# Query Classification in Content-Based Image Retrieval

Ilya MARKOV [a,1] and Natalia VASSILIEVA [b]
[a] *Department of Computer Science, Saint-Petersburg State University*
[b] *HP Labs, Saint-Petersburg*

**Abstract.** Image retrieval by using content analysis is known as a difficult task. In our previous studies [1] and [2] mixed-metrics were proposed in order to combine color and texture metrics for image retrieval task. It was shown that it is always possible to mark out the best mixed-metrics for every group of similar images and improve retrieval effectiveness. In order to get the proper mixed-metrics a particular query-image should be classified to one of the predefined groups of perceptually similar images and this should be done in the real-time mode while processing the query and retrieving the result. In our previous work [3] the highly specialized classification method was proposed to solve this task. In the current study Naive Bayes and SVM classifiers are discussed and applied to the mixed-metrics approach to image retrieval. Classification result for these classifiers in comparison with the classifier, proposed in [3], are presented.

**Keywords.** content-based image retrieval, image classification

## Introduction

Content-based image retrieval is a way of searching in image collections when no additional information about images is given. Every image in a collection is represented by one or several feature vectors. These feature vectors are numerical descriptors of image content. Feature vectors are built upon the information extracted from pixel values. Usually one feature vector represents one of the low-level image features (color, texture, shape). At a query time feature vectors are extracted from a query-image and compared against existing vectors of images stored in an image collection.

Common approach is to match vectors describing different low-level features independently. All vectors of an image collection describing the same feature form multidimensional vector space (feature space). To measure image similarity one should define metrics for every feature space. The main assumption here is that two images are similar to each other if and only if corresponding vectors are similar too in accordance with the metrics specified.

Many researchers showed that it is necessary to combine various features for effective image retrieval. Merging results obtained for the same query by using different metrics in different feature spaces is known as a challenging task.

[1]Corresponding Author: Ilya Markov, Saint-Petersburg State University, Universitetsky 28, 198504 Saint-Petersburg, Russia; E-mail: ilya.markov@gmail.com.

Color and texture are the common features which are used for searching in natural images. In [1] we proposed a technique to combine color and texture metrics taking into account a particular query-image. Weighed linear combination of color and texture metrics (mixed-metrics) is considered as a fusion function. Our approach is based on the hypothesis, proposed and proved in [1], that there are optimal weights to combine color and texture metrics for every query-image and these weights are unique for a given query. By using these optimal weights one can improve retrieval results. We showed that it is always possible to mark out the best mixed-metrics for every group of similar images (and thus for every query-image). In [2] it was shown that mixed-metrics based approach outperforms widely used CombMNZ data fusion method in some cases, and has close results in the others.

It was stated that in order to perform a search over an image collection by using mixed-metrics one should prepare training set which represents the collection well. The training set should be somehow divided into groups of similar images and best mixed-metrics for every group should be calculated. To perform a search itself, one should classify a query-image as belonging to one of the groups. After that a search can be performed by using mixed-metrics of the group which the query-image is classified to.

The approach described above improves retrieval efficiency only if classification task is solved. A simple and highly specialized solution for that task in the described context was proposed in our previous work [3]. According to that approach every group of perceptually similar images is represented with a set of ”centers” or centroids. A query-image classification is performed based on the distances between its feature vectors and groups' centroids. The better centroids represent the groups the more precise classification and retrieval are.

In this study we solve the same classification task by using Naive Bayes and SVM classifiers, which was chosen for their known classification performances in pattern recognition and CBIR context. Their applicability to the query image classification task is discussed and comparative classification results are given.

## 1. Related Works

Grouping images into semantic classes by using low-level visual features is an important task. Many researchers claim that solving this problem can not only significantly improve image retrieval results ([4], [5], [6]), but could also allow using more convenient query forms instead of query-by-example ([7], [8]).

The first researchers who followed the idea of general image classification proposed the usage of binary classifier to decide whether an image belongs to a certain semantic class or not. These studies include techniques that can decide whether an image is indoor/outdoor ([6], [9]) or city/landscape ([10]). Vailaya et al. [10] stated that multiple two-class classifications may be more feasible than multi-class classifications. The usage of weighted k-NN classifier is typical for these binary classification tasks.

The same authors summarized binary classifiers in [4], proposing binary hierarchy of vacation images, where the first level of the hierarchy is indoor/outdoor images, the second is city/landscape and other levels are related to the specific classes of natural scenes. Bayesian classifier is used to assign an image to its best related group for each level of the binary hierarchy.

Application of a Bayesian frameworks in CBIR is also discussed in [11], [12] and [13]. Cox et al. [11] developed the Bayesian image retrieval system, called PicHunter, which is a simple instance of a general Bayesian framework.

The usage of SVM classifiers in CBIR is widely discussed during past 10 years ([14], [15], [16], [17]). Chang et al. [14] summarized the problems, which arise when applying SVM techniques to the visual information retrieval. The first one is "scarcity of training data", when a number of training instances is less than a dimension of a feature space. The second problem is "imbalance of training classes", when a number of negative training instances is much greater then a number of positive ones. These problems make class prediction unreliable. Several SVM modifications are proposed to deal with stated challenges. Application of Kernel functions is discussed in [15], [16] and [17]. [14] proposes active learning, recursive subspace co-training and adaptive dimensionality reduction techniques.

## 2. Task Definition

In [1] we proposed a technique to combine image similarity measures which takes into account a particular query-image. We introduced mixed-metrics obtained from color and texture metrics ($C$ and $T$ respectively) by using their weighted linear combination $a \cdot C + (1-a) \cdot T$, where $a$ is a varying coefficient which depends on a query-image. We stated and proved the hypothesis that optimal value of $a$ is the same for similar query-images.

It was shown that in order to perform searches over an image collection by using mixed-metrics, training set, which represents the collection well, should be prepared. The training set should be somehow divided into groups of similar images and optimal mixed-metrics coefficient $a$ should be calculated for every group by using the algorithm proposed in [1].

To perform a search with a particular query-image, optimal coefficient $a$ should be found for the given query. Thereto the query-image has to be classified to only one of the groups of similar images defined on the previous step. The classification is performed in the real-time during retrieval process, therefore it should be as fast as possible and simple enough involving just a few computations. Thus we have two important requirements for the classification algorithm to be used:

- an object should be classified to exactly one of the groups;
- computational complexity of the prediction stage should be low.

## 3. Classification Algorithms

The goal of classification is to take an input vector $x$ and to assign it to one of $K$ discrete classes $C_k$ where $k = 1, ..., K$ [18]. In the most common scenario classes are disjoint, so that each input is assigned to one and only one class. Binary classification can be considered as a special case of multi-class problem when the number of classes $K = 2$. Well-known classification methods including decision trees, Bayesian networks, support vector machines and neural networks were initially devised for the binary case. Although many binary classifiers can be naturally extended to multi-class domain by using winner-take-all strategy [19], this can be computationally heavy when the number of classes is big. When it is not possible to directly extend a binary method to a multi-class modification or the resultant method is computationally complex, multi-class classifiers are often built based on a set of binary classifiers by using one-versus-the-rest, one-versus-one or error-correcting-output-coding strategy.

### *3.1. Centroid-Based Classifier*

In [3] we proposed a simple and highly specialized solution for the classification task, described in section 2, which satisfies all the stated requirements. According to that approach every group of perceptually similar images is represented with a set of centroids. One centroid is built for every feature. In our case color and texture features are combined, therefore two centroids have to be built for every group: one representing "central" color of the group and other representing "central" texture.

According to the "best image in the group" centroid building method, proposed in [3], color/texture group's centroid should be an image, belonging to the group, having the highest precision when performing a search by using color/texture metrics with that image as a query. Building algorithm is the following: for every image in the group a search by using color/texture metrics should be performed and retrieval precision should be calculated. Images from the same group as the query-image are treated as relevant, while others are not. Image with the highest precision is treated as a color/texture centroid of the group.

At a search-time the distances between query-image and groups' centroids are calculated. The query is classified to the group with the nearest set of centroids. Therefore the computational complexity of the proposed classification algorithm is equaled to the number of groups multiplied by the number of features (one centroid per feature) because the query is compared to the groups' centroids only.

### *3.2. Naive Bayes Classifier*

Naive Bayes is a simple probabilistic learning algorithm based on applying Bayes' theorem with the following "Naive" assumption: distributions of input features are assumed to be independent. This assumption is rarely true in the real-world applications, but in spite of this Naive Bayes classifier performs surprisingly well in many complex classification tasks.

