Classifying Images Collected on the World Wide Web

CAMILLO JORGE SANTOS OLIVEIRA, ARNALDO DE ALBUQUERQUE ARAÚJO
CARLOS ALBERTO SEVERIANO JR., DANIEL RIBEIRO GOMES

NPDI - Núcleo de Processamento Digital de Imagens, DCC - Departamento de Ciência da Computação
ICEx - Instituto de Ciências Exatas, UFMG - Universidade Federal de Minas Gerais
Caixa Postal 702, CEP: 30.161-970, Belo Horizonte, MG, Brasil.
{camillo, arnaldo, carlos, danielrg}@dcc.ufmg.br
http://www.npdi.dcc.ufmg.br

Abstract. This work presents the classification of images collected on the World Wide Web, using a supervised classification method, called ID3 (*Itemized Dichotomizer 3*). The classification consists in separating the images into two semantic classes: graphics and photographs. Photographs include natural scenes, like people, faces, animals, flowers, landscapes and cities. Graphics are logos, drawings, icons, maps, and backgrounds, usually generated by computer. To validate the classifier we used the *k*-fold cross-validation method. In the experimental tests 95.6% of the images were correctly classified.

1. Introduction

Because of the WWW expansion we have an enormous amount of information available to the Web user, like videos, text documents and images. Besides, another important factor is how fast images are added and moved on the Web. We cannot access or make use of the information unless it is organized so as to allow efficient browsing, searching, and retrieval.

Image retrieval has been active research area since the 1970s, with the trust from two major research communities, database management and computer vision. These two research communities study image retrieval from different angles, one being text-based and the other visual-based.

The text-based image retrieval can be traced back to the 1970s. A very popular framework of image retrieval. Image characteristics were represented by text and then a DBMS (Database Managing System) was used to deal with information retrieval, Chang *et al.* [1, 2]. However, there exist two major difficulties, especially when the size of image collection is large. One is the vast amount of labor required in manual image annotation. The other difficulty, which is more essential, results from the rich content in the images and the subjectivity of the human perception. That is, for the same image content different people may perceive it differently. The perception subjectivity and annotation impreciseness may cause unrecoverable mismatches in later retrieval processes.

In the early 1990s, because of the emergence of large-scale image collections, the difficulties faced by the manual annotation approach became more and more acute. To overcome these difficulties, content-based image retrieval was proposed. That is, instead of being manually annotated by text-based key words, images would be indexed by their own visual content like color, shape, and texture. Since then, many techniques in this research direction, Gupta *et al.* [3], Gudivada *et al.* [4] and Picard *et al.* [5], have been developed and many image retrieval systems (research and commercial) have been built. Bimbo [6] presents some these systems: QBIC, Virage, Visual Retrievalware, Macs-Hermes, Chabot, IRIS, Picasso, ICARS, Photobook, CANDID, VisualSeek, FIBSSR, CORE and NeTra. Content-based visual systems examples for Web are: WebSeer (Frankel *et al.* [9]), Webcrawling (Lew *et al.* [16]), ImageRover (Sclaroff *et al.* [17]). A very contributing area for this approach is the computer vision.

Until now, there are no generic algorithms to process all kinds of images. The image classification step becomes very important for the development of a WWW tool that retrieves images. Images can belong to different semantic classes, like photographs, graphics, maps, caricatures, pictures of people, cards, faces, color images, etc. After classification, images can be searched by classes, Abbadeni *et al.* [7].

This work presents the classification of images collected on the World Wide Web in two semantic classes: photographs and graphic (Oliveira [14] and Oliveira *et al.* [15]). It was used the ID3, supervised non-parametric classifying method, developed by Quinlan [11]. The ID3 is easy to comprehension and the direct approximation. This classification is important to photograph indexing. It is the first step in a content-based image retrieval system.

The rest of the paper is organized as follows. Section 2 shows the difference between photographs and graphics.

Section 3 presents the metrics, that are procedures capable to differentiate the two image types. Section 4 describes the ID3 classification method. Section 5 describes the experiments, followed by Section 6 that shows the experimental results. And finally, Section 7 gives some concluding remarks.

