Content-Based Image Classification: A Non-Parametric Approach

Paulo M. Ferreira, Mário A.T. Figueiredo, Pedro M. Q. Aguiar

***Abstract* — The rise of the amount imagery on the Internet, as well as in multimedia systems, has motivated research work on visual information retrieval (VIR) systems and on automatic analysis of image databases.**
In this work, we develop a classification system that allows to recognize and recover the class of a query image based on its content. Such systems are called Content-Based Image Retrieval (CBIR).
CBIR systems describe each image (either the query or the ones in the database) by a set of features that are automatically extracted. Then, the feature vectors are fed into a classifier.
In this thesis, the processes of image feature selection and extraction uses descriptors and techniques such as Scale Invariant Feature Transform (SIFT), Bag-of-Words (BoW) and Spatial Histograms (SP).
For the classifier, we employ the Naive Bayes Nearest Neighbor (NBNN) algorithm, which belongs to the category of non-parametric classifiers. We also present a brief description of other classifiers used in image classification.
In addition, our work herein described tests and compares the image-to-class and image-to-image distances, in order to decide which leads to better performance.

***Index Terms* — Image classification, CBIR, feature extraction, NBNN**

I. Introduction

The number of digital images has grown astronomically, a consequence of the intense use of digital cameras, multimedia services and due to the storefront that the Internet turned into. Besides, in many areas, the use of image analysis has increased. Faced with this situation, the ability to classify images into semantic categories and objects (e.g. mountains, animals, humans, airplanes) is essential in order to manage and organize the collection of images on a database.
Most image search engines are supported on metadata (e.g. file name, author, file data and file size). Naturally, these systems fail to provide meaningfull results in terms of what is usually pretended from an image query. Besides, manually labeling large databases of images is hardly feasible or very expensive.
Content-based image retrieval [1] systems filter images based on their semantic content (e.g. objects, categories, relationships, and meanings), providing better indexing and giving more accurate results.
The objective of this paper is implementing a CBIR that classifies images by their object categories, through efficient approaches in image classification. Given a data set of images the goal is to classify them according to their object category (e.g. leopards, airplanes, sunflowers, faces, pizza). For this purpose, it is necessary to have image databases, which have become popular in computer vision, such as Caltech-101.
The scheme of image classification system consists of three modules, as Figure 1 demonstrates.
In the first module, the features of a group of images from the database and the features of the query image are extracted. This stage uses a set of descriptors to take out the features into vectors. Thus, two groups of feature vectors are created in the database and in the query. In this step, the system developed will test the classification system image with only a singular local descriptor (e.g. SIFT).
The second phase of the system has the purpose of comparing the query image features with the set of features of the database images. This module applies the classifiers and algorithms for image classification.
The methods of classification can be divided into two families: parametric classifiers (learning-based classifiers) and non-parametric classifiers.
Parametric methods require an intensive learning/training phase of the classifier (e.g. Support Vector Machines (SVM), Decision Trees (DT), Artificial Neural Network (ANN). The most usual image classifiers are supported on learning, especially SVM-based methods.

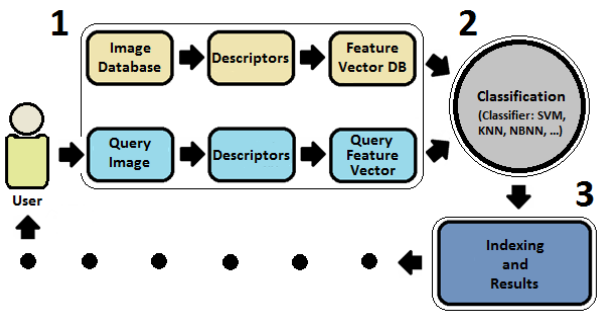


Figure 1: Scheme of image classification system.