In order to perform multi-class classification of an input, Naive Bayes algorithm computes a posterior probability that the input belongs to a class for every class in the system. The result of the classification is the class with the highest posterior probability.

An advantage of the Naive Bayes classifier is that it requires a small amount of training data to estimate the parameters necessary for classification. Its disadvantage in the CBIR area is that Bayesian probability model operates with discrete features. Therefore image features should be discretized first before using Naive Bayes classifier, which is not a trivial task.

### *3.3. SVM Classifier*

Support vector machines are a set of supervised learning methods used for classification and regression. The goal of SVM classifier is to find the best hyperplane separating classes. The best hyperplane has the maximum distances to the nearest data points from the classes to be separated.

Initial SVM method is linear and assumes that points from different classes are linearly separable. Several extensions to the initial case were proposed including soft margin extension to deal with noisy data sets and non-linear SVM classifier (also known as Kernel Machine) to deal with non-linear spaces such as image feature vector spaces.

Although Kernel Machines method is known to be a very powerful non-linear classifier, it has several disadvantages. It is computationally demanding to train and, what is more crucial in our case, to run. Sensitivity to noisy data is among other known disadvantages, which is also critical for the task, described in section 2.

## 4. Experiments and Result Analysis

We use experimental image database from [2] which consists of 650 images from Corel Photo Set collection. It is divided into 9 groups based on images content by 2 experts. Resultant groups are: City, Clouds, Coasts, Buildings, Fields, Lakes, People, Rocks and Trees. This set of images can be considered as a training set for some larger collection.

For every image in the database color and texture features are extracted and for every pair of images color and texture distances are computed. Metrics values are normalized according to the following rule:

$$metrics_{result}(image) = (metrics(image) - Average)/Deviation. \quad (1)$$

Therefore color and texture metrics values distributions have the same $Average$ and $Deviation$.

We reuse classification results for centroid-based classifier from [3], where every image is classified to exactly one of the groups. On the contrary, only binary versions of Naive Bayes and SVM classifiers are used because of their simplicity in comparison with multi-class modifications. Therefore they perform as many classifications as many groups we have. However the binary case still shows the main trends of using such classifiers in the area of content-based image retrieval.

We use Peltarion Synapse framework [20] to get the results for Naive Bayes and SVM classifiers. Synapse utilizes Kernel Adatron SVM modification because it is reasonably fast in comparison to other SVM models.

Comparative classification results are shown in Table 1. Naive Bayes and SVM classifiers are run with the color information only because of the dimensionality problem, which is discussed below in this section.

It can be seen, that Naive Bayes classifier tends to reject most of the images, which results in the high false negatives rate. It means that Naive Bayes classifier can simply reject an image from the right group. Exceptions are ”Clouds” and ”Rocks”, where false negatives rate is low. It is very natural, that clouds can be easily separated by color from all other groups, therefore all the classifiers perform good on this class with low both false positives and false negatives rates.

Vise versa, SVM classifier accepts most of the images and has low precision of positives and high false positives rates in most cases. It means that SVM classifier can easily misclassify an image and put it to the wrong group. Therefore we cannot trust the classification results obtained by using SVM classifier too.

**Table 1.** Classification results for Centroid-Based, Naive Bayes and SVM classifiers (%).

| Group | Precision of positives | | | Precision of negatives | | |
|---|---|---|---|---|---|---|
| | Centroids | Bayes | SVM | Centroids | Bayes | SVM |
| City | 19 | 100 | 6 | 92 | 28 | 75 |
| Clouds | 86 | 96 | 95 | 97 | 93 | 99 |
| Coasts | 44 | 89 | 21 | 95 | 25 | 76 |
| Buildings | 71 | 98 | 37 | 87 | 31 | 71 |
| Fields | 66 | 94 | 40 | 92 | 25 | 74 |
| Lakes | 56 | 81 | 28 | 91 | 23 | 76 |
| People | 35 | 100 | 18 | 99 | 43 | 76 |
| Rocks | 28 | 100 | 43 | 100 | 87 | 98 |
| Trees | 78 | 94 | 51 | 94 | 37 | 49 |
| **Group** | **False positives** | | | **False negatives** | | |
| | Centroids | Bayes | SVM | Centroids | Bayes | SVM |
| City | 81 | 0 | 94 | 8 | 72 | 25 |
| Clouds | 14 | 4 | 5 | 3 | 7 | 1 |
| Coasts | 56 | 11 | 79 | 5 | 75 | 24 |
| Buildings | 29 | 2 | 63 | 13 | 69 | 29 |
| Fields | 34 | 6 | 60 | 8 | 75 | 26 |
| Lakes | 44 | 19 | 72 | 9 | 77 | 24 |
| People | 65 | 0 | 82 | 1 | 57 | 24 |
| Rocks | 72 | 0 | 57 | 0 | 13 | 2 |
| Trees | 22 | 6 | 49 | 6 | 63 | 51 |

**Table 2.** Classification results of using color, texture and both features with Naive Bayes classifier (%).

| Group | Precision of positives | | | Precision of negatives | | |
|---|---|---|---|---|---|---|
| | Color | Texture | Both | Color | Texture | Both |
| City | 100 | 4 | 4 | 28 | 99 | 99 |
| Clouds | 96 | 0 | 0 | 93 | 100 | 100 |
| Coasts | 89 | 2 | 1 | 25 | 99 | 99 |
| Buildings | 98 | 2 | 2 | 31 | 100 | 100 |
| Fields | 94 | 2 | 2 | 25 | 99 | 99 |
| Lakes | 81 | 6 | 8 | 23 | 97 | 97 |
| People | 100 | 0 | 0 | 43 | 100 | 100 |
| Rocks | 100 | 0 | 0 | 87 | 100 | 100 |
| Trees | 94 | 13 | 11 | 37 | 97 | 97 |

Centroid-based classification results lay between those borderline cases described above having acceptably high precision of positives in most cases, but high false positives rate in some of them.

All the classifiers face the "imbalance of training classes" problem here. The training set consists of 650 images divided into 9 groups, therefore the number of positive instances for each group is 8 times less on the average than the number of negative ones.

The results shown in Table 1 are obtained by using color information only for Naive Bayes and SVM classifiers. Comparative results of using color, texture and both features together are shown in Table 2 and Table 3.

**Table 3.** Classification results of using color, texture and both features with SVM classifier (%).

| Group | Precision of positives | | | Precision of negatives | | |
|---|---|---|---|---|---|---|
| | Color | Texture | Both | Color | Texture | Both |
| City | 6 | 19 | 19 | 75 | 87 | 87 |
| Clouds | 95 | 22 | 22 | 99 | 87 | 87 |
| Coasts | 21 | 22 | 22 | 87 | 87 | 87 |
| Buildings | 37 | 22 | 22 | 71 | 86 | 86 |
| Fields | 40 | 47 | 47 | 74 | 86 | 86 |
| Lakes | 28 | 35 | 35 | 76 | 85 | 85 |
| People | 18 | 0 | 0 | 76 | 88 | 88 |
| Rocks | 43 | 10 | 10 | 98 | 87 | 87 |
| Trees | 51 | 20 | 20 | 49 | 87 | 87 |

The "scarcity of training data" problem is seen here. Color feature is represented with 45-dimension vector which is much less than the size of the training set (650 images). Therefore both classifiers show acceptable results when using color feature only. Texture has 714 dimensions which is greater than 650. Therefore both Naive Bayes and SVM classifiers cannot separate groups from each other, showing nearly constant precision across all the groups when using texture or both features together. To use texture feature with these classifiers the dimension should be decreased, which takes additional resources and does not feet the requirement that computational complexity of the prediction stage of a classifier should be low.

For centroid-based classifier dimensionality problem is hidden inside the metrics calculation and affects only metrics evaluation time, therefore proposed classifier does not depend on the feature space dimension at all.

## 5. Conclusion

In this study we addressed query image classification problem applied to the mixed-metrics approach developed in [1]. It was shown that Naive Bayes and SVM classifiers cannot be directly applied to solve the classification task described in 2, while centroid-based classifier, proposed in [3], suit that task well, being very simple and still showing acceptably good classification results in comparison with Naive Bayes and SVM classifiers.

## Acknowledgments

This work was partially supported by RFBR (grant 07-07-00268a).

## References

[1] Markov, I., Vassilieva, N., & Yaremchuk, A. (2007) Image retrieval. Optimal weights for color and texture features combining based on query object (In Russian). In Proceedings of 9th Russian Scientific Conference on Digital Libraries (RCDL 2007), October 16–18, Pereyaslavl-Zalessky, Russia, pp. 195–200.

