Image Classification to Improve Printing Quality of Mixed-Type Documents

Rafael Dueire Lins, Gabriel Pereira e Silva
UFPE, Recife, Brazil
rdl@ufpe.br, gfps@cin.ufpe.br

Steven J.Simske, Jian Fan, Mark Shaw,
HP Labs., Palo Alto, USA
{steven.simske, jian.fan, mark.q.shaw}@hp.com

Paulo Sá, Marcelo Thielo
HP Labs., Porto Alegre, Brazil
paulo.sa@hp.com
marcelo.resende.thielo@hp.com

Abstract

Functional image classification is the assignment of different image types to separate classes to optimize their rendering for reading or other specific end task, and is an important area of research in the publishing and multi-Average industries. This paper presents recent research on optimizing the simultaneous classification of documents, photos and logos. Each of these is handled during printing with a class-specific pipeline of image transformation algorithms, and misclassification results in pejorative imaging effects. This paper reports on replacing an existing classifier with a Weka-based classifier that simultaneously improves accuracy (from 85.3% to 90.8%) and performance (from 1458 msec to 418 msec/image). Generic subsampling of the images further improved the performance (to 199 msec/image) with only a modest impact on accuracy (to 90.4%). A staggered subsampling approach, finally, improved both accuracy (to 96.4%) and performance (to 147 msec/image) for the Weka-base classifier. This approach did not appreciable benefit the HP classifier (85.4% accuracy, 497 msec/image). These data indicate staggered subsampling using the optimized Weka classifier substantially improves the classification accuracy and performance without resulting in additional "egregious" misclassifications (assigning photos or logos to the "document" class).

1. Introduction

Image clustering has been researched by the database community since the early 1980's aiming to make efficient information retrieval in image databases [1][2]. In that kind of application one image is used to search the database looking for either the same or similar images. The basic idea is to try to organise the images in the database using some "common" features [3][2]. The same "features" are used to analyse the image that will serve as the "search-key". Instead of stepping through the whole database image-by-image, the retrieval process tries to match the properties of the search-key image, also known as *query image*, with the different image clusters in the database. This largely reduces the search-space making the retrieval process far more efficient. One of the features that has presented greater success in image retrieval was the analysis and clustering by the colour histogram [4][5]. The semantics of images have also been used as a clustering method [6] in database retrieval. Images that have similar "motifs" are most likely to have properties that are common to each other forming clusters. On the other hand, images whose theme completely uncorrelated should exhibit very different properties.

Image classification is used in all-in-one and multi-functional devices to differentially render images belonging to different clusters. In particular, document, photo and logo images require widely different imaging pipelines to optimize their appearance when copied or printed. Documents (text, tables), for example, require sharpening that would damage the appearance of photos and logos. Logos use a palette that would "posterize" photos. Photos, in turn, can be rendered with a lower resolution (but greater bit depth) than either documents or logos. Figure 01 shows examples of typical representatives of the three classes of interest herein.

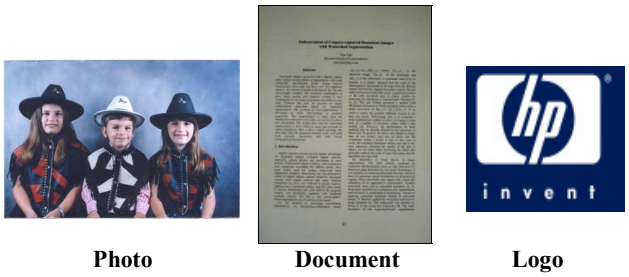


Photo **Document** **Logo**

Figure 01. Example images of the clusters of interest

This paper improves the results of the classifier described in reference [7] and presents a new, Weka-

978-0-7695-3725-2/09 $25.00 © 2009 IEEE
DOI 10.1109/ICDAR.2009.167

based classifier [8] used to distinguish between these three types of images. Part 2 describes the experiments performed. Part 3 summarizes the results. We conclude with a discussion of the results.

