Automatically Detecting and Classifying Noises in Document Images

Rafael D. Lins
Gabriel Pereira e Silva
UFPE, Recife, Brazil
{rdl, gabriel.psilva}@ufpe.br

Serene Banerjee
Anjaneyulu Kuchibhotla
HP Labs, Bangalore, India
{serene.banerjee,anji}@hp.com

Marcelo Thielo
HP Labs., Porto Alegre, Brazil
marcelo.resende.thielo@hp.com
{

ABSTRACT
Image filtering to remove noise in document images follows two different approaches. The first one uses human classification of the noise present in an image for identifying a noise filter to use. The second approach is to blindly apply a batch of filters to an image. The former approach, although widely used, may insert noise in the filtering process due to the incorrect classification of the noise. This project aims at doing a more accurate and computationally efficient document cleanup by pre-characterizing the noise that is present in the document based on a set of human labeled training samples. The current focus of the project is on pre-characterization of the following types of noise: back-to-front interference or bleed through, skew and orientation, and frames.

Categories and Subject Descriptors
I.4.9 [**Image Processing and Computer Vision**]: Applications.

General Terms
Measurement, Documentation, Performance.

Keywords
Noise characterization, documents, borders, skew, back-to-front interference, bleeding, show-through, orientation, classification.

1. INTRODUCTION
Finding ways to classify images and grouping them in sets of similar features has been researched by the database community for almost three decades aiming to make efficient information retrieval in image databases [1][2]. In such systems one image, known as a *query 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 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 achieved greater success in image retrieval was the analysis and clustering by using colour histograms **Error! Reference source not found.**[5]. The semantics of images have also been used as a clustering method **Error! Reference source not found.** 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 is completely uncorrelated should exhibit very different properties. Recent works [7][8] address the problem of image classification from the perspective of a printer that is fed with a raster file and classifies its content as belonging to one of the four clusters: photo, logo, document and complex. Such classification allows the printer to load color enhancement filters which yield better printing quality. Along the same research line of image classification, reference [9] attempts to identify the digitalization device of a given document deciding whether a document image was scanned or photographed. The photographed documents were processed using PhotoDoc [10], a tool developed for processing document images acquired with portable digital cameras. The latter group of images is further split into images acquired with and without the strobe flash on.

Document cleanup is important while scanning and copying documents. Current approaches for document cleanup typically are either based on human noise detection and filter selection or is performed automatically. They try to focus on certain types of cleanups and perform the filtering "blindly" on documents. The problem is that the latter strategy leads to inefficiency of document cleanup and, more importantly, could yield to degradation of the document image if the type of noise being cleaned up does not exist in the document or does not match the strength of the filter. If one could determine the type of noise and could do more intelligent document cleanup on the document it could enhance both the efficiency and the quality of the cleanup. Noise recognition, classification, understanding of its nature and strength is fundamental for suitable noise removal.

Noise characterization and even classification is a relatively new area of research [11]. The important point is to determine which features of a document should be used for noise characterization in a very efficient manner. Given a database of document images with ground truths that have been manually labeled indicating the type of noise, can one then determine the classification of the noise for a document the system has not seen before given that:

- The document may have no noise (i.e. no noise should also be one class)
- The document may have more than one type of noise.

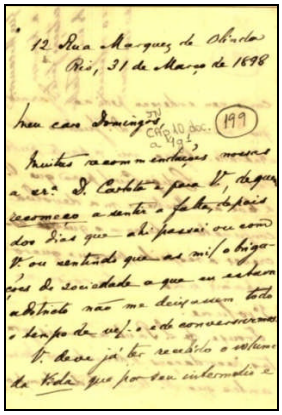
The classifier reported in this work is able to classify the existence of noise in a given document into the following five categories:

- Back-to-front interference (or bleed or show through)
- Frame or border noise
- Skew

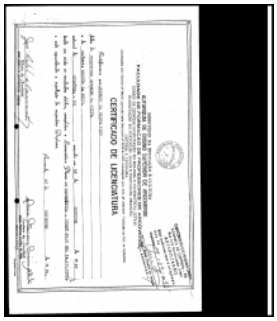
SAC'10, March 22-26, 2010, Sierre, Switzerland.

- Orientation (0, 90, 180, 270 degrees)
- No noise

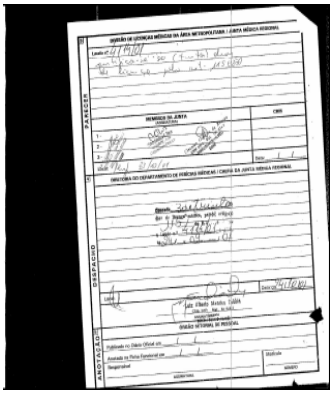
Figure 01 presents some sample images with the different kinds of noise classified here. As one may observe, one often finds more than one kind of noise per image.



Back-to-front interference



Frame and orientation noises



Skew and black frame

Orientation

Figure 1 – Document images with noises of interest

2. CLASSIFICATION STRATEGY

The classifier developed herein works in parallel for the detection of the different clusters of noises. In the case of back-to-front interference the overall classifier is the result of cascading three classifiers that split the noise into strong, medium and light interference. The architecture of the classifier is shown in Figure 2. The classifier used is Random Forest [14] which was implemented in Weka [12], an open source tool for statistical analysis developed at the University of Waikato, New Zealand. A number of features were extracted from each image to allow classification. The details of the training and test sets are provided in Table 1.