[2] Markov, I., & Vassilieva, N. (2008) Image retrieval. Color and texture combining based on query-image. In Proceedings of International Conference on Image and Signal Processing (ICISP 2008), July 1–3, Cherbourg-Octeville, France, pp. 430–438.
[3] Markov, I., & Vassilieva, N. (2008) Building Up Low-Level Centroids for Groups of Perceptually Similar Images. In Proceedings of 8th International Baltic Conference on Databases and Information Systems (Baltic DB&IS 2008), June 2–5, Tallinn, Estonia, pp. 341–348.
[4] Vailaya, A., Figueiredo, M.A.T., Jain, A.K., & Zhang, H.-J. (2001) Image classification for content-based indexing, IEEE Transactions on Image Processing, 10(1): 117–130
[5] Kaick, O., & Mori, G. (2006) Automatic Classification of Outdoor Images by Region Matching. In Proceedings of 3rd Canadian Conference on Computer and Robot Vision (CRV 2006), June 7–9, Quebec, Canada.
[6] Serrano, N., Savakis, A., & Luo, L. (2002) A computationally efficient approach to indoor/outdoor scene classification. In Proceedings of 16th International Conference on Pattern Recognition (ICPR 2002), August 11–15, Quebec, Canada, pp. 146–149.
[7] Vassilieva, N., & Novikov, B. (2004) A Similarity Retrieval Algorithm for Natural Images. In Proceedings of 6th International Baltic Conference on Databases and Information Systems (Baltic DB&IS 2004), June 6–9, Riga, Latvia.
[8] Vassilieva, N., & Novikov, B. (2005) Establishing a correspondence between low-level features and semantics of fixed images (In Russian). In Proceedings of 7th Russian Scientific Conference on Digital Libraries (RCDL 2005), October 4–6, Yaroslavl, Russia, pp. 236–240.
[9] Szummer, M., & Picard, R.W. (1998) Indoor-outdoor image classification. In Proceedings of IEEE International Workshop on Content-Based Access of Image and Video Database (CAIVD 1998), January 3, Bombay, India, pp. 42–51.
[10] Vailaya, A., Jain, A.K., & Zhang, H.-J. (1998) On Image Classification: City Images vs. Landscapes, Pattern Recognition, 31(12): 1921–1935
[11] Cox, I.J., Miller, M.L., Minka, T.P., Papathomas, T.V., & Yianilos, P.N. (2000) The bayesian image retrieval system, PicHunter: Theory, implementation and psychophysical experiments, IEEE Transactions on Image Processing, 9(1): 20–37
[12] Heller, K.A., & Ghahramani, Z. (2006) A Simple Bayesian Framework for Content-Based Image Retrieval. In Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006), June 17–22, New York, USA, pp. 2110–2117.
[13] Vasconcelos, N., & Lippman, A. (1998) A bayesian framework for content-based indexing and retrieval. In Proceedings of Conference on Data Compression Conference (DCC 1998), March 30–April 1, Snowbird, USA.
[14] Chang, E., Li, B.T., Wu, G., & Goh, K.S. (2003) Statistical learning for effective visual information retrieval. In Proceedings of IEEE International Conference on Image Processing (ICIP 2003), September 14–18, Barcelona, Spain, pp. 609–612.
[15] Chapelle, O., Haffner, P., & Vapnik, V. (1999) SVMs for histogram based image classification, IEEE Transactions on Neural Networks, 10: 1055–1064
[16] Chen, Y., Zhou, X.S., & Huang, T.S. (2001) One-class SVM for learning in image retrieval. In Proceedings of IEEE International Conference on Image Processing (ICIP 2001), October 7–10, Thessaloniki, Greece, pp. 34–37.
[17] Gosselin, P.H., & Cord, M. (2004) A Comparison of Active Classification Methods for Content-Based Image Retrieval. In Proceedings of 1st International Workshop on Computer Vision meets Databases (CVDB 2004), June 13, Paris, France.
[18] Bishop C.M. (2006) Pattern Recognition and Machine Learning (Information Science and Statistics). Springer Science and Business Media.
[19] Aiolli, F., & Sperduti, A. (2005) Multiclass Classification with Multi-Prototype Support Vector Machines, Journal of Machine Learning Research, 6: 817–850
[20] Peltarion Synapse documentation, (available at http://www.peltarion.com/WebDoc)