UNIVERSIDAD POLITÉCNICA SALESIANA ECUADOR
INGENIUS
Revista de Ciencia y Tecnología

Combination of HSV and RGB histograms and MPEG-7 Descriptor: Preliminar Results and Future Work

Javier Poveda Figueroa[1,*] y Vladímir Robles Bykbaev[2]

Abstract

In this paper we present the partial results obtained with a new approach to combine HSV and RGB histograms and MPEG-7 CLD descriptor. The combination was conducted using Borda Voting-Schemes in three databases: Wang, ZuBud an UW. Despite the poor initial classification performed with CLD descriptor, our proposal has achieved good results for the Wang database (82.66 %), outperforming the precision of HSV (72.33 %) and RGB histograms (79.33 %), and CLD descriptor (53 %). In the other databases (ZuBud and UW) the combination approach was unable to perform a significant improvement.

Keywords: HSV histogram, RGB histogram, Colour layout descriptor, Borda voting schemes, KNN.

Resumen

En este artículo presentamos los resultados parciales obtenidos con un nuevo enfoque para combinar los histogramas RGB, HSV y el descriptor CLD del estándar MPEG-7. La combinación fue realizada usando esquemas de votación Borda en tres bases de datos: Wang, UW y ZuBud. A pesar de la baja clasificación inicial realizada con el descriptor CLD, nuestra propuesta ha alcanzado mejores resultados para la base de datos Wang (82.66 %) que los histogramas HSV (72.33 %), RGB (79.33 %) y el descriptor CLD (53 %). En las otras bases de datos (ZuBud y UW) la combinación no generó mejoras significativas.

Palabras clave: histograma HSV, histograma RGB, colour layout descriptor, esquemas de votación Borda, KNN.

[1,*] *Colaborador del Laboratorio de Investigación en Sistemas Informáticos e Inteligencia Artificial, Carrera de Ingeniería de Sistemas, Universidad Politécnica Salesiana, sede Cuenca. Autor para correspondencia ✉: jpoveda@est.ups.edu.ec*
[2] *Máster en Inteligencia Artificial, Reconocimiento de Formas e Imagen Digital, Ingeniero en Sistemas, Estudiante de Doctorado en Informática, Universidad Politécnica de Valencia, Encargado del Laboratorio de Investigación en Sistemas Informáticos e Inteligencia Artificial, Universidad Politécnica Salesiana, sede Cuenca.*

Recibido: 30 - Octubre - 2012, *Aprobado tras revisión: 21 - Noviembre - 2012*

Forma sugerida de citación: Poveda, J. y Robles, V. (2012). “Combination of HSV and RGB histograms and MPEG-7 Descriptor: Preliminar Results and Future Work”. *INGENIUS*. N.°8, (Julio/Diciembre). pp. 3-10. ISSN: 1390-650X

1. Introduction

At present several descriptors exist to extract relevant features from images. One of the most used descriptors is the color histogram, because it is simple to implement and understand. However, the color histograms have limitations when they are used to classify images with a similar color distribution but different content.Other descriptors are more robust to classify image using textures, color distribution and shape descriptors.

Several approaches have been presented to perform image retrieval through the combination of different descriptors. Some proposals use the machine-learning approach (Support Vector Machines and Neuro-Fuzzy Networks) [1], while others directly combine the features set [2],[3] to show the improvement achieved with combination.

In this paper we introduce the results of a new approach presented in [4], to combine HSV and RGB histograms and MPEG-7 CLD descriptor. In the previous paper we explained the color histograms RGB and HSV and the MPEG-7 color Layout descriptor. Then, we applied the supervised learning algorithm K-Nearest Neighbors to get the votes of each image, so that we can classify them and use the votes with the Borda Voting Scheme. Finally, we use the Borda Voting Scheme to combine the votes to classify the data we are working with.

The results achieved with our approach have obtained good results for the Wang database, improving the precision to 82.66 %. With the other two databases that we used, the combination has not significantly improved the precision.

The rest of the paper is organized as follows. In Section 2 we review the databases and tools used, and the process followed to extract the features. The technical details of the approach's implementation are reviewed in Section 3. In Section 4 we present the results obtained. Finally, discussion and future work are described in Section 5.

2. Design phase of the experiment

2.1 Corpus selection

The process selected to choose the corpus images is explained in the next steps. UW, Wang and ZuBud corpuses were chosen because of their features. Each of these corpuses was made by a group of experts in the area of computer vision to apply machine-learning algorithms to them so that they can classify the classes of images that the image corpus has.