Paulo M. Ferreira is a MSc student at Instituto Superior Técnico, Av Rovisco Pais, 1049-001 Lisbon, Portugal (e-mail: pmb.ferreira@gmail.com).
Mário A. T. Figueiredo is a Associate Professor, Department of Electrical and Computer Engineering, Instituto Superior Técnico (IST), Av Rovisco Pais, 1049-001 Lisbon, Portugal (e-mail: mtf@lx.it.pt).
Pedro M. Q. Aguiar is a Auxiliary Professor of Department of Electrical and Computer Engineering, Instituto Superior Técnico (IST), Av Rovisco Pais, 1049-001 Lisbon, Portugal (email: aguiar@isr.ist.utl.pt).

On the other hand, more common non-parametric methods are based on Nearest Neighbor (NN) distance estimation and classify an image by the class of its similar image on the database. Non-parametric classifiers use measures (e.g. Euclidean distance) to compute the similarity between the query image and an image on database – distance image-to-image.
In recent works**,** new approaches supported in non-parametric classifiers have obtained interesting results, such as Naive Bayes Nearest Neighbor (NBNN) [2]. The idea is to compute direct image-to-class distances without descriptor quantization, under the Naive-Bayes assumption. The advantages of NN-based classifiers are simplicity, efficiency and not requiring a learning phase. The NBNN classifier will be employ in the organism of the system and its performance using the image-to-class and image-to-image distance will be tested.

For comprehension, section II describes the problems to represent the visual content of an image.
Section III gives some background about feature extraction. Section IV present the datasets used in our experiments. Section V introduces the classifier employ in CBIR system, and how image-to-class distance differs from image-to-image distance. Section VI discusses about the implementation details and results are presented in section VII. Final the section VIII summarizes the conclusions of this work.

II. Variations in Image

When the aim is image retrieval, the challenge is to describe in fine a way the visual content. However there are many kinds of variations in an image that affect the classification, such as:

A. Illumination

Lighting causes important variations in the value of the intensity of the pixels. Illumination changes in image have a key influence on its appearance. Illumination and the occurrence of shadows sometimes change the appearance of objects which makes the recognition of an image difficult.

B. Scale and Size

An image can contain an object in front or far away. An object may appear alternative at different scales in the image. The scale and size of objects can considerably manipulate the similarity to other object of other classes.

C. Background Clutter

A complex background may result in confusion between objects in the foreground and background image. This problem Increases false-positive results in object retrieval.

D. Viewpoint and Pose

The position of the camera in relation to the object can change the appearance of an object in an image, which may lead to different results in the classification of the object.

E. Occlusion, Truncation and Articulation

The visibility of some part of the object may be damaged because of the proximity or overlapping of another object in the image or position of the same object. This causes large variations between samples of the same class and increases the intra-class variation.

F. Intra-Class Variability

Variation among instances between the objects belonging to the same class.

G. Inter-Class Variability

Confuses scenes of various categories that are quite similar.

III. Feature Extraction

In the field of image retrieval all content-based image systems require an appropriate representation of the input data – image. An image is formed by pixels, which may or may not represent features. A feature is defined as an interesting part of an image and is used as a starting point for computer vision algorithms.
An image can be represented globally or locally [3]. Global approach uses whole image to describe. While in local models, the selection of several regions or blocks of the image is utilized to characterize it. In this case, there are sparse and dense representations.
Sparse representation detects interest points or regions in the image. Then, this representation is extracted by a feature descriptor from each region. The proprieties of a good local feature are [4]:

- Must be highly distinctive - a good feature should allow for correct object identification with low probability of mismatch;
- Should be easy to extract;
- Invariance – a feature should be tolerant to image noise, changes in illumination, uniform scaling, rotation, and minor changes in viewing direction;
- Should be easy to match against a large database of local features.

Sparse representation requires an interest point detector to select the best points, edge segments or regions which characterize the image. Even if the original image is rescaled or modified by illumination and viewpoint changes, the detector must localize points that can be repeated. One of the most common interest point detectors used in image recognition is Difference-of-Gaussians (DoG). DoG [5] is an approximation of the Laplacian, and involves convolving the grayscale image with a Gaussian at several scales, creating a scale space pyramid of convolved images. The key points are detected by selecting positions in the image, which are stable across scales. Stable points are searched in these DoG images by determining local maxima, which appear at the same pixel across scales (Figure 2).

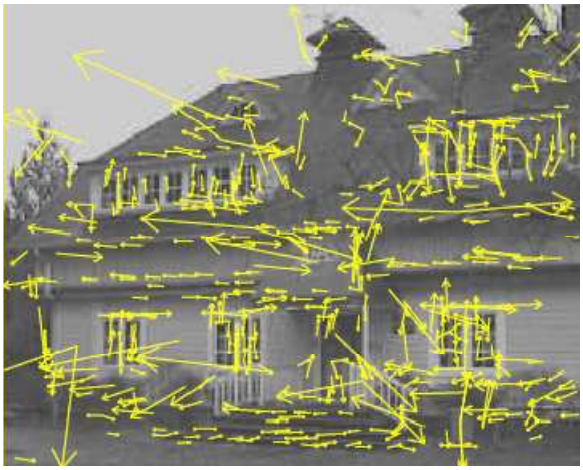
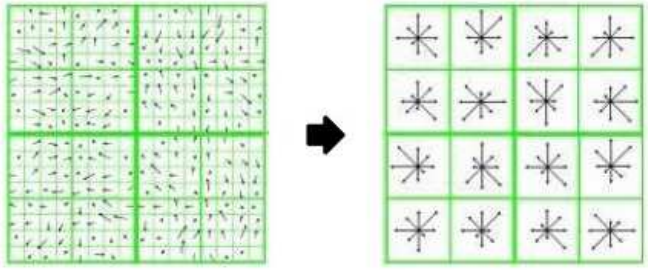


Figure 2: Example of Difference-of-Gaussian [5].

On the other hand, dense representation means that the features are not extracted at the key points, but the sense that each pixel contributes to the features description of the image on a dense grid.

Once features have been detected, the second step of the feature extraction process is characterizing the region around each interest point. For that, feature descriptors are used to compute these regions.
In computer vision hundreds of descriptors have been introduced. There are descriptors just for color features (e.g. color histogram, color moments, color correlogram), shape information (e.g. moments invariants, Fourier descriptors), and texture attributes (e.g. Tamura features, fractal model). However, for a good performance in object recognizing task you need descriptors which characterize features invariant to scale, orientation, affine distortion and partially invariant to illumination changes. Thus, in 1999, David Lowe created an algorithm to detect and describe features with these attributes. This descriptor was designed by Scale Invariant Feature Transform (SIFT).
SIFT [5] is decomposed in two stages. The first stage of the SIFT is finding the keypoint localization. For that, this descriptor uses DoG detector. The second step is keypoint orientation assignment and the keypoint descriptor computation (Figure 3). So for each interest point in an image there is a descriptor. A region around each keypoint is created and divided into orientation histograms on pixel neighborhoods (4 x 4). Each histogram contains 8 bins and each descriptor contains a *4 x 4* array of 16 histograms around the keypoint. This leads to a SIFT feature vector with 128 elements (4 x 4 x 8). Each image contains *n* keypoints, so an image is *n x 128* elements.



(a) (b)

Figure 3: SIFT descriptor computation. (a) The gradients of an image patch around a keypoint. These gradients are then accumulated over *4 x 4* sub-regions, as shown on the (b). The length of the arrow corresponds to the sum of the gradient magnitudes in that direction.

Image representation may include quantization technique - Bag-of-Word (BoW) [6]. BoW is based on regions and points of interest and corresponding features descriptions. BoW uses a clustering method to quantize the features descriptors. The bag-of-words, also know by bag-of-features, is a histogram of words which is applied to images by using a visual analogue of a word formed by vector quantization of visual features. Each interest point is indexed into a visual codebook or vocabulary. This vocabulary is formed by clustering the feature descriptors. So, the dataset of images is clustered into k representative clusters, where each cluster stands for a visual word. The resulting cluster can be more or less compact, thus representing the variability of similarity for individual features matches. For clustering, most often *k*-means is used.