2. Experiments Performed

The starting point for this work was collecting images that are representative of the different clusters of interest. Images were classified one-by-one by one person and the data-set was checked by three other people to avoid misclassifications and repetitions of the same image. Sometimes the "same" image appears in the test set in different file formats, for instance an image may appear in jpg, tiff, and bmp, as their features (palette, gamut, size) change from a format to another. Figure 01 shows an example of each of the classes of image of interest for this work. Images that do not belong to any of those classes are classified as "Don´t know".

The "Photo" cluster encompassed many different sorts of photos, which ranged from people, landscapes, objects and even documents. Most photos were true-color although there were grey scale ones. The resolution also varied widely from VGA (480x640 pixels) to 7.2 Mpixels. The photos were collected from family albums of the people linked to the authors to ones obtained from the Internet. As professionals of many different areas start to use portable digital cameras to acquire images of documents, such images were included in this study, bringing an extra level of difficulty: a document acquired with a camera is classified as a photo or a document? The answer to that question is not straightforward and may puzzle even humans. The criterion adopted here was that if the image encompasses only the document it is classified as "document" if parts of the surroundings are included it is classified as "photo". That means that if the document image in the rightmost part of Figure 02 is classified as "photo" and the same image after being processed in a tool such as PhotoDoc [9]. The "photo" test set has 7,968 photos of people and landscape and 500 photos of documents.

Grey-level	Landscape	Document

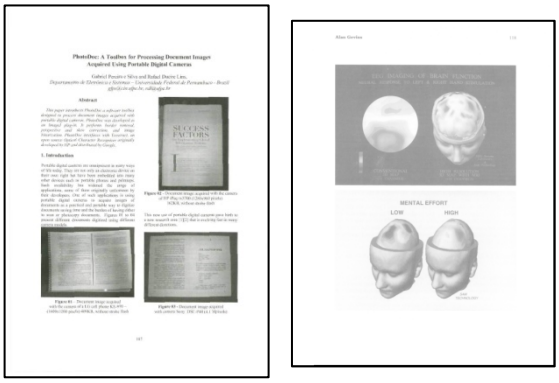
Figure 02. Example of photos

The 3,051 logos in the test set were collected from the Internet and from many different sources. Logos tend to exhibit a palette with a small number of colors, although very often they are saved in jpeg file format, introducing hues not originally intended. Figure 03 presents some examples of images classified as logos.

Figure 03. Examples of logos

The "Document" cluster was formed by 3,856 images of documents acquired from several different ways. Five hundred documents were photographed with a Sony Cybershot digital camera DSC-W55 in 5 and 7.2 Mpixels, with and without mechanical support, in-built strobe flash on and off, and then processed with PhotoDoc [9] that crops the framing border and corrects perspective and skew, should be classified as "document". About half of the remaining documents were scanned documents with different resolutions (from 100 to 300 dpi) and saved in bmp, tiff, and jpeg, which although not suitable for such kind of image is often used by people in general [10]. The remaining photos were obtained by saving Adobe pdf documents into tiff and jpeg.

Figure 04 presents some examples of document images used in this work.

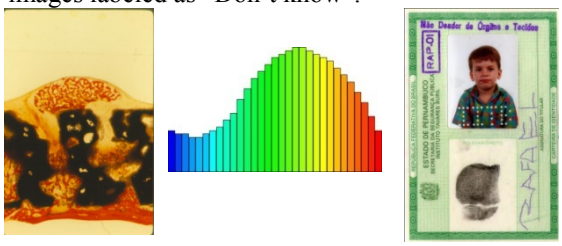


PhotoDoc processed **Scanned document**

Figure 04. Examples of documents

The last cluster of images in the test set is the "Don´t Know" images. These images were included as to increase the possibility of misclassifications. They are images that appear in the "real world" and range widely in nature from biological images, to vector graphics (obtained by softwares such as Excell®,

Powerpoint®, etc.). Figure 05 shows examples of images labeled as "Don´t know".



Biomedical **Vector graphics** **Document**

Figure 05. Examples of "Don´t Know" images

Table 01 shows the numbers of images per file format in the test set.