The University of Washington made the UW database and it contains 1109 images. The images were chosen from 18 vacation places. As [3] points out, the classification process is made with annotations, so that the image could be classified in a category based on its annotations.

For example, given a query image of a category that is going to be compared with all the categories in order to determine to which category they belong to. A retrieval task could be looking for the image of the same building, or the image of any place of vacation. Figure 1 illustrates the UW database with its annotations.

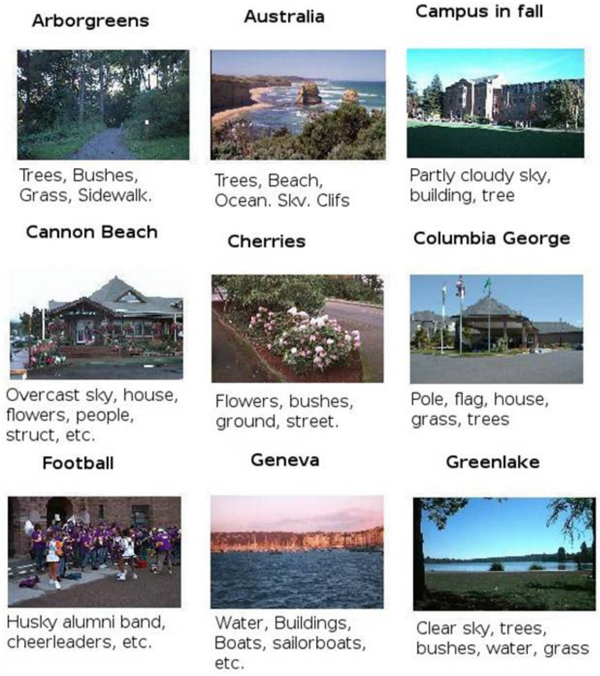
Arborgreens
Australia
Campus in fall
Trees, Bushes, Grass, Sidewalk.
Trees, Beach, Ocean. Skv. Clifs
Partly cloudy sky, building, tree
Cannon Beach
Cherries
Columbia George
Overcast sky, house, flowers, people, struct, etc.
Flowers, bushes, ground, street.
Pole, flag, house, grass, trees
Football
Geneva
Greenlake
Husky alumni band, cheerleaders, etc.
Water, Buildings, Boats, sailorboats, etc.
Clear sky, trees, bushes, water, grass

Figure 1. Description of the UW Database, which can be found at: http://www.cs.washington.edu/research/imagedatabase/groundtruth.

The Wang database is a collection of 1000 images from theCorel database, which are divided in 10 categories of 100 images each one. The Content-Based Image Retrieval process for this database searches from a query database comparing all the images of the database to seek where it belongs to so that it can be classified in any category. Figure 2 shows this database with its categories [3].

The ZuBud database was created at the Swiss Federal Institute of Technology in Zurich and it contains 1005 images of 201 buildings. Each building is represented in this database by 5 images; each one was taken by different viewpoints, under different conditions and

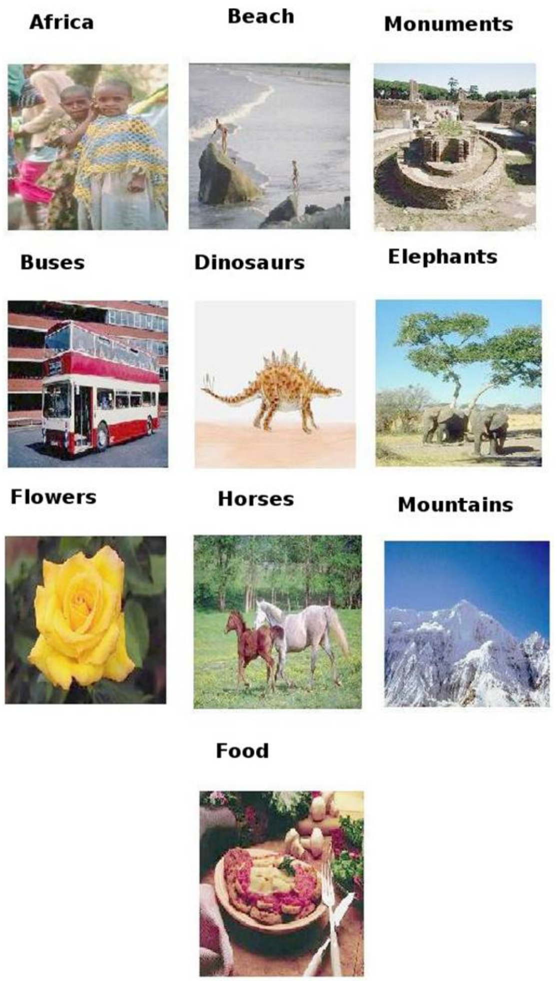

Figure 2. Images of the Wang database, which can be found at: wang.ist.psu.edu/docs/related

with two different cameras. The Content Based Image retrieval process is the same process that was explained with the UW and Wang databases. The difference being that the query image, which is used must recover an image from the same building. Figure 3 shows this database with its categories [3].

2.2 Tools which were used to do the experiment

The tools that were used to do the experiment are: OpenCV 2.4.0 for UNIX, the C++ version 4.6 for UNIX, Visual Studio 2008 Express Edition, Java 1.6.1 and Net Beans 5.5. OpenCV2.4.0 was used to get the image features from the color spaces RGB, HSV and the MPEG-7 standard Color Layout Descriptor. Also, we used the OpenCV tool for transforming RGB to HSV color space and to the YCbCr Color space. After that, the GUI Visual Studio Standard Edition was used to program the K-Nearest Neighbors Algorithm. Then, the GUI NetBeans 5.5 with the programming language Java 1.6.1 to program the Borda Voting Schemes. Finally, the results of the experiment were saved on a Microsoft Word 2007 text document to analyze them and get the results.

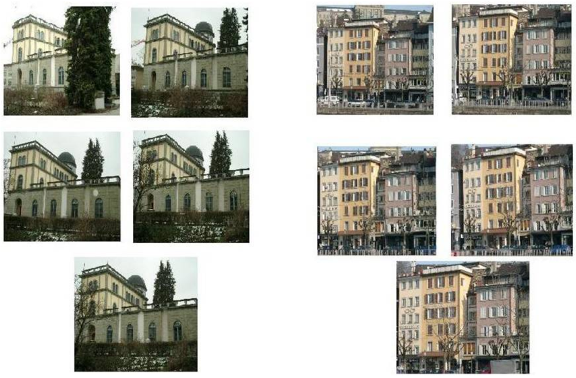

Figure 3. Images of ZuBud database, which can be found at: http://www.vision.ee.ethz.ch/showroom/zubud/

2.3 Process followed to do the experiment

The process of the experiment was divided in two parts. The first part consists in the extraction of the characteristics of the images, the quantization and shuffle of them. The second part is the application of the machine-learning algorithm K-Nearest Neighbors and the Borda Voting Schemes. In this subsection we willexplain in more detail the design phase of the experiment.

The process goes as follows:

Features Extraction: This process generates the color RGB and HSV histograms and CLD descriptor. To obtain the bins of the HSV and the CLD it is necessary to convert the RGB color space of the images to the color spaces HSV and YCbCr. This process is done because the RGB histogram is the most basic histogram that can be converted to any color space [5].

To covert the RGB to the HSV color space we have to use the following formulas [5]:

$$H = \cos^{-1} \frac{0{,}5\left[(R-G)+(R-B)\right]}{\sqrt{(R-G)^2+(R-G)(R-B)}} \tag{1}$$

Equation 1. Conversion from RGB to Hue.

$$S = 1 - \frac{3}{R+G+B}\left[min\left(R,G,B\right)\right] \tag{2}$$

Equation 2. Conversion from RGB to Saturation.

$$V = \frac{1}{3}(R + G + B) \qquad (3)$$

Equation 3. Conversion from RGB to Value.

To convert RGB to YCbCr we use the following formulas [6]:

$$Y = 0{,}299R + 0{,}587G + 0{,}1148B \qquad (4)$$

Equation 4. Conversion from RGB to Value.

$$C_b = -0{,}127R - 0{,}339G + 0{,}5211B + 128 \qquad (5)$$

Equation 5. Conversion of RGB to C_b.

$$C_r = 0{,}511R - 0{,}428G - 0{,}083B + 128 \qquad (6)$$

Equation 6. Conversion RGB to C_r.

Quantization: The quantization process is used to reduce the dimensionality of the color features [4] and improve the performance of the system.The processing time needed to analyze them is limited because of the computer memory and the time to process them must be as shorter as possible. For this reason, the color data bins of each image are quantized to facilitate the elements that were explained.