IV. Data Set

The Caltech-101 [7] dataset is formed from 102 object categories and contains 9145 images. Each category includes between 40 and 800 images. Most images are medium resolution, about 300 x 300 pixels.
This dataset presents large inter-class variability and most images have little or no clutter. Objects are well aligned within each class and centered in each image. Most objects are presented in a stereotypical pose.

V. Classifier

Image classification is also an active area in the field of machine learning, in which it uses algorithms that map sets of input, attributes or variables – a feature space $\mathbf{X}$ - to set of labeled classes $\mathbf{Y}$. These algorithms are known as classifiers. Basically what a classifier does is assign a pre-defined class label to a sample. There are two main stages in a classification system: training and testing stage.
Training is the process of defining criteria by which features are recognized. In this process the classifier learns its own classification rules from a training set. In the training process, images are captured and stored in a database. Then there is the process of feature extraction. As previously stated, an image is represented by a set of descriptors that structure the feature

vectors. These feature vectors are considered input variables and are introduced in a learning component. The outputs are labels associated with classes (e.g. airplanes, faces, flowers).
In the learning component you have the discriminative and the generative models. The first model maps input variables directly to output variables in order to perform classification.
The generative field models the distribution of features and learning is based on the likelihood of the data.
In the testing stage, the feature vectors of test image works as input. A classifier decides on the bases of learning model, with its own classification rules, as to which class that feature vector belongs.
In literature, there are several approaches for classifiers, which can be characterized by two types of families: unsupervised learning (UL) and supervised learning (SL).
In UL, the feature space of the entire dataset is clustered on the basis of some similarity criteria, forming a set of clusters. Each cluster belongs to a specify class. The main problem of classification with UL is how to take a decision between the feature vectors provided. Another problem is the selection of an algorithm that will cluster the vectors, since different clustering algorithms lead to different clusters.
On another hand, the SL involves a set of training data and category labels. The classifier is projected by utilizing this prior known information. In this case, the knowledge of the number of classes and their location in the feature space is the prior information. The problem of this learning is that it takes some time to develop a classifier.
There are many techniques to design a classifier using supervised learning, which are based on two different categories: parametric and non-parametric.

A. Parametric methods

These methods based on statistical parameters that assume a normal distribution and require an exhaustive learning or training phase of the classifier parameters. Examples of parametric classifiers are: SVM, DT and ANN.

B. Non-parametric methods

These methods base their classification decision directly on the data, and do not require an intensive learning or training phase of the classifier. Examples of non-parametric classifiers are: Naive Bayes, KNN and NBNN.

Recently, in 2008, Oren Boiman et al. [2] proposed a trivial classifier based on k-Nearest Neighbor and on Naive Bayes assumption. This algorithm was designed by Naive Bayes Nearest Neighbor (NBNN) and regained the status of non-parametric classifiers due to its good performance in datasets with large intra-class variances (e.g. Caltech-101).
The Naive Bayes assumption make the conditional independence of the features given the class membership. A NN search algorithm finds the closest descriptors of each class associating with its distance to descriptors of the query image.
A class with the lowest total distances is chosen the classified category for the query image.

So, the NBNN algorithm consists of the following:

1. Compute descriptors $d_1, \dots d_n$ of the query image Q.
2. $\forall d_i \forall C$ compute the nearest neighbor of d_i in C: $NN_c(d_i)$.
3. The class of the query image is defined by:

$$\hat{C} = arg\,\min_C \sum_{i=1}^{n} \|d_i - NN_C(d_i)\|^2 \qquad (1)$$

NBNN is extremely simple, efficient and requires no learning or training phase. It uses the term ‘labeled images’ instead of ‘training images’, i.e. the classifier is fixed for all database image sets.