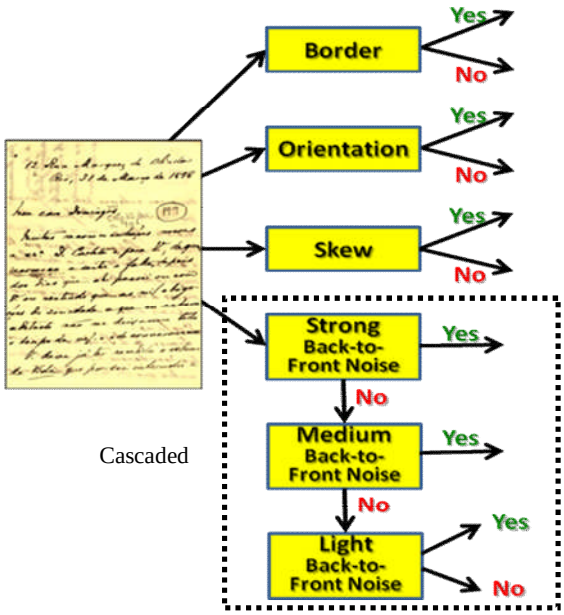


Figure 2 – Classifier architecture

Skew	**Noise**	**No_Noise**
Synthetic	6,200	8,000
Original	3,800	2,000
Tiff (BW)	3,000	2,400
Tiff (gray)	3,000	3,000
png (color)	3,000	3,600
Jpg (color)	1,000	1,000
Orientation	**Noise**	**No_Noise**
Synthetic	6,200	8,000
Original	3,800	2,000
Tiff (BW)	3,000	2,400
Tiff (gray)	3,000	3,000
png (color)	3,000	3,600
Jpg (color)	1,000	1,000
Border	**Noise**	**No_Noise**
Synthetic	5,200	7,000
Original	5,346	2,011
Tiff (BW)	500	2,011
Tiff (gray)	2,600	5,000
png (color)	2,000	1,000
Jpg (color)	946	1,000
Back-to-Front	**Noise**	**No_Noise**
Synthetic	--------	---------
Original	2,027	3,000
Tiff (BW)	--------	---------
Tiff (gray)	--------	---------
png (color)	--------	---------
Jpg (color)	2,027	3,000

Table 1 – 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, keeping in mind that quality matters more than size. Table 03 presents the relative size of the training and test sets.

	Test	Training	%
Skew	20,000	1,600	**8.00**
Orientation	20,000	1,600	**8.00**
Border	19,557	1,651	**8.44**
Back-to-Front	5,027	510	**10.14**
Total	**64,584**	**5,351**	**8.80**

Table 2 – Sizes of Training x Test sets

2.1 Sub-sampling

Very often classifiers do not use the whole original image for classification, as their feature extraction is a time intensive task. The larger the image file, the richer it is in data redundancy. Thus, if the redundant data is thrown away, the efficiency both in time and classification increases. 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!). The cascaded sub-sampler presented in reference [8] was used here. It performs the following operations:

size = height*width

- If size ≤ 300,000 break;
- If 300,000< size ≤ 500,000: remove even lines or columns (whatever the larger);
- If 500,000 < size ≤ 700,000: remove even lines and columns;
- If 700,000 < size ≤ 900,000:
  remove 2 lines in every 3 lines and even columns, (if height>width)
  remove even lines and 2 columns in every 3 columns, otherwise;
- If 900,000 < size remove 2 lines and 2 columns in every 3 lines / columns;

Code for the "cascaded" sub-sampler

2.2 Classification features

The choice of the features to be extracted from each image is of paramount importance to the success of the classifier. The following set of features, based on the classifier described in reference [8], was chosen:

- 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)
- Shannon´s entropy on three different regions of the document shown in Figure 3.

Image binarization is performed by using Otsu [11] algorithm. 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 it is a color image). 8-bits size is either the size of the original image if in grey 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. The features above are extracted for each sub-sampled image and placed in a vector of features.

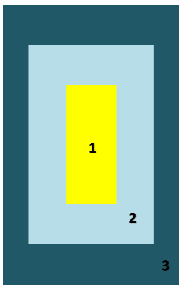


Figure 3 – Areas of interest for entropy calculation

3. Results

This section presents the results of noise classification of the images in the test set for each of the noise classifiers. It is worth observing that the classifiers act in parallel as shown in Figure 2. Thus given an image the different noises may be observed simultaneously.

3.1 Border Noise Detection

The images in the test set presented borders of all kinds:

- White borders,
- Black borders (uniform, irregular, etc.),
- Textured black borders.

Figure 4 presents examples of the different kinds of border noise.

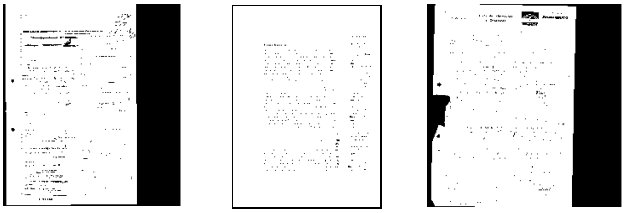


Figure 3 – Different kinds of borders (from left-to-right: solid black, white, and textured noisy border in a thorn off document).

The results of classification for the border noise detection classifier are presented by the confusion matrix presented in Table 3. One should stress that all images were sub-sampled for reasons of increasing the time efficiency of the feature extractor and classifier.

Border Noise	With	Without	Ratio
With	9,514	1,032	90.2142
Without	1,425	7,586	84.1859

Table 3 – Confusion matrix of the **border noise** classifier with sub-sampled images

Although the results obtained for detecting the border noise shown in the confusion matrix above are quite reