A Bayesian Framework for Recognizing Textured Objects in a Content-Based Image Retrieval System

Victor Eruhimov, Maria Lyashko, Elena Martinova, Sergey Molinov
Intel Russia Researh Center
Nizhny Novgorod, Russia
{Victor.Eruhimov, Maria.Lyashko, Elena.Martinova, Sergey.Molinov}@intel.com

Abstract

We present an image retrieval system for finding textured objects by text query. The algorithm considers each image as a set of independent segments. It learns the relationship of segments features with text from a training set of images where a set of segments is manually labeled. The algorithm is capable of generating text labels from a segment and finding images that are relevant to a query consisting of text labels. We present the first experimental results.

Keywords: *Image retrieval, image segmentation, mean shift, Bayesian approach.*

1. INTRODUCTION

In the last ten years fast growth of digital image collections, both commercial and private, stimulated creation and development of image retrieval systems as special tools for effective image collection managing. These systems resolve the following two typical tasks:

- image search by query (in terms of object, image features, textual description or query by example);
- automated image annotation.

The first task is important for image collection organization, browsing support (for instance, on web-pages of museums), and for auto-illustration. The second one may be important for service of huge image repositories. In all cases it is desirable to decrease necessarily of manual operations of any kind – both in time of adding image to the repository or browsing and search.

There are some typical stages in any image retrieval system:

- image preprocessing: analysis and extraction of key information;
- data analysis and classification;
- image search by query.

One of the most important and challenging problem for the systems is contradiction between high level user categories from real world and low level image features (like color, texture, pixel coordinates etc.), which are usual for operation of image analysis algorithm. The reason of this so-called "semantic gap" lies, in our opinion, in the current state of object recognition methods that cannot work in high-dimensional spaces such as image spaces. Also a reliable recognition algorithm very often requires a huge base for training. The appearance of real life objects is diverse and complex. Computer vision still has no universal methods effective enough to be comparable with human recognition and learning capability.

At the same time in the last few years the progress in recognition and mining methods allowed to create image retrieval systems with real practical usage. The evolution took place for all system components: image mining and indexing, image properties modeling, search methods and query forms. The first systems started from search by keywords, like Corbis [1], Altavista Photofinder [2], WebSeer [3]. Later in order to have possibility to search by image content, different methods for content analysis were developed. We will refer to these methods as Content-Based Image Retrieval (CBIR). [4] uses correlation between extracted color and textual features. In CANDID [5] color, texture and shape features are determined at every pixel, the feature vectors of all pixels together form a point set in higher-dimensional space. The clusters are formed, and mean vector and covariance matrix are computed for each cluster. Each image is represented by signature consisting of a weighted sum of Gaussian functions. Matching between query image and analyzed one is based on calculation of normalized Euclidean distance as dissimilarity between two image signatures. This approach was improved in the Blobworld [6], where fully automated image segmentation is used. Some very special approaches were realized. For instance, in ImageFinder [7] matching is done using a Boltzman Machine, a special kind of probabilistic artificial neural network. The network must first be trained on a set of images. In some cases wavelets are used for image structure description and matching, see WISE [8] as example.

At the same time image matching methods were developed. In many modern commercial systems concepts of image similarity and distance between images are used (Blobworld, QBIC[9], CLUE [10]). Query form also changed and became more complex. Now it is possible to search an image by example, by sketches or/and selected color, texture pattern, keywords or use these possibilities in any combination. But it is still very difficult to get the image that contains certain object, if this image was not previously marked with the corresponding keyword. In order to resolve this problem some attempts were done to incorporate the knowledge about real world into the system. Barnard et al. [11] presented a statistical model for organizing image collections which integrates semantic information provided by associated text and visual information. Text analysis algorithm produces hierarchical relationships between words. While the model learns relationships between image features and text, the corresponding image segments also get hierarchical relationships. These relations are used in probabilistic model and promise to increase search effectiveness. There are some papers, in which authors declared "object oriented approach" to image retrieval, but in fact incorporated in their models some kind of hierarchy [12-13].