The color data bins are reduced the next form. RGB is reduced to 32 bins for each color channel, but for the experiment they were reduced to 256 bins for all the channels, instead of reducing them to 32 bins for each color channel; HSV is reduced to 16 bins in H, 4 bins in S, and 4 bins in V; finally, the CLD color descriptor produces 12 bins because its structure was defined previously.

Shuffle: The objective of this process is to mix the image set of features as a necessary stage for the experiment, because the system must not learn the image features in the same order each time [4].

Test and Train for experimentation: The set of color features is divided in two subtests, which are going to be used in the experiment: train and test. The train subset has 70 % of the features that are going to be used as a base to compare the features to a new one, and the test subset of images that is going to be compared with the training set to determine which category the images belong to [7].

K-Nearest Neighbors Algorithm: The K-Nearest Neighbor Algorithm was chosen to do the experiment because of its characteristics. The K-Nearest Neighbor Algorithm uses two sets of data, one as a standard and the other to compare each group of data with the standard set of data. Another important characteristic of this algorithm is the structure, because it processes the group of data without any loss of it [7].

Because of the nature of the KNN algorithm, which is a machine-learning supervised algorithm, it uses two subsets of data that are the train and the test sets. As it was explained previously in the Test and Train subarea of experimentation we use the train subset as a base to compare features and the test subset to compare the images with the train subset so that we can determine which image belongs to each category of the set of images.

In this case, the train and test set are used for comparing each color feature of the test set with each one of the training set, in this method the next equations [8] are applied, Equation 7 represents the Euclidean distance which is used for measuring the distance on the RGB and HSV histogram, and Equation 8 measures the distance of the CLD color descriptor:

$$Euclidean = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \qquad (7)$$

Equation 7. Euclidean distance formula which determines the distance between two points. In this case x_i represents the value of the test subset, while the y_i represents the value of the train subset.

$$Dcld = \sqrt{\sum_i w_i^Y \left(Y_i - Y_i'\right)^2} + \sqrt{\sum_i w_i^{cb} \left(Cb_i - Cb_i'\right)^2} + \ldots$$
$$\ldots + \sqrt{\sum_i w_i^{Cr} \left(Cr_i - Cr_i'\right)^2} \qquad (8)$$

Equation 8. Formula which compares the distance between the color values of YCbCr.

Where:

- Y_i represents the value of Y of the test image and represents the value of Y of the train image.
- Cb_i represents the value of Cb of the test image and $Cb_i^{'}$ represents the value of Cb of the train image.
- Cr_i represents the value of Cr of the test image and $Cr_i^{'}$ represents the value of Cr of the train image.
- w_i^Y represents the weights of Cb, and the Equation 9 represents the weight of Cb where j and k are the positions of the train and test images respectively :

$$w_i^Y = \frac{0{,}4}{(j+k)^2} \qquad (9)$$

Equation 9. Weight of the Cb value.

- w_i^{Cb} represents the weights of Y, and the equation 10 represents the weight of Y where j and k are the positions of the train and test images respectively :

$$w_i^Y = \frac{0{,}2}{\left(j+k\right)^2} \tag{10}$$

Equation 10. Weight of the Y value.

- w_i^{Cr} represents the weights of Cr, and the Equation 11 represents the weight of Cr where j and k are the positions of the train and test images respectively :

$$w_i^{Cr} = \frac{0{,}4}{\left(j+k\right)^2} \tag{11}$$

Equation 11. Weight of the Cr value.

Borda Voting Schemes: The Borda Voting Schemes are an ensemble technique developed in France in the 18h century by the french Mathematician Charles Borda. This method was developed because the voters made very bad decisions in the voting process. This was because the voters chose the candidate not for its merits, but for his popularity. Because of this reason, Borda developed a method, which helped to choose the correct candidate by his merits [9].