	JPG	TIFF	BMP	Total
Photo	7,476	35	457	**7,968**
Logo	2,984	0	67	**3,051**
Doc	3,048	808	0	**3,856**
Don´t know	202	0	327	**529**
Total	**13,710**	**843**	**851**	**15,404**

Table 01 – Images per file formats

2.1 Features Tested

The choice of the features to be extracted and tested is the key to the success and performance of the classification. Image entropy is often used as the key for classification [7]. It has a large computational cost, however. Entropy calculation demands a scan in the image to calculate the relative frequency of a given color, for instance, which is than multiplied for its logarithm and added up. The existing classifier is based on the binary classification approach originally described in [7]. It assumes a Gaussian distribution for each of the features, and its performance degrades in proportion to the non-Gaussian nature of the data.

This work assumed that decreasing the gamut of an image, analyzed together with its grey scale and monochromatic equivalents would provide enough elements for a fast and efficient image classification. The features tested are:

- Palette (true-color/grayscale)
- Gamut
- Conversion into Grayscale (if RGB)
- Gamut in Grayscale (if RGB)
- Conversion into Binary (Otsu)
- Number of black pixels in binary image.
- (#Black_pixels/Total_#_pixels)*100%
- (Gamut/Palette)*100% (true-color/grayscale)

Image binarization is performed by using Otsu [11] algorithm. The data above are extracted for each image and placed in a vector of features.

2.2 Training and test sets

Table 02 summarizes some of the features of the images in the test set. The height and width stand for the number of pixels in the image. RGB size stands for the true color size of the image (if a color image). 8-bits size is either the size of the original image if in gray scale or the size of the grey-scale converted from true-color. #B_pixels stands for the number of black pixels in the monochromatic converted image.

Photo	**Average**	**Median**	**Variance**	**Deviation**
height	1138	1104	387968	622
width	1274	1132	548014	740
RGB size	1.98MB	1.49MB	460646	214627
8-bits size	667KB	578KB	675668	7548028
gamut RGB	13641	9956	171547287	5884
gamut gray	231	247	1288	0,707
#B_pixels	1373240	755150	23511	153332
Logo	**Average**	**Median**	**Variance**	**Deviation**
height	232	180	19324	139
width	253	221	13066	114
RGB size	15KB	7KB	45800	214010
8-bits size	9KB	5KB	50658	609718
gamut RGB	5324	3848	37149265	6095
gamut gray	230	241	1102	33
#B_pixels	38785	6579	59580	244095
Doc	**Average**	**Median**	**Variance**	**Deviation**
height	1896	1734	325997	570
width	1437	1328	201513	448
RGB size	1.10MB	879KB	111962	334581
8-bits size	674KB	568KB	207357	143999
gamut RGB	2700	2097	16317414	4039
gamut gray	181	211	5502	74
#B_pixels	1310386	749770	157416	3967564
Don´t Know	**Average**	**Median**	**Variance**	**Deviation**
height	477	363	180173	424
width	574	490	208282	456
RGB size	1.55MB	1.33MB	129678	3387135
8-bits size	520KB	386KB	323578	172717
gamut RGB	3954	2934	207165	5514
gamut gray	217	198	959	28
#B_pixels	101896	54475	82345	3169822

Table 02 – Main features on the images in the test set

The training set was carefully selected to guarantee the diversity of the images in the test set, having in mind that quality matters more than size. Table 03 presents the relative size of the training and test sets.

	Test	Training	%
Photo	**7,968**	**668**	**8.34**
Logo	**3,051**	**412**	**10.22**
Doc	**3,856**	**276**	**4.70**
Don´t know	**529**	**0**	**0**
Total	**15,404**	**1,356**	**8.80**

Table 03 – Sizes of Training x Test sets

The Weka [8] classification strategy used was the Random Forests (number of trees equal to 10) [12].

2.3 Sub-sampling

The factors for the feature extractor to improve were:

1- The larger the file - the richer in data redundancy - thus if the redundant data are thrown away the efficiency both in time and classification.

2- The selection of points should not be random. It should somehow provide a "reduced" version of the original image (although in some cases it may be distorted by unequal scaling!).

Twenty different sub sampling strategies were evaluated on the images. Two are presented here. The first, designated the "simple" subsampling technique, consisted of splitting the image in blocks of 4x4 pixels and averaging their values. This sub sampler provided the best overall improvement in performance for a subsampling approach that did not involve a decision tree. The second, the cascaded subsampling strategy, consisted of removing more points from the larger image files and provided the best overall accuracy of any classification schema, while simultaneously significantly improving performance, as shown in the next section. The cascaded subsampler performs the following operations:

size = height*width

- If size ≤ 300,000 break;
- If 300,000< size ≤ 500,000:
  remove 1 line or 1 column (whatever the larger);
- If 500,000 < size ≤ 700,000:
  remove 1 line and 1 column;
- If 700,000 < size ≤ 900,000:
  remove 2 lines and 1 column, (if height>width)
  remove 1 line and 2 columns, otherwise;
- If 900,000 < size remove 2 lines and 2 columns;

Code for the "cascaded" sub-sampler

3. Results

This section presents the results of classification of the images in the test set comparing the current classifier [7] and the one proposed herein. Both classifiers had the same training and test sets. The results are divided into three groups: original, simple (averaging), and cascaded subsampling.

3.1 Original Data

The results of classification are presented by the confusion matrices obtained. Table 04 presents the results for the current classifier, while Table 05 shows the results obtained with the new classifier.

Current	Photo	Logo	Document	DK	A
Photo	7280	620	12	56	0.914
Logo	429	2104	96	422	0.690
Document	206	351	3299	0	0.856
Don´t Know	70	225	0	234	0.442

Table 04 – Confusion matrix of the **current** classifier with original images

The mean accuracy of the current classifier (A) for the Photo, Logo and Document images was 12638/14875 = 85.3%.

New	Photo	Logo	Document	DK	A
Photo	7554	363	14	37	0.948
Logo	282	2730	23	16	0.894
Document	277	266	3314	0	0.859
Don´t Know	151	309	17	52	0.098

Table 05 – Confusion matrix of the **new** classifier with original images

The mean accuracy (A) for the Photo, Logo and Document images was 12893/14875 = 86.3%, which shows that the proposed classifier is slightly better than the current one.

3.2 Simple Subsampler Performance

The results for the 4x4-pixel averaging subsampler are shown in Tables 06 and 07.

Current	Photo	Logo	Document	DK	A
Photo	5009	2209	586	164	0.628
Logo	1280	1005	115	651	0.329
Document	1245	376	2183	52	0.566
Don´t Know	101	154	15	259	0.489

Table 06 – Confusion matrix of the **current** classifier with subsampled images (simple)

The mean accuracy (A) for Photo, Logo and Document for the simple subsampler with the current classifier was 8197/14875=55.1%.

New	Photo	Logo	Document	DK	A
Photo	6674	1043	138	113	0.837
Logo	263	2751	20	17	0.901
Document	381	61	3414	0	0.885
Don´t Know	124	330	26	49	0.092

Table 07 – Confusion matrix of the **new** classifier with subsampled images (simple)

The mean accuracy (A) for the Photo, Logo and Document images was 12839/14875 = 86.3%.

The simple subsampler performed as well as the original version of the new classifier, but drastically degraded the accuracy of the current one. The simple subsampler halved the time for feature extraction of the new classifier as is shown in section 4.

3.3 Cascaded Subsampler Performance

The results for the cascaded subsampler are found in tables 08 and 09.

Current	Photo	Logo	Document	DK	A
Photo	7603	275	18	72	0.954
Logo	385	1929	135	602	0.632
Document	311	373	3167	5	0.821
DK	77	174	128	150	0.283

Table 08 – Confusion matrix of the **current** classifier with **cascaded** subsampled images

The mean accuracy (A) for the Photo, Logo and Document images for the cascaded subsampler using the current classifier is 12699/14875 = 85.3%

New	Photo	Logo	Document	DK	A
Photo	7740	164	34	30	0.971
Logo	258	2761	11	21	0.904
Document	93	41	3722	0	0.965
DK	110	300	30	89	0.343

Table 09 – Confusion matrix of the **new** classifier with **cascaded** subsampled images

The mean accuracy (A) for the Photo, Logo and Document images was 14223/14875 = 95.6%. As one may observe the cascaded subsampler largely improved the performance of the new classifier and has a positive effect in performance, as shown below.