In our opinion it is hardly ever possible in modern state of computer vision to have method, which would be constantly successful for recognition of all kinds of objects – both having complex structure (faces, cars, animals etc.) and spatially-homogeneous (like sky, sea, grass and sand). It looks more promising to use separate methods and image data models for different tasks. As example of such approach Photobook[14] system may be considered. There are three different approaches in this system for faces, 2D shapes and textured images modeling

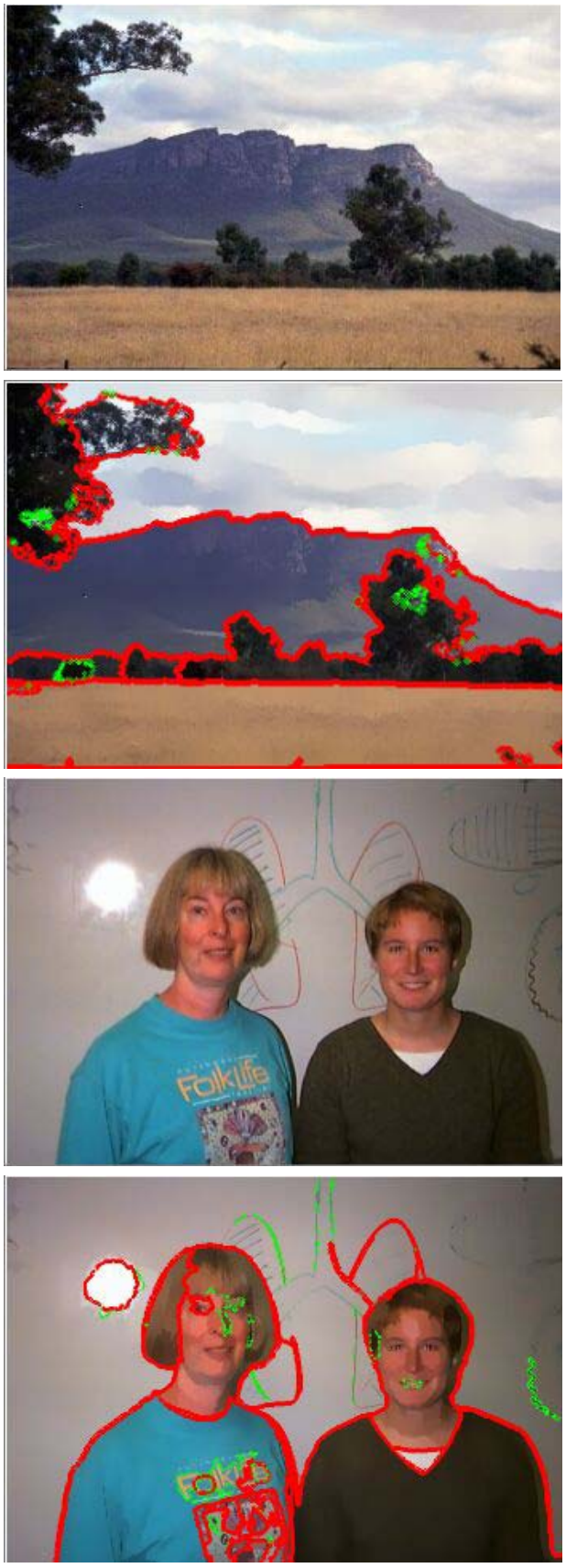

Figure 1: **Examples of segmentation results.**

and search.

The final goal of presented work is the creation of a system, which will be convenient for use in home segment. It must:

- efficiently auto-annotate input images;
- search by queries in the terms of keywords and concrete objects.

This paper describes preliminary results of application to the image retrieval of simple segment-label model and its acceptability appraisal. The work describes the use of statistical approach to the resolving of annotation (section 2) and image retrieval (section 3) tasks. The model of system performance evaluation is described. Some results, obtained for training base, are presented in section 4.

2. ANNOTATION TASK

2.1 General scheme

The task of image annotation can be formulated as creation of image description based on image content. This task assumes that the model of correspondence between images and textual description was built. The simplest task of image annotation is to find a set of separate words (without a linguistic model) corresponding to objects on the image. The training data may consist of image set with set of words assigned either to the whole image (which means this word/object is present on the image) or to the certain image parts. The former makes the task of training database creation quite simple (just specifying the set of appropriate words) but probably requires a very complex model. The later requires image segmentation and more manual work for segment labeling, but hopefully this approach enables building quite a simple model for words-segments correspondence.