The process of the Borda Voting Scheme goes as follows. The Borda Voting scheme consists of a group of candidates $X = \{1, 2, 3, \ldots, x_n\}$ where $n \geq 3^2$ and m voters where $m \geq 3$ [7]. Defined in a formal ways as [9] explains: "*The relationship P^k represent the preference of the voter k, $k = \{1, 2, 3, \ldots, m\}$, over the collection of n alternatives of X*". The relationship can be represented by the matrix [9]:

$$P^k = \begin{bmatrix} r_{11}^k & r_{12}^k & \cdots & r_{1n}^k \\ r_{21}^k & r_{22}^k & \cdots & r_{2n}^k \\ \vdots & \vdots & \ddots & \vdots \\ r_{n1}^k & r_{n2}^k & \cdots & r_{nn}^k \end{bmatrix}$$

Where

$$r_{ij}^k = \begin{cases} 1 & (if\left[x_i P x_j\right]) \\ 0 & (in\left[other\right]\{case\}) \end{cases}$$

It can be defined by the next Equation [9]:

$$r_k\left(x_i\right) = \sum_{j=1}^{n} r_{ij}^k \tag{12}$$

Equation 12. The Borda Voting Schemes formula. If you want to know how it resolves a problem please go to [4].

3. Approach implementation

This section will explain the process done to make the experiment. First, the processing of each corpus will be explained. Second, how the KNN and Borda Voting schemes algorithms were programmed.

3.1 Preprocessing of each corpus

As it was explained in section one, we used 3 image corpuses UW, Wang and ZuBud, each one of these corpuses are divided in categories which are: 23 for UW, 10 for Wang and 201 for ZuBud. Every image of the corpuses was processed with OpenCv 2.4.0. However, some categories of the UW database could not be processed and others must be converted to the jpeg image format in order to process them and extract information.To do this the ImageMagick program was used. The categories which could not be processed were: Barcelona2 and those which were converted are: Columbia George and Green lake and those which ended in a different format than JPEG. Columbia George in its original form could not be processed, so it was necessary to reconvert it to the jpeg format again.

3.2 Programming of the modules of the system

The module that extracts the characteristics of the RGB histogram was programmed in OpenCV 2.4.0 in Ubuntu Hardy and it had the following characteristics. First, the features of the images are extracted. Second, the image features are quantized in 256 bins. Finally, the quantized bins are written on a word processor, so that they can be used by the K-Nearest Neighbors Algorithm.

The HSV module is like the RGB module, but it has two differences. The first difference is the conversion of the RGB color space to the HSV color space. The second difference is the way the HSV histogram was quantized by the OpenCv standard for HSV, the value for H is 180 and for S and V is 256. The results of this module are written in a word processor so that they can be used by the K-Nearest Neighbors.

The module of the Color Layout Descriptor was programmed in C++ with OpenCV in Ubuntu Hardy and it follows the next process. Firstly, it extracts the color features on the RGB color space. Secondly, it transforms the color features of RGB in the color space YCbCr. After that, it divides the image in a matrix of 8x8. Then, the Discrete Cosine Transformation is applied to reduce the low-frequency noises. Finally, the coefficients of the CLD are written in a word processor so that they can be used by the K-Nearest Neighbors.

The K-Nearest Neighbors module was programmed in Visual Studio 2008 express edition. It follows the next steps. First, a model for extracting the color features of the color descriptors and the Color Layout Descriptor is defined. For this the word processor documents which were explained on the previous section were used. Second, those color features are shuffled in order to reduce error from the experiment. After that, the features are separated in two subsections: the train subsection that has 70 % of the color features and the test subsection that has the 30 % of the color features. After that, the elements of the test subsection are compared with the elements of the train subsection. Finally, the results are written in a word processor to be used by the Borda Voting Schemes.

The Borda Voting Scheme module was programmed in java and it follows the next steps. First, it reads the word processor which was written previously. Second, it gets the results of the KNN algorithm and uses them to form the matrices. Finally, it sums the matrices and gets the results. On Figure 4 it is shown the process stated above:

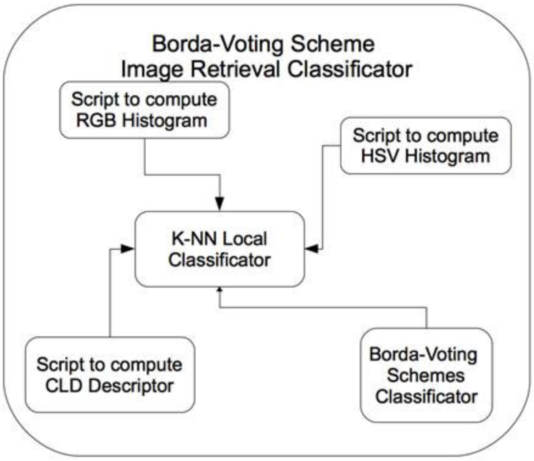
Borda-Voting Scheme
Image Retrieval Classificator
Script to compute RGB Histogram
Script to compute HSV Histogram
K-NN Local Classificator
Script to compute CLD Descriptor
Borda-Voting Schemes Classificator

Figure 4. Diagram that explains the relationships between the system modules.