The NBNN classifier can assume two different ways regarding the Neighbor Nearest distance: image-to-class and image-to-image.
In this context, we test the two distances in order to choose which one will be used in CBIR implementation. The measure used in computation is Euclidean distance. In order to improve the runtime and the computational complexity of classification we use the *kd*-tree [12] algorithm.
Next two subsections will explain the algorithm NBNN with the two distances.

C. Image-to-class distance

In the training phase all training images I ($I \in class\ C$) of the database compute and add descriptors $d_1, d_2, \dots, d_n$ to a *kd*-tree T_C. In the test stage, the algorithm computes descriptors $d_1, d_2, \dots, d_n$ of the query image. Then, $\forall d_i\ \forall T_C$ compute $NN_C(d_i)$ - the nearest neighbor of d_i to class C.
Finally, the NBNN classifies the class of the query image by the step 3 of the algorithm, the equation 1

D. Image-to-image distance

In the labeled step all training images I of the database compute and add descriptors $d_1, d_2, \dots, d_n$ to a *kd*-tree T_I.
In the test level, the algorithm computes descriptors $d_1, d_2, \dots, d_n$ of the query image. Then, $\forall d_i\ \forall T_I$ compute $NN_I(d_i)$ - the nearest neighbor of d_i to image I.
Finally, the NBNN classifies the class of the query image by the class of the nearest neighbor image of database. For that, it uses:

$$\hat{C} = \arg\min_I \sum_{i=1}^{n} \|d_i - NN_I(d_i)\|^2 \qquad (2)$$

VI. Implementation

The CBIR system implementation consists, as previously cited, of three modules:

- Feature extraction of database and query image;
- Compute the distances between features;
- Results classification.

Regarding the first module, feature extraction, the system randomly selects from Caltech-101 database 15 images per class. The features are extracted in these images, through the dense SIFT descriptor. The dense representation is used instead of sparse representation due to its best performance. The SIFT descriptors are computed over the gray scale image and on a regular grid with spacing *p* pixels (*p=3*). At each grid point the descriptors are computed over. A spatial bin with size *s* (*s=16*) covers these p pixels. This bin size is related to the SIFT keypoint scale.
This implementation does not use BoW representation, because it is a quantization technique which damages non-parametric classifier [2]. To evaluate the image-to-class and image-to-distance in NBNN algorithm it was necessary implement two different CBIR systems.

A. Implementation Image-to-Class

The set of the descriptors extracted from the 15 images per class composes the data which represent the class. This means that all image descriptors, each one with a different dimension $(128 \times n_i)$, are merged in a single matrix, with a dimension $128 \times (n_1 + n_2 + \cdots n_{15})$.
The next step is to index the classes and to create a *kd*-tree per class in order to improve runtime and computational complexity. Figure 4 illustrates the implementation of CBIR system using image-to-class distance.
In the interactive part of these systems, the user introduces a query image. This image suffers the transformation of the dense SIFT descriptor with the same parameters applied in image database, in order to obtain the query image features.
Therefore, the distances between features are compute using the approximation of the nearest neighbor (*kd*-tree) and the equation 1 – image-to-class distance.
The results of the classification are indexed by class approaching to the query image. Then, the images of the nearest class to the query image are displayed in an interface.

B. Implementation Image-to-Image

In the implementation image-to-image the classes have 15 representations based on the descriptors extracted of each image. This means that each class is represented by 15 matrixes with different dimensions $(128 \times n_i)$.
The next step is to index the image classes and to create a *kd*-tree per each image descriptors. Figure 5 shows the implementation of CBIR system using image-to-image distance.
In this system, the query image suffers the same process previously described. The distances between features are compute using the approximation of the nearest neighbor (*kd*-tree) and the equation 2 – image-to-image distance.
Then, the results of the classification are indexed by approaching of database image class.