In our work we use the following scheme for the task of image annotation. On the training phase we

1) split images into segments using one of image segmentation algorithms;
2) manually assign one or several words to segments of interest;
3) build compact representation of every labeled segment by extracting useful features like texture, color and shape descriptors; the set of segment features forms a real-valued vector in a multidimensional space;
4) finally we build a statistical model that represents a joint probability distribution of words and segments.

Recognition phase consists of finding the most probable word or set of words for a given feature vector (obtained by steps 1 and 3).

2.2 Segmentation

For image segmentation the ‘Mean-Shift’ algorithm [15] was chosen as an appropriate combination of algorithm quality and implementation availability. The algorithm divides the input image into ‘color homogeneous‘ regions, inside which color may vary not more than defined by a given parameter. Every segment may be marked with one or several words. For example forest can be marked both by ‘forest’ and ‘tree’ etc. Segmentation works stable and gives reasonable results for objects with uniform color (sea, sky). It is possible to achieve appropriate segmentation for some nonuniform objects using additional image smoothing prior segmentation (trees, animals, grass). Obviously the algorithm can’t represent people wearing multicolored clothes as single segment. The same problem takes place for any object with

complex structure and/or nonuniform illumination. Some examples of segmentation results are shown on Fig.1.

2.3 Feature extraction

Color, texture and shape are usually used as the most natural segment descriptors. All these characteristics are well-studied and there are number of ways to represent them.

The classical way to represent 2D shape is to compute spatial moments of the shape. Although general spatial moments are sensitive to shape scale and orientation, they can be combined in order to get independence of scale and orientation. The well known Hu moments are used as such combinations.

The color of the segment can be represented by computing the average over segment and this is the natural way for uniformly colored objects. Also color histogram can be built and it would give more accurate representation of the color distribution for segment. We build normalized (i.e. all elements are summed to 1) color histogram in 2D subspace (chromatic components CrCb) of YCrCb color space. The number of histogram elements extracted into features vector depends on color space quantization and presence of different colors in training segments. In our experiments we use from 8 to 75 features provided by color histograms.

We used texture descriptors suggested in [6]: two real numbers expressing the average values of anisotropy and contrast of image area. It is possible to use the histogram of these parameters over segment.

2.4 Joint segments-text model

After feature vector extraction there are a set of multidimensional vectors and a set of text labels. Each vector is assigned with one label[1]. We need to build a model which for any input vector will give corresponding label. In other words, we want to build a classifier that sorts query vectors into classes corresponding to different labels. We use a so-called naive Bayesian approach. The probability distribution of a feature vector f corresponding to a specific label l is modeled by a multivariate Gaussian distribution $N(\bar{f}_l, \Sigma_l)$ with a mean vector $\bar{f}_l$ and a covariance matrix Σ_l. The joint features-labels probability distribution is modeled with a Gaussian multiplied by a constant indicating the frequency of each label: $P(l, f) = a_l N(\bar{f}_l, \Sigma_l)$. The model is trained by fitting the feature vectors corresponding to the same label with a Gaussian. Inference for a given feature vector f is based on comparing the values of the joint distribution function for different labels. The resulting label is defined by $l^*(f) = \arg\max_l P(l, f)$. One of the main problems with this approach is its instability in high dimensional spaces so we tried to limit ourselves to a small set of features, 2 texture and 2 color. Surprisingly our experiments with an extended set of features that included histogram values and more powerful classifiers like Gradient Boosting Trees [16] that is much more stable in high-dimensional spaces did not bring substantial improvement to the results.

[1] Although there are no theoretical limitations of having multiple labels assigned to a single feature vector, for the sake of simplicity we will restrict our consideration to a single label case.

3. IMAGE RETRIEVAL TASK

The annotation problem being quite important for a CBIR system does not solve the image retrieval problem directly. A CBIR system should return an ordered list of items based on their relation to the query. A query in the framework of this paper is a set of conditions indicating whether a specific label is present or absent in the required images. Thus the relevance of the image to the query can be defined as a Boolean variable. A good algorithm will put forward items that a user needs most judging from the query and will not show items that are not relevant at all. At this point we would like to define a metric or error for a CBIR system showing how well a particular algorithm performs in response to a given query. Later we will show that if our annotation task is solved by a Bayesian classifier (i.e. optimally) then we can build a simple algorithm for minimizing the expectation of CBIR error under the assumption of independent segments.

3.1 Image Retrieval performance metric

Let our database of labeled images D be divided into a training set D_{TR} and test set D_{TS}. The CBIR algorithm learns the relationship of images and labels on the training set and then searches for images relevant to the given query in the test set. The output of the algorithm is an ordered set of images. It is important that the algorithm is unaware of labels in D_{TS}. The measure of how well the algorithm performs should satisfy two obvious conditions:

1. The images that are more likely to be relevant to the query should be put forward in the output list
2. The images that are less relevant should be either put back in the list or not included in the list.

We suggest a simple metric that satisfies these two conditions. For the output list L with a fixed size the metric looks like the following:

$$E_1 = \sum_{i \in L} s_i p_i \qquad (1)$$

Here E_1 is a non-negative function that has a meaning of error i.e. it takes smaller values when the algorithm performs better. i enumerates images from the output list, s_i is equal to 0 if the image i is relevant to the query and 1 otherwise, p_i is a penalty function that is positive and monotonically decreasing. The cost function (1) penalizes for the images irrelevant to the query and the penalty is larger for the images in the beginning of the list. Obviously for the output list with a dynamic size the cost function should have a negative term that rewards relevant images:

$$E = \sum_{i \in L} \left(s_i p_i - (1 - s_i) r_i \right) \qquad (2)$$

Here r_i is a reward function that is positive and monotonically decreasing. The balance between the reward and penalty functions

defines the risk of putting the image into the output list. One can see that (2) contains a constant reward for including an image into the list in addition to the penalty and reward for relevance.

3.2 CBIR algorithm

Following section 2 we will assume that the segments are independent, i.e. the probability distribution of labels $P(l \mid S \in I)$ on a particular segment S in the image I depends only on this segment. The probability distribution of labels on a given image is as follows:

$$P(l \mid I) = 1 - \prod_{S \in I} (1 - P(l \mid S)) \quad (3)$$

Let us assume that we have a Bayesian classifier for annotation task, i.e. we know the correct values of $P(l \mid S)$ for every segment in every image from the test set. Then for a given list of images $L = \{I_i, i = 1...n\}$ it is straightforward how to calculate the expectation of error (1,2) if the query contains a single label l:

$$\langle E \rangle = \sum_{i=1}^{n} [(1 - P(l \mid I_i)) p_i - P(l \mid I_i) r_i] \quad (4)$$

(4) suggests a simple CBIR algorithm that is optimal in terms of (2). We will consider a simple realistic version where $r_i = \alpha\, p_i$, $\alpha > 0$:

1. Calculate $P(l \mid I)$ for every $I \in D_{TS}$
2. Build $L = \left\{ I \in D_{TS} \mid P(l \mid I) > \frac{1}{1+\alpha} \right\}$ with items ordered by decreasing $P(l \mid I)$.

Note that the expectation (4) and the CBIR algorithm can be easily generalized for the query consisting of several conditions.

4. EXPERIMENTAL RESULTS

We will present results for both annotation task and image retrieval task. The experimental setup is based on a database of about 50 images. Each image is segmented and a subset of segments is labeled manually. The set of labels is chosen in a way that each object can be identified with only one label. As a result the dataset consisted of about 800 segments and 14 labels, each label corresponding to at least 10 segments. The dataset was divided into training and test sets. The training set was used for learning a joint model of segments and labels. Then the annotation algorithm and image retrieval algorithm were run on the test set as if it is not labeled.

The annotation algorithm was run using the Gaussian model together with two texture features and two color channels (Cr and Cb) averaged over segments. The misclassification rates for different labels are presented in Table 1. One can see that the algorithm recognizes several labels sufficiently well but fails on the rest. There are several reasons for such a behavior. First, the segmentation often puts several objects or parts of objects into a single segment. Second, several objects cannot be recognized by themselves without context. A good example is "building" and "rock". They have similar feature values and in certain cases only image context can help. Table 2 presents the distribution of labels predicted by the annotation algorithm for segments with the same true "observed" label. Note that "rock" is often misclassified as "building" and "bush" is almost always misclassified as "tree".

label	misclassification rate
building	0.31
cliffs	0.93
sky	0.12
sea	0.68
rock	0.49
grass	0.85
tree	0.21

Table 1. Misclassification rate for different labels in annotation task.