4. Experimental results

This section will explain the results obtained from the experiment. The section is divided in three parts: the first one explains the variables used on this experiment, the second part explain the precision of the system, and finally, the third part explains the recall of the system.

4.1 Variables to measure the system

The variables used are precision and recall. Precision represents the number of recovered images, which are relevant for the search. This variable is represented in the next equation [10]:

$$P = \frac{a}{a+b} \tag{13}$$

Equation 13. This equation represents the precision of the system where a represents successful recovered images and b represents the irrelevant images.

Recall represents the number of relevant images of a database, which are recovered in response of a search. This is represented on the next quation [10].

$$C = \frac{a}{a+c} \tag{14}$$

Equation 14. This equation represents the recall of the system where a represents successful recovered images and c represents unsuccessful recovered images.

4.2 System precision and recall

In this subsection the system precision on recovering images will be explained. For this reason the subsection will explain the precision for each corpus UW, Wang and ZuBud.

The UW database had a precision of 66.42 %. This is because of the variances between the images and the distribution of color of them is not very similar. On Figure 5 the correct and incorrect answers can be appreciated.

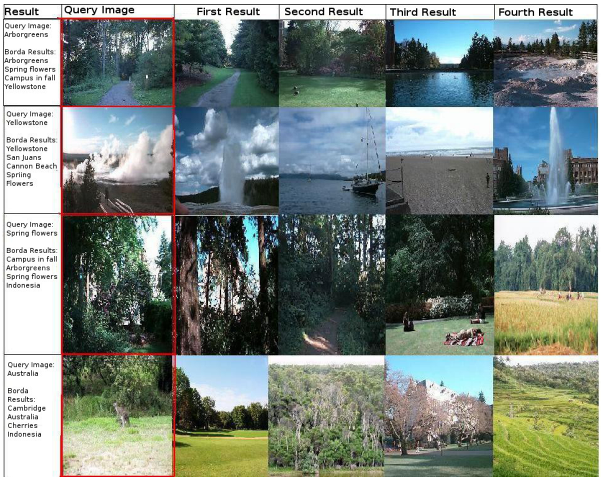
Result
Query Image
First Result
Second Result
Third Result
Fourth Result
Query Image: Arborgreens
Borda Results: Arborgreens Spring flowers Campus in fall Yellowstone
Query Image: Yellowstone
Borda Results: Yellowstone San Juans Cannon Beach Spriing Flowers
Query Image: Spring flowers
Borda Results: Campus in fall Arborgreens Spring flowers Indonesia
Query Image: Australia
Borda Results: Cambridge Australia Cherries Indonesia

Figure 5. Images of the UW database where the Query Images are highlighted in red and the other four images are the top four results for this query image.

The Wang database had a precision of 82.66 %. This is because the color distribution is very similar on each image of the categories. On Figure 6 the correct and incorrect answers can be appreciated.

The ZuBud database has a precision of 50 %. This result is because the Color Layout Descriptor has more

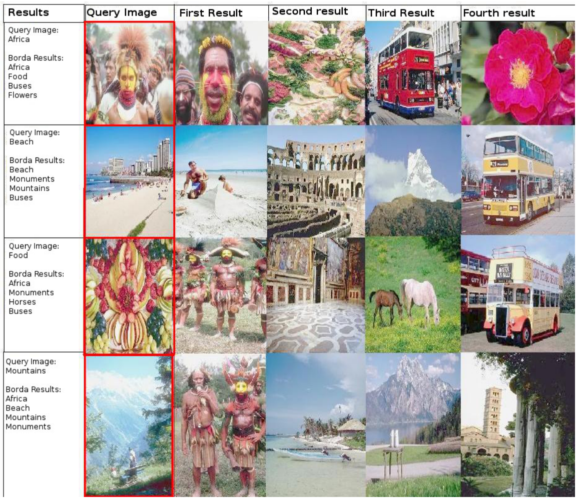

Figure 6. Images of the Wang database where the Query Images are highlighted in red and the other four images are the top four results for this query image.

precision with images that have better color distribution and less border structure. On Figure 7 the correct and incorrect answers can be appreciated.

