: Video google: A text retrieval approach to object matching in videos. In: ICCV 2003, pp. 1470–1477 (2003)
30. Steger, C.: An unbiased detector of curvilinear structures. IEEE Trans. Pattern Anal. Mach. Intell. 20(2), 113–125 (1998)
31. van de Weijer, J., Schmid, C., Verbeek, J.: Learning color names from real-world images. In: CVPR (June 2007)
32. Winn, J., Criminisi, A., Minka, T.: Object categorization by learned universal visual dictionary. In: Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference, vol. 2, pp. 1800–1807 (2005)