Obs\pred	building	sky	rock	tree	bush
building	81.48%		7.41%	11.11%	
sky		100%			
rock	35.72%		57.14%	7.14%	
trees	3.70%		7.41%	88.89%	
bush				80%	20%

Table 2. Distribution of predicted labels (in rows) corresponding to the same observed label in the annotation task.

We present the results for the image retrieval task for queries consisting of a single label. Figure 2 presents the results for the query "building". Note that first four images are related to the query. There are several images with rocks returned. One can see from the annotation task that buildings are often mixed up with rocks. We think this is because they are close to each other in our feature space. Figure 3 illustrates how different segments vote for the image to be included into the result of the query. It shows the percentage of the segments from image 1, Fig.2, that have annotation probability corresponding to "building" label in a certain range of values (horizontal axis). About 50% of the segments have probability of being labeled as "building" greater than 0.9 and they all contribute to the overall rating for the image. Fig.4 shows the results for the query "tree".

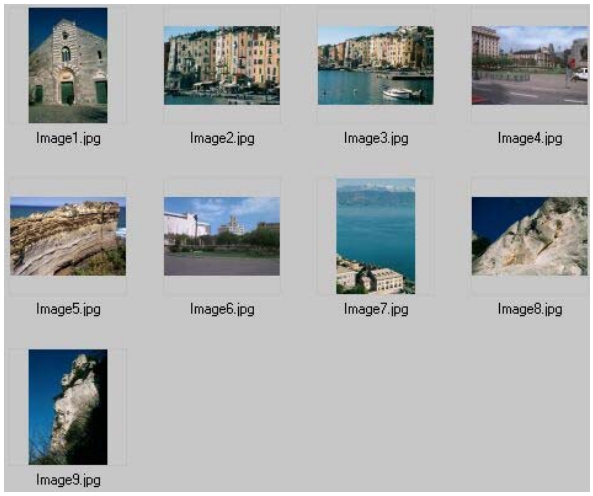

Figure 2. Results of image retrieval task with the query "building".

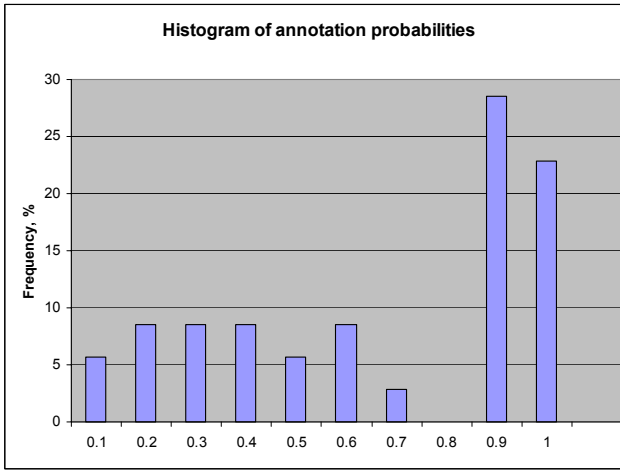

Figure 3. Histogram of probabilities of different segments from image 1, Fig.2, to be labeled as "building"

Again several first images contain trees. Note that although images 1 and 3 in Fig.2 contain segments with trees they are not the central part of the image. It is important that our image retrieval system does not give preference to large segments. We believe the size of the segment should be a separate parameter in the query as in [6]. Figure 4 shows the results of the query "sea". Both images are not relevant to the query, obviously the "sea" has been mixed up with the "sky" label. However the list of images is much shorter, the system cuts off the images with low probabilities values (3). One can adjust the threshold α balancing between the size of the result and the confidence level.