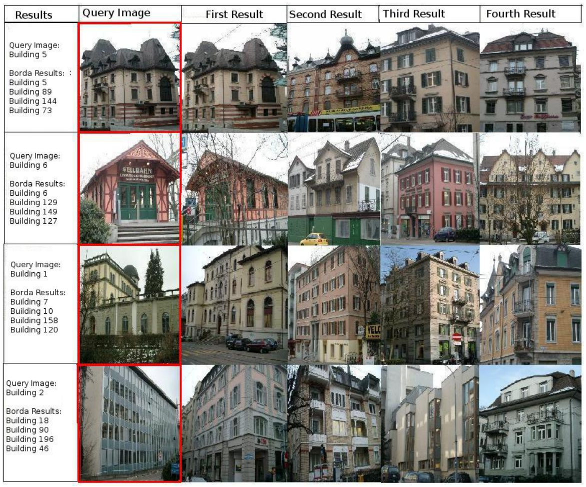

Figure 7. Images of the ZuBuddatabase where the Query Images are highlighted in red and the other four images are the top four results for this query image.

The recall of the system was 100 % because of the characteristics of the K-Nearest Neighbors.

4.3 Comparing Precision with other systems

4.3.1 General Results of the system

The results of the designed system were:

Table 1. Results of the experiment with RGB, HSV, CLD and the Borda Voting Schemes done with the experiments.

Corpus	RGB	HSV	CLD	Borda RGB+HSV+CLD
UW	87.31	94.72	31.11	50
Wang	79.33	72.33	53	82.66
ZuBud	77.18	82.08	38.39	66.42

5. Conclusion and future work

The approach presented in this paper has outperformed the precision level of HSV and RGB Histograms, and MPEG-7 CLD Descriptor (Wang database). The precision of CLD descriptor was poor, and this fact significantly affected the overall process of classification.

In the same way, the classification process in the other databases presents low level of precision, due to CLD descriptor precision level.

Given the above, is important to count on descriptors with similar levels of precision or recall. In this way, one descriptor can be good to classify some types of images, and other descriptor can be good to classify other types, giving us better results in higher global precision.

For future work we will incorporate the Boosting-Based classification, using uncorrelated MPEG-7 descriptors to analyze if is possible to improve precision in CBIR tasks.

Acknowledgment

This work was funded by the project "Using visual MPEG-7 descriptors for Content - Based Image Retrieval" of the Fifth Call for Research Founds of the Universidad Politécnica Salesiana.

References

[1] E. Spyrou, H. Le Borgne, T. Mailis, E. Cooke, Y. Avrithis, and N. O'Connor, "Fusing mpeg-7 visual descriptors for image classification," *Artificial Neural Networks: Formal Models and Their Applications–ICANN 2005*, pp. 747–747, 2005.

[2] P. Reddy, A. Reddy, and K. Devi, "HSV color histogram and directional binary wavelet patterns for content based image retrieval," *International Journal on Computer Science and Engineering (IJCSE)*, vol. 4, 2012.

[3] T. Deselaers, D. Keysers, and H. Ney, “Features for image retrieval: an experimental comparison,” *Information Retrieval*, vol. 11, no. 2, pp. 77–107, 2008.

[4] J. Poveda and V. Robles, “Image retrieval based on the conbination of RGB and HSV’s and color layout descriptor,” *Ingenius N-7*, pp. 3–10, 2012.

[5] K. Mekaldji and S. Boucherka, “Color quantization and its impact on color histogram based image retrieval,” in *Procedings of the Second Conference International sur l’Informatique et ses Applications (CIIA’09), Saida, Algeria, May 3-4*, 2009.

[6] Intersil, “YCbCr to RGB considerations.” [Online]. Available: http://www.intersil.com/content/dam/Intersil/documents/an97/an9717.pdf

[7] P. Harrington, *Machine Learning in Action.* New York: Manning Publications, 2012.

[8] O. Boullosa García, “Estudio comparativo de descriptores visuales para la deteccion de escenas cuasi-duplicadas,” Madrid, Spain, 2011, Proyecto de fin de carrera.

[9] V. Robles, “Esquemas de votación borda aplicados al etiquetado de roles semánticos,” Master’s thesis, Universidad Politécnica de Valencia, Valencia, Spain, 2010.

[10] S. Jeong, “Histogram-Based Color Image Retrieval,” *Psych221/EE362 Project Report. Palo Alto, CA: Stanford University*, 2001.