2. Differences Between Photographs and Graphics

To perform semantic classification of in the Web-collected images, it is necessary to define metrics that are able to distinguish the two types of images mentioned. The metrics are procedures that, when applied to an image, return numeric values capable of characterizing it. In this work, the metrics are based on the differences between photographs and graphics.

Figures 1, 2, 3, and 4 show an image (a) and its relative color histogram (b) in RGB (Red, Green, and Blue) format, where each pixel has a certain amount of color in each RGB channel. Each channel ranges in $[0,255]$. Analyzing the color histogram, we can see the number of colors, the prevalent color, colors that are absent, and in a qualitative view, the transition between the pixels of the image (more or less accentuated).

Some characteristics presented by the photographic images of our database can be seen in Figures 1 and 2. Usually, photographs present real objects with a tendency to have texture and the absence of regions with constant colors. Other remarkable characteristics are: small differences in aspect ratio (height x width); few occurrences of regions with high saturation of colors; presence of a large number of used colors.

Some characteristics presented by the graphic images of our database can be seen in Figures 3 and 4. Usually, graphics present artificial objects with well defined borders and the presence of regions covered with saturated colors. Other remarkable characteristics are: big differences in aspect ratio (height x width) and tendency to have a smaller size than photographs.

Graphics should be logos, drawings, maps, buttons and icons, usually computer generated, Abbadeni *et al.* [7], Athitsos *et al.* [8], and Frankel *et al.* [9].

3. Metrics

To implement a classifier, we need precise procedures, that we can apply to an image and get back results that give us information about the type of the image. This procedures we called metrics, which map images to real numbers. Photographs and graphics tend to have different ranges in the metrics suggested below. Because of that, the metrics scores are evidence that we can use to differentiate between those two types.

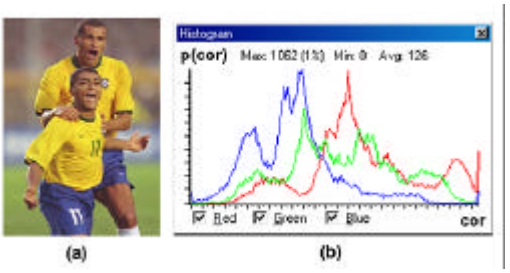

Figure 1 (a) JPEG Photograph and (b) histogram.

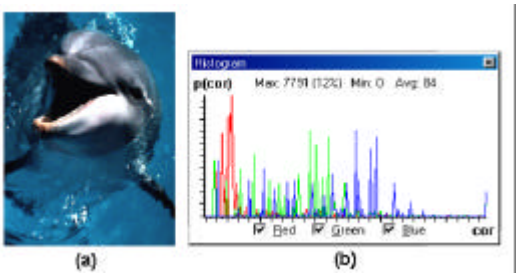

Figure 2 (a) GIF Photograph and (b) histogram.

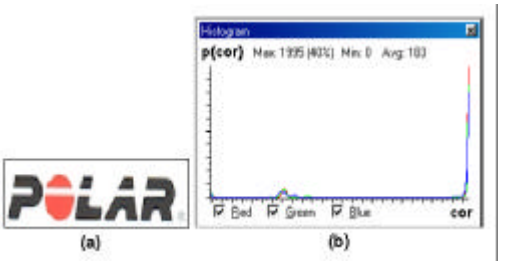

Figure 3 (a) JPEG Graphic and (b) histogram.

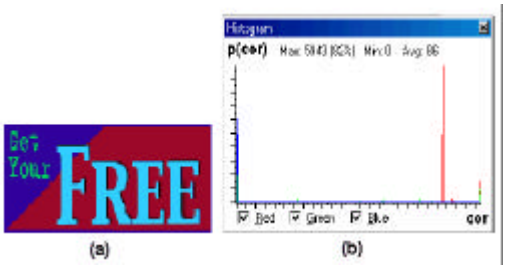

Figure 4 (a) GIF Graphic and (b) histogram.

4. Metrics

To implement a classifier, we need precise procedures, that we can apply to an image and get back results that give us information about the type of the image. This procedures we called metrics, which map images to real numbers. Photographs and graphics tend to have different ranges in the metrics suggested below. Because of that, the metrics scores are evidence that we can use to differentiate between those two types.