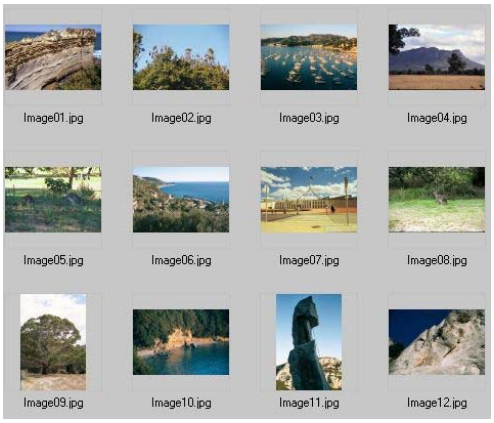

Figure 4. Results of image retrieval task with the query "tree".

Figure 5. Results of image retrieval task with the query "sea".

5. CONCLUSION

We have demonstrated preliminary results for the image retrieval system based on learning the relationship between segments and text labels. We show that even under simple assumptions of segments independence, simple features and Gaussian model one can obtain relatively good results. Our future work will go in several directions. We will explore the model with segments interaction. For instance the boat is usually found somewhere near the sea, the sky is often near the clouds. We will investigate other segmentation algorithms that can learn segmentation from examples. Finally we will add other recognition modules that can

find complex objects like human faces and fuse the data from several recognition modules into a single rating.

6. REFERENCES

[1] http://www.corbis.com/corporate/overview/overview.asp

[2] http://image.altavista.com/cgi-bin/avncgi.

[3] Michael J. Swain, Charles Frankel, and Vassilis Athitsos. Webseer: An image search engine for the world wide web. Technical Report TR-96-14, Department of Computer Science, University of Chicago, July 1996.

[4] Mori, Y., Takahashi, H. and Oka, R., 1999. Image-to-word transformation based on dividing and vector quantizing images with words, First International Workshop on Multimedia Intelligent Storage and Retrieval Management (in conjunction with ACM Multimedia Conference 1999), Orlando, Florida.

[5] P. M. Kelly, T. M. Cannon, and D. R. Hush. Query by image example: the CANDID approach. In SPIE Vol. 2420, Storage and Retrieval for Image and Video Databases III, pages 238-248, 1995.

[6] C. Carson, S. Belongie, H. Greenspan, and J. Malik. Blobworld: Image segmentation using expectation-maximization and its application to image querying. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(8):1026–1038, 2002.

[7] http://attrasoft.com/abm3_4.html.

[8] James Ze Wang, Gio Wiederhold, Oscar Firschein, and Sha Xin Wei. Wavelet-based image indexing techniques with partial sketch retrieval capability. In Proceedings of the Fourth Forum on Research and Technology Advances in Digital Libraries, Washington D.C., May '97, pages 13-24, 1997. http://www-db.stanford.edu/ wangz/project/imsearch/ADL97

[9] M. Flickner, H. Sawhney,W. Niblack, J. Ashley, et al. Query by image and video content: The QBIC system. *IEEE Computer*, 28(9):23–32, Sept. 1995.

[10] Yixin Chen, James Z. Wang and Robert Krovetz, ``Content-Based Image Retrieval by Clustering," Proc. 5th International Workshop on Multimedia Information Retrieval, in conjunction with ACM Multimedia, pp. 193-200, Berkeley, CA, ACM, November 2003.

[11] Barnard, K., P. Duygulu, D. Forsyth, N. de Freitas, D. Blei, and M. Jordan, "Matching Words and Pictures", Journal of Machine Learning Research 3, (2003), 1107–1135.

[12] T. Y. Lui & E. Izquierdo, "Scalable Object-based Image Retrieval", IEEE International Conference on Image Processing, ICIP 2003, Barcelona, Spain Sep 2003.

[13] V.Mezaris, I.Kompatsiaris, M.G.Strintzis: "An Ontology Approach to Object-Based Image Retrieval", IEEE International Conference on Image Processing, ICIP 2003, Barcelona, Spain, September 2003.

[14] Alex Pentland, Rosalind W. Picard, and Stanley Sclaroff. Photobook: Content-based manipulation of image databases. *International Journal of Computer Vision*, 18(3):233-254, June 1996.

[15] Dorin Comaniciu, Peter Meer, “Mean Shift: A Robust Approach Toward Feature Space Analysis”, IEEE Trans., Pattern Analysis and Machine Intelligence, vol.21, no.5, pp. 603-619, May 2002.

[16] Friedman, J. H. (1999). Stochastic gradient boosting. http://www-stat.stanford.edu/~jhf/ftp/stobst.ps