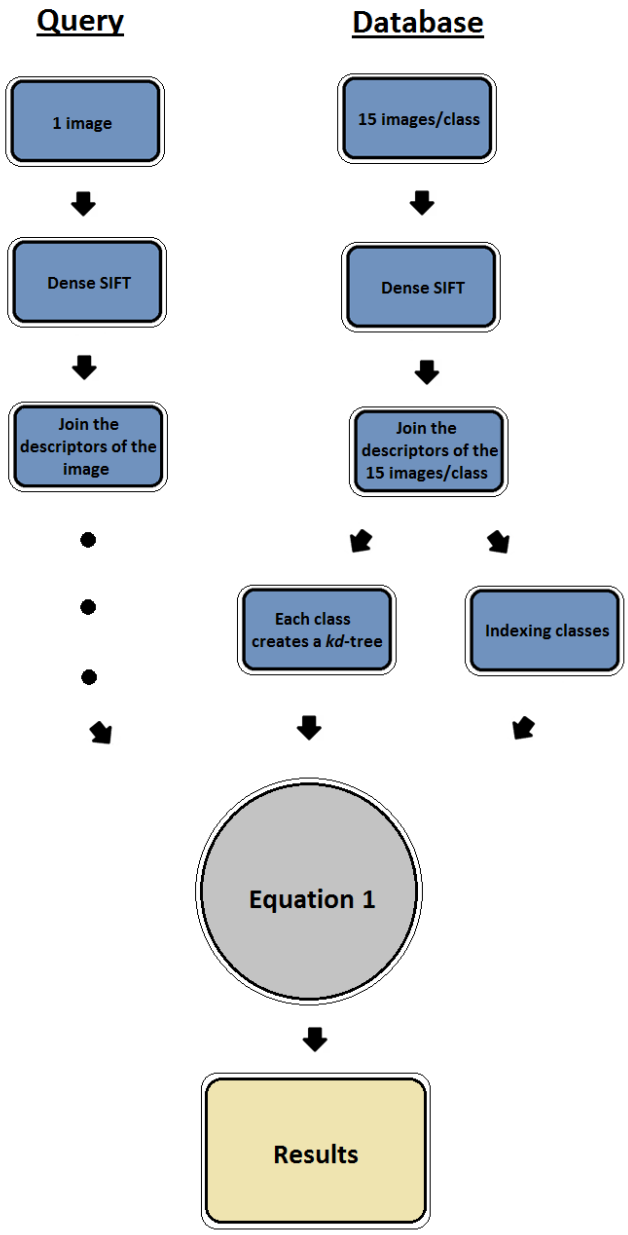


Figure 4: CBIR system implementation using image-to-class distance.

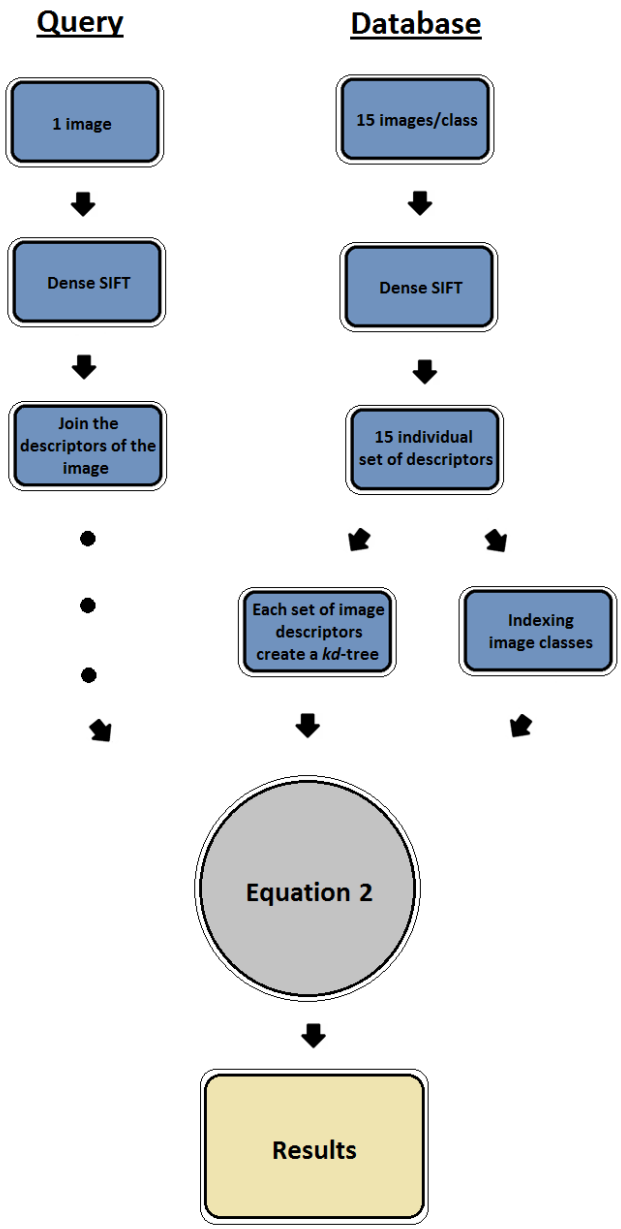


Figure 5: CBIR system implementation using image-to-image distance.

VII. Results

The SIFT descriptor and the *kd*-tree used in this experiment was developed by VLFeat [8] open source library that implements popular computer vision algorithms. The simulations are realized in MATLAB [9].

To test the accuracy of these systems, we use 20 query images per class – total of 2040 images. If in these images per category, x images - $\{x \in \mathbb{N} \mid 0 < x \leq 20\}$ - match to the true class, the accuracy of the class is $x/20$.

For the CBIR using the image-to-class distance, Figure 6 shows the distribution of number of classes for different levels of accuracy. Although many of the classes present a good performance (17 classes very well and 21 well classified), certain classes had a low classification (21 unsatisfactorily and 17 poorly categorized), which did reduce the overall performance. The average time spent to classify a query image is 2.2 seconds per class.

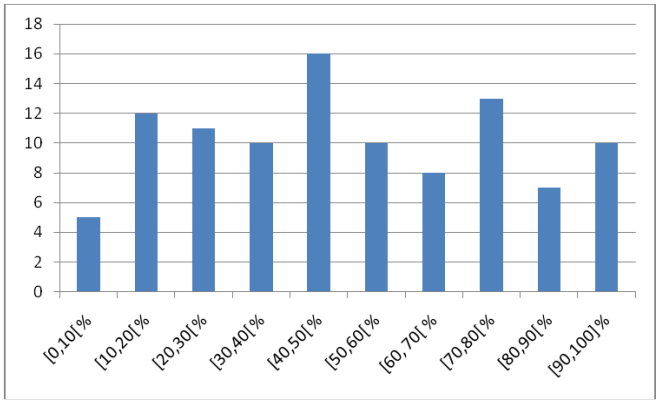


Figure 6: Number of classes for different levels of accuracy for image-to-class distance.

For the CBIR using the image-to-image distance, Figure 7 shows the distribution of number of classes for different levels of accuracy. In this case, only 14 classes very well classified, and 12 well categorized. Certain classes had a low classification - 27 unsatisfactorily and 27 poorly categorized.

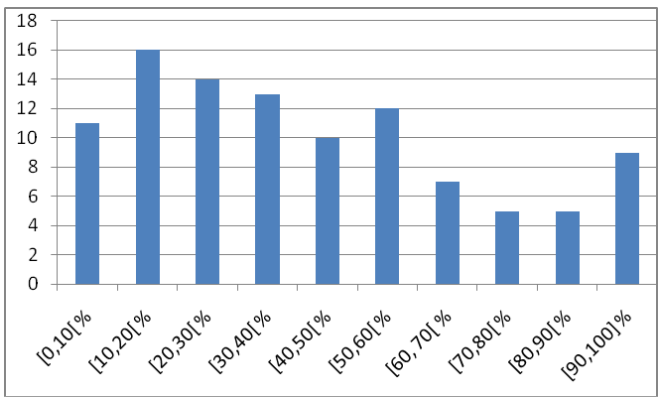


Figure 7: Number of classes for different levels of accuracy for image-to-image distance.