The metrics, according to Athitsos *et al.* [8], are: the number of colors, the prevalent color, the farthest neighbor,

the saturation, the color histogram, the farthest neighbor histogram, the dimension ratio, and the smallest dimension.

We assume that images are represented by three two-dimensional arrays, each corresponding to their RGB color bands. The entries of these arrays are integers from 0 to 255. The color vector of a pixel p is defined to be (r, g, b), where r, g and b are respectively the red, green and blue components of the color of the pixel.

We define next the metrics used in this work (Athitsos *et al*. [8]).

The number of colors metric. The score of the images in this metric is the number of distinct colors that appear in the image.

The prevalent color metric. It represents the most frequently occurring color in the image. The score of the image is the fraction of pixels that have that color.

The farthest neighbor metric. This metric is based on the assumptions we made about color transitions in graphics and photographs. For two pixels p_1 and p_2, with vectors (r_1, g_1, b_1) and (r_2, g_2, b_2). It was defined the color distance d as $d = |r_1 - r_2| + |g_1 - g_2| + |b_1 - b_2|$. Since color values range from 0 to 255, d ranges from 0 to 765. Each pixel p_1 (except for the outer pixels) has neighbors up, down, left and right. A neighbor p_2 of p_1 is considered to be the farthest neighbor of p_1 if the color distance between p_1 and p_2 is not smaller than the color distance between p_1 and any other of its neighbors. The transition value of p_1 is the distance between p_1 and its farthest neighbor.

The saturation metric. This metric is based on the assumption that highly saturated colors are more common in graphics than in photographs.

For a pixel p with color vector (r, g, b), let m be the maximum and n be the minimum among r, g and b. The saturation level of p is $|m - n|$.

The color histogram metric. This metric is based on the assumption that certain color occurs more frequently in graphics than in photographs. In contrast to the saturation metric, it does not assume anything about the nature of those colors. It consists to collect statistics from a large number of graphics and photographs and construct histograms which show how often each color occurs in images of each type. The score of an image depends on the correlation of its color histogram to the graphics histogram and the photographs histogram.

A color histogram is a three dimensional table of size $16 \times 16 \times 16$. Each color (r, g, b) corresponds to the bin indexed by $\left(\left\lfloor \frac{r}{16} \right\rfloor, \left\lfloor \frac{g}{16} \right\rfloor, \left\lfloor \frac{b}{16} \right\rfloor\right)$ in the table. The color histogram of an image contains at each bin the fraction of pixels in that image whose colors correspond to that bin.

The correlation $C(A, B)$ between two histograms A and B is defined as:

$C(A, B) = \sum_{i=0}^{15} \sum_{j=0}^{15} \sum_{k=0}^{15} \left(A_{i,j,k} B_{i,j,k}\right)$, where $A_{i,j,k}$ and $A_{i,j,k}$ are respectively the bins in A and B indexed by (i, j, k).

A color histogram H_g is created by picking hundreds of thousands of graphics, and taking the average of their color histograms. A color histogram H_p is created using a large set of photographs.

Suppose that an image I has a color histogram H_i. Let $a = C(H_i, H_g)$ and $b = C(H_i, H_p)$. The score of the image in the color histogram metric is defined as $s = \frac{b}{a + b}$. Clearly, as $C(H_i, H_p)$ increases, s goes up, and as $C(H_i, H_g)$ increases, s goes down. Therefore, we expect photographs to score higher in this metric.

The farthest neighbor histogram metric. This metric is based on the same assumptions as the farthest neighbor metric, but provides a different means of testing an image.

The farthest neighbor histogram of an image is a one dimensional histogram with 766 bins (as many as the possible transition values for a pixel). The *i-th* bin contains the fraction of pixels with transition values equal to i. A graphic histogram F_g is created by averaging the farthest neighbor histograms of hundreds or thousands of graphics. We create a photographs histogram F_p in the same way, using a large set of photographs. The correlation $D(A, B)$ between histograms A and B is $D(A, B) \sum_{i=0}^{755} A_i B_i$, where A_i and B_i, are respectively the *i-th* bins of A and B.

Let F_i be the farthest neighbor histogram metric of the image, $a = D(F_i, F_g)$ and $b = D(F_i, F_p)$. Then, the score s of the image in this metric is defined as $s = \frac{b}{a + b}$. It is expected that photographs presents scores higher than graphics.

The dimension ratio metric. Let w be the width of the image in pixels, h be the height, m be the greatest of w

and h, and l be the smallest of w and h. The score of an image is $\frac{m}{l}$.

The smallest dimension metric. The score of an image is the length of its smallest dimension in pixels. It is much more common for graphics to score below 50 in this metric than it is for photographs.

5. Classification

Han *et al*. [10] said that classifying consists in separating distinct sets of objects or annotations, allocating new objects or annotations into groups previously defined. To perform classification, it is necessary an algorithm to separate and allocate objects or annotations. This algorithm is called classification technique. The final objective of a classification method is to provide relevant results or reply a specialist judgement. The relative performance of different classification techniques may depend of data conditions.

Wu [12] said that knowledge acquisition from databases has been worked over by researches in several disciplines including Artificial Intelligence for over twenty years. Although a lot of work has been done and some commercial learning packages are available already, existing work has concentrated on the following four aspects: (1) building knowledge bases for expert systems, (2) designing various learning algorithms; (3) adding an induction engine to an existing data base system in an *ad hoc* way to implement rule induction from data bases; and (4) designing a specific engine to learn from a domain-specific data set. Along with the recognition of the so called knowledge bottleneck problem in transforming knowledge from human experts to knowledge-based systems, learning from examples has played a major role in machine learning and knowledge-based system and is still an important research frontier for both machine learning and data base technology.

4.1 ID3 Development

Using the concept formation model, Quinlan [11] developed the ID3 method, which is a supervised learning method, with the ability of generating rules through a decision tree. To construct the decision tree, is necessary to calculate the entropy of the features of the training samples. The entropy tells us the disorder degree among the features. We can induce the construction of a tree, respecting the hierarchy among the features. This is the TDIDT (Top-Down Induction Decision Tree) method. Quinlan [11] adopted the strategy of dividing to conquer associated with logic. That means, to use the classification criterion to separate in smaller groups. The result of this experiment was impressive, with short time processing.

4.2 ID3 Implementation

Before the algorithm implementation, some terminologies must be defined. Define I to be a set of training samples. Each subset of I, including I itself on the first iteration, will be called "window".

Here is an outline of the ID3 algorithm:

1. Randomly select a "window" (subset) of I.
2. Use the CLS (Concept Learning System) to generate a rule, considering the decision tree structure produced by the "window".
3. Perform iterations in all subset I, The elements that have not been classified by the rules generated in item 2.
4. Generate a new "window" adding to the current "window" the elements not classified in item 3.
5. Repeat item 2 until there are not unclassified elements in set I.

4.3 Entropy

A mathematical definition of ID3 is that it algorithmically determines the greatest gain in information content while decreasing system entropy (TDIDT method). The concepts of information content and entropy recur in many aspects of computer science, including information theory, algorithms, and data compression. Stated succinctly, information content is the amount of data held in each unit of representation (usually bits), and entropy is the least unit of representation necessary to communicate a given data set.

Let W be a subset ("window") of a training sample, m be the number of elements of the "window" W, and n_a be the number of instances of the element m in W. The probability p_a of choosing a in W is defined as:

$$p_a = \frac{n_a}{m} \tag{1}$$

For a simple system with classes c_i, i = 1, 2,...,C, where C be the number of classes. The entropy of the system can be defined as:

$$Entropy = \sum_{i=1}^{C} - p_i \log_2 p_i \tag{2}$$

4.4 General Case

In this section, a mathematical interpretation of ID3 is taken. The complexity of the equation can make ID3 method quite difficult to follow due to the large number of variables.

Let N be the number of elements (known patterns) partitioned into sets of matching pattern classes c_i,

i=1,2,...,C, and the number of elements in "window" c_i is N_i. Assuming that all features have J values, let each pattern have K features and each feature have J_k values.

The overall mathematical goal of ID3 is to reorganize data so as to create an efficient decision tree. This is done following the steps below:

1. Calculate the initial entropy of the training set T according to Equation (2).
2. (a) A feature must be selected to be the root node of the decision tree. This is done partitioning the training set T into K training subsets: for each feature A_k, k=1,2,...,K, where J values of the features A_k. The number of features in the a_{kj} branch is n_{kj}. Note that these sub trees have no inherent relationship to the final group criteria.

   (b) The number of patterns belonging to a class c_i in any branch of the population n_{kj} is defined as $n_{kj}(i)$. The entropy of each branch n_{kj} is evaluated by:

$$Entropy(T, A_k, j) = \sum_{i=1}^{C} -\frac{n_{kj}(i)}{n_{kj}} \log_2 \frac{n_{kj}(i)}{n_{kj}} \qquad (3)$$

   (c) The entropy of each subset is a partial result – the overall system entropy is what ID3 needs to consider. This is calculated as:

$$Entropy(T, A_k) = \sum_{j=1}^{J} \left(\frac{n_{ij}}{\sum n_{ij}} \right) Entropy(T, A_k, j) \qquad (4)$$

   (d) The decision tree with the least entropy will be the one ID3 selects. The reduction in entropy can be easily calculated as:

$$Entropy(T) - Entropy(T, A_k) = \Ä Entropy(k) \qquad (5)$$

   (e) Find the feature A_{k_0} that gives the greatest decrease in entropy. Mathematically, find A_{k_0} such that $\Ä Entropia(k_0) > \Ä Entropia(k)$, for all k=1,2,...,K, and $k \neq k_0$.

   (f) The node A_{k_0} becomes the root of the decision tree.
3. Synthesize the next level of the decision tree by finding the feature A_k that, after testing all branches, yields the greatest decrease in system entropy. Form subsets of the previous level by separating T according to the value of A_k. It is important to note that the same feature is tested along the breadth of each tree depth level.
4. Repeat this algorithm until all subsets are made up of the same conclusion criteria or until overall system entropy is zero.

Figure 5 shows an example of decision tree, where each non-leaf node is a feature and each leaf node is a class (a known pattern).

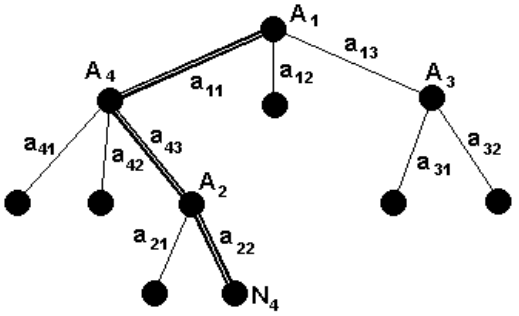

Figure 5 Example of the decision tree.

The decision tree built is a structure which groups the rules learned during the training. A rule is a logical expression that is able to produce as a result the belonging class of the object. An example of logical expression synthesized from a decision tree could be $A_1(a_{11}) \wedge A_4(a_{43}) \wedge A_2(a_{22}) \rightarrow C(N_4)$, which represents the highlighted path in Figure 5. For the feature A_1 applying a threshold results the subset a_{11}, for the feature A_4 applying a threshold results the subset a_{43}, for the feature A_2 applying a threshold results the subset a_{22}, which results in the leaf node N_4 which is the class, result of the classification. This expression can be represented using conditional selection structures *if-then*, what would be similar to:

If (feature value $A_1 = a_{11}$ and feature value $A_4 = a_{43}$ and feature value $A_2 = a_{22}$)

Then (all classifications $c_i \in N_4$ can be assumed)

The set $N_4 \subseteq C$ is the set of one or more conclusions, $c_i \in C$, that occurs in this node. The expression $C(N_4)$ indicates that all classifications $c_i \in N_4$ can be confirmed. The best case would be that each path in the tree would finish in a leaf node containing a unique conclusion ($|N_i| = 1 \forall i$). Each classification tree should be able to distinguish each one of the examples that it learned from the training sample T.

4.5 Advantages and Drawbacks

One of the strongest benefits of ID3 is its simplicity, when compared to other learning algorithms. ID3 is much more straightforward in its approach. Its cognition-based modeling make it relatively simple for humans to understand how ID3 works. Unfortunately, the decision trees produced by ID3, when used to process large or noisy data sets, tend to be confusing to human perception.

ID3 performs very well given large and complex data sets. One of the way used by ID3 to do this is to find

"hidden" data and relationships. It looks at problems using a simple-minded divide-and-conquer technique.

Another benefit of ID3 is its conservative use of system resources. The computational time involved in ID3 is linear and can be calculated as the product of the number of training objects, the number of possible features which describe each object, and the complexity of the final selection criteria (measured as the number of nodes in the decision tree).

A major disadvantage of ID3 is that the decision trees produced are essentially immutable - one can not efficiently change the decision tree without rebuilding it from scratch. Using a patchwork method of updating the tree tends to yield a decision tree, which is no longer optimal, thus refuting the original purpose of forming the ID3 decision tree.

6. The Experiments

A software robot obtained the image collection. The robot received as input a *URL* (*Universal Resource Locator*) list of various servers. For each server, each of the main *HTML* (*Hypertext Markup Language*) file was analyzed. All links for the others *HTML* files were extracted and the addresses of images with extension JPEG (*Joint Photographic Experts Group*) or GIF (*Graphics Interchange Format*) found were marked to retrieve the images and saved them to the local disk (where the software robot was run). Figure 6 shows the robot sketch. The robot was put in execution for approximately eight days, when it collected approximately ten gigabytes of GIF and JPEG format images from various domains.

Training samples were separated. This step consisted in separating the images into four groups, which are: GIF graphics, GIF photographs, JPEG graphics, and JPEG photographs. The process of separating training samples (Figure 7) was performed by visual inspection. This turned the work a little boring. We separated 1350 GIF photographs, 3058 GIF graphics, 4763 JPEG photographs and 1434 JPEG graphics. Images that have one of the dimensions smaller than 50 pixels were not considered.

For each training sample the following metrics were applied: the number of colors, the prevalent color, the farthest neighbor, the saturation, the color histogram, the farthest neighbor histogram, the dimension ratio, and the smallest dimension. The result was a numeric vector (with its values normalized), called tuple.

The next step was the application of the ID3 method, generating a decision tree based on the entropy of the features.

To evaluate the model (decision tree), we used the k-fold cross validation method, described by Kohavi [13], with k = 2, 4, 5, 8, 10 and samples with 200, 400, 800, 1000 and 1200 images.

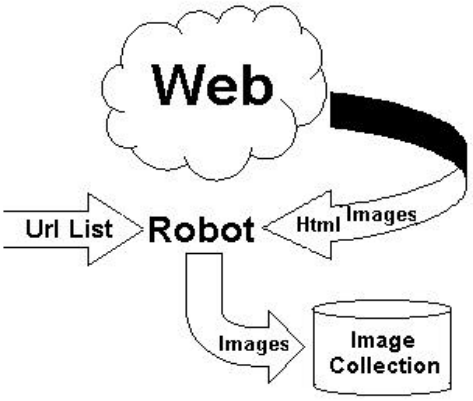

Figure 6 The robot sketch.

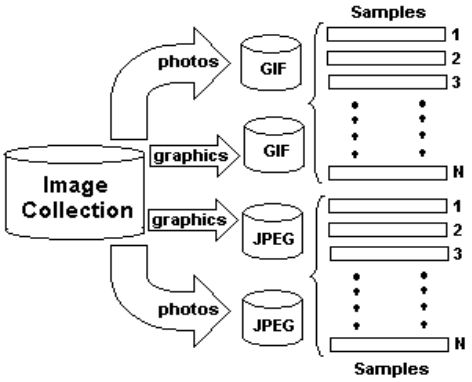

Figure 7 Obtaining the training samples.

To verify the accuracy of the rule set (decision tree), it was necessary to calculate the percentage of the number of test data item correctly classified, viz.:

$$\%\,Accuracy = \frac{Number\ \ of\ \ correct\ \ classification}{Number\ \ of\ \ test\ \ data\ \ items} \quad (6)$$

As data handling is distinct for the two image formats, we generated decision trees for both image formats (JPEG and GIF).

7. Experimental Results

The results presented by k-fold cross validation method show that the model is stable. The standard deviation on error rates remained almost constant independent of the value of k. Figures 8, 9, 10, and 11 show that the best decision trees generated were the samples with 800, 1000, and 1200 images.

Figure 8 Error rates for GIF images.

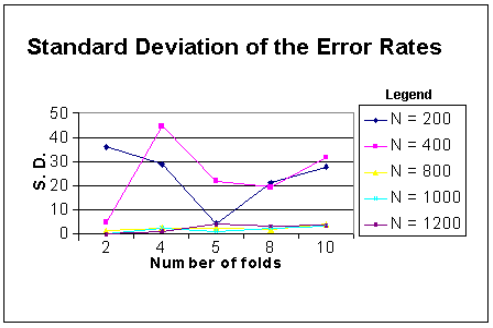

Figure 9 Standard deviation of error rates for GIF images.

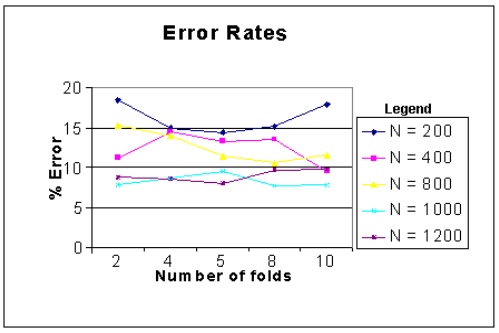

Figure 10 Error rates for JPEG images.

One iteration consists in generating the decision tree for the images that are inside the “window” and testing the images that are outside the "window". If success is not obtained when classifying all the test images, we include in the “window” those images that have wrong classification and start another iteration. This process is repeated until all sample elements are classified correctly (100% of correct classification).

Figures 12 and 13 show the average number of iterations for each k-fold cross validation, with k = 2, 4, 5, 8, and 10 and samples with 200, 400, 800, 1000, and 1200 images. The number of iterations does not grow proportionally to the number of folds and does not grow proportionally to the sample size. There is no significant difference between the number of iterations for the two image formats.

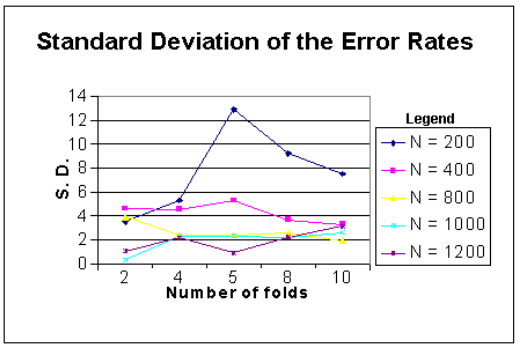

Figure 11 Standard deviation of error rates for JPEG images.

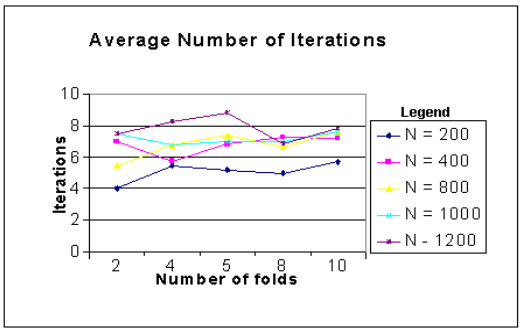

Figure 12 Number of iterations for GIF images.

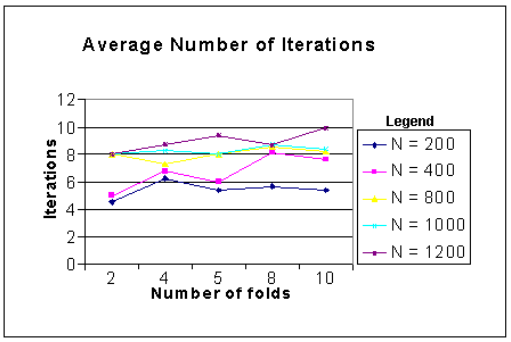

Figure 13 Number of iterations for JPEG images.

Using the decision tree generated from a training sample of 1200 images, we perform classification of unknown images (images that did not participate on the training). We evaluated 10 sets (samples) of 250 images and the average result of classification was 97.3% for GIF images and 93.9% for JPEG images, with standard deviation of 1.6 and 2.6, respectively.

Some images have graphical and photograph information, for example, the images that have border. Some images have appearance and comportment of one type but they are other type. This images did not obtain the correct classification. Figure 14 shows some examples of images that obtained wrong classification.

Figure 14 The images that obtained wrong classification.

8. Conclusions

This work presents the classification of images collected on the World Wide Web, using a supervised classification method, called ID3 (*Itemized Dichotomizer 3*). The classification consists in separating the images into two semantic classes: graphics and photographs.

The ID3 method generated a stable rules (induced decision tree) that was capable to separate the images into two classes (photographs and graphics). The average result of classification was 97.3% for GIF images and 93.9% for JPEG images.

In future work we want to implement new metrics, use other classifiers such as CN4.5-rules (it is a commercial version of the ID3) or Neural Nets. Finally, we are initially focusing on techniques to classify images into the classes "textured" vs. "non-textured", "indoor" vs. "outdoor", "city" vs. "landscape", and "with people" vs. "without people".

9. Acknowledgements

The authors are grateful to CNPq, CAPES/COFECUB, FAPEMIG and the SIAM DCC/PRONEX Project for the financial support of this work.

References

[1] Chang, S.-K.: *Pictorial data-base systems*, IEEE Computer, (1981).

[2] Chang, S.-K., Yan, C. W., Dimitroff, D. C., Arndt, T.: *An intelligent image database system*, IEEE Transactions on Software Engineering, 14(5), (1988).

[3] Gupta, A., Jain, R.: *Visual information retrieval*, Communications of the ACM 40(5), (1997), 71-79.

[4] Gudivada, V. N., Raghavan, J. V.: *Special issue on content-based image retrieval systems*, IEEE Computer Magazine 28(9), (1995), 18-22,.

[5] Picard, R. W., Pentland, A. P.: *Introduction to the special section on digital libraries: representation and retrieval*, IEEE PAMI 18(8), (1996), 769-770.

[6] Bimbo, A. D.: *Visual information retrieval*, Morgan Kaufmann, 270p., 1999.

[7] Abbadeni, N., Ziou, D., Wang. S.: *Image classification and retrieval on the* WWW, Proceedings of the 4th ACM Conferences on Digital Libraries, (1999), 208-209.

[8] Athitsos, V., Swain, M. J., Frankel, C.: *Distinguishing photographs and graphics on the* WWW, Proceeding of IEEE Workshop on Content-Based Access of Image and Video Libraries, (1997).

[9] Frankel, C., Swain, M. J., Athitsos, V.: *WebSeer: an image search engine for the* WWW, University of Chicago, Technical Report 9614, (1996), Computer Science Department.

[10] Han, I., Chandler, J. S., Liang, T. -Peng.: *The impact of measurement scale and correlation structure on classification performance of inductive learning and statistical methods*, Expert Systems With Applications, Elsevier,, 10(2), (1996), 209-221.

[11] Quinlan, J. R.: *Induction of decision tress*, Machine learning, v. 1, (1986), 81 – 106.

[12] Wu, X.: *Induction by attribute elimination*, IEEE TKDE, September/October, 11(5), (1999), 805-812.

[13] Kohavi, R.: *A study of cross-validation and bootstrap for accuracy estimation and model selection*, IJCAI, (1995).

[14] Oliveira, C. J. S.: *Classificação de imagens coletadas na web*, dissertação de mestrado, Departamento de Ciencia da Computacao, Universidade Federal de Minas Gerais, (2001), 73p.

[15] Oliveira, C. J. S., Araújo, A. de A., Severiano Jr., C. A., Gomes, D. R.,: *Proposal of a Classifier of Images Collected in the World Wide Web*, Proceedings of the 14th Brazilian Symposium on Computer Graphics and Image Processing, IEEE Computer Society Press, Brazil, October (2001).

[16] Lew Lew, M. S., Lempinen, K., Huijsmans, D. P., *Webcrawling using sketches*, Technical Report, Computer Science Department, Leiden University, The Netherlands, (1997).

[17] Sclaroff, S., Taycher, L., La Cascia, M., *A content-based image browser for the world wide web*, Proceedings of the IEEE Workshop on Content-Based Access of Image and Video Libraries, (1997).