4. Time Performance

Table 10 presents the feature extraction and classification times together with information about the language those procedures were implemented into. Besides classification accuracy per cluster, the average feature extraction and classification times are presented. The entries with an "S" superscript denote the "simple" subsampler, while de "C" superscript stand for the "cascaded" subsampler. One should also remark that there is a difference in time scale between feature extraction and classification.

	Feature extraction		**Classification**	
	Time (s)	**Language**	**Time (ms)**	**Language**
Current	**1.4576**	**C#**	**6.13**	**C#**
New	**0.4174**	**C++**	**0.12**	**C#**
Current^{S}	**0.719**	**C#**	**6.13**	**Java**
New^{S}	**0.199**	**C++**	**0.12**	**C#**
Current^{C}	**0.497**	**C#**	**6.13**	**Java**
New^{C}	**0.1470**	**C++**	**0.12**	**C#**

Table 10 – Feature extraction and classification times

5. Discussion and Conclusions

Weka has shown to be an excellent test bed for statistical analysis. The choice for a Random tree classifier was made after performing several experiments with the large number of alternatives offered by Weka, although results did not vary widely. Amongst them a preliminary comparison between the new statistical classifier proposed here and a MLP neural classifier provided worse results (Photos 91.37%, Logos 85.48%, and Documents 94.54%).

The choice of the images in the training set is of paramount importance to the performance of the classifier. Quality has proved more important than size. For some reason not fully understood, the current classifier seems to be more sensitive to the quality of the training set than the one proposed herein. Enlarging the training set with incorrectly recognized data has proved efficient, but should be used with parsimony. Very small images seem to pose a higher degree of difficulty for classification, as they were more often misclassified.

The test set used here attempted to be representative of the universe of images of interest and incorporated two other test sets developed by Steven Simske and Mark Shaw that proved effective in the tuning of the current classifier. Every effort was made in the correct labeling of images and to avoid image duplication.

The new classification scheme provided here decreased the error rate by a factor of 3.2 (from 14.6% to 4.4%) while simultaneously improving performance by a factor of ten (from 1458 to 147 msec/image processing time) compared to the current scheme based on [7]. The increased accuracy improves the appearance of the printed output while the greatly improved performance frees up computing resources for additional printing tasks.

5. References

[1] H.Frigui and R.Krishnapuram. Clustering by competitive agglomeration. *P. Recognition*, 30(7), 2001.

[2] M.A.Hearst and J.O.Pedersen. Reexamining the Cluster Hypotesis: Scatter Gathet on Retrieval Results, SIGIR, 1996.

[3] S.Krishnamachari and M.Abdel-Mottaleb. Image Browsing using Hierarchical Clustering, IEEE Symposium on Computers and Communications, ISCC'99, July 99.

[4] P.Scheunders. Comparison of Clustering Algorithms Applied to Color Image Quantization, Patt. Recog. Letters, v18(11-13):1379-1384, 1997.

[5] G.Park, Y.Baek and L.Heung-Kyu. A Ranking Algorithm Using Dynamic Clustering for Content-Based Image Retrieval. CIVR'2002, pp.328—337, LNCS 2383, Springer Verlag, 2002.

[6] K.Barnard and D.Forsyth. Learning the Semantics of Words and Pictures, Inter. Conf. C. Vision, 2001.

[7] S.J. Simske, "Low-resolution photo/drawing classification: metrics, method and archiving optimization," *Proceedings IEEE ICIP*, IEEE, Genoa, Italy, pp. 534-537, 2005.

[8] Weka 3: Data Mining Software in Java, website http://www.cs.waikato.ac.nz/ml/weka/.

[9] G.Pereira e Silva and R.D.Lins. PhotoDoc: A Toolbox for Processing Document Images Acquired Using Portable Digital Cameras. CBDAR´2007, pp.107-114, 2007.

[10] Lins, R.D. and D.S.A. Machado, A Comparative Study of File Formats for Image Storage and Trans., v13(1):175-183, Journal of Electronic Imaging, 2004.

[11] N. Otsu. "A threshold selection method from gray level histograms". IEEETrans.Syst.Man Cybern. v(9):62-66, 1979.

[12] L. Breiman, "Random Forests", Machine Learning, 45(1), pp. 5-32, 2001.