The global accuracy of each system is shown in Table 1, where the CBIR system using the distance image-to-class has 48,5% of accuracy and the CBIR system using the image-to-image distance presents 39,7% of accuracy.

The results of this testing showed a difference of 8.8% (different accuracy of distances of the) favorable to distance-to-class.
Oren Boiman et al. [2] showed a 17% difference between the distances, favorable to distance image-to-class. That research uses the database Caltech-101 with 30 images labeled per class and the descriptor SIFT in five different scales with image. The distance used was the Kullback Leibler[1] distance.
In relation to the experience in [2], this test used 15 labeled images per class and only applied the SIFT descriptor on one scale. The measure used was Euclidean distance.

Table 1: Accuracy of CBIR using the image-to-class and image-to-image distances.

Distance	Accuracy
Image-to-class	48,5%
Image-to-image	39,7%

Phil Huynh [11] showed a difference of 0.2% favorable image to the distance-to-class in the Central Park[2] database, using the Euclidean distance.
The difference of this result to the result in [11] must be the database used. While the Central Park database has no inter-class variability, the Caltech-101 presents this variability since it contains objects of different classes.
The results decide that image-to-class distance leads to better performance in the CBIR system.

VIII. Conclusion

In the NBNN classifier, this work tested two different ways on the nearest neighbor measures: image-to-class and image-to-image distances. By comparing the two distances, the conclusion is that the image-to-class distance showed a better performance with an 8,8% difference to the image-to-image distance. Although it is applied in systems with parametric classifiers, the image-to-image distance limits the ability of classification of non-parametric methods.

Overall, the result of classification (2040 images tested – 20 per category) was 48.5%. This performance is satisfactory as this work only applied one single descriptor to extract the image features. Another reason is that the database Caltech-101 used contains a large number of categories – 102 classes. This fact caused some inter-class variability in feature comparison, which makes a poor classification in some classes. On the other hand, the problems of object retrieval like pose, viewpoint or articulation of the image content do not aim the process of classification.

[1] is a non-symmetric measure of the difference between two probability distributions

[2] stock leaf images in 143 species

References

[1] Long, F. H., H. J. Zhang and D. D. Feng, Fundamentals of content-based image retrieval, *in Multimedia Information Retrieval and Management - Technological Fundamentals and Applications*, Springer-Verlag, pp.1-26, New York, 2003.

[2] Boiman, O., Shechtman, E. and Irani, M., In defense of Nearest-Neighbor based image classification, in *IEEE Conference on Computer Vision and Pattern Recognition*, CVPR, pp.1-8, 2008.

[3] Bosch, A., Image classificaion for a large number of object categories, PhD thesis, in *University of Girona*, 2007.

[4] Tuytelaars, T., Mikolajczyk K., Local Invariant Feature Detectors: A Survey, foundations and trends in computer graphics and vision, 2008.

[5] Lowe, D. G., Distinctive image features from scale invariant features, *International Journal of Computer Vision*, Vol. 60, No. 2, pp. 91-110, 2004.

[6] Sivic, J. and Zisserman, A., Video google: A text retrieval approach to object matching in videos, in *International Conference on Computer Vision*, 2003.

[7] Caltech-101, available at: http://www.vision.caltech.edu/Image_Datasets/Caltech101

[8] VLFeat open source library, available at: http://www.vlfeat.org/

[9] Image Processing Toolbox for use with MATLAB User's Guide, The MathWorks, 2008.

[10] Huynh, P., An evalution of nearest neighbor images-to-classes versus nearest neighbor images-to-images.

[11] Duda, R.O., Hart, P.E., and Stork, D.G., *Pattern Classification*, 2nd Edition, John Wiley, New York, 2001.

[12] Deng, K., Omega: On-line memory-based general purpose system classifier, Chapter 5: Efficient Memory Information Retrieval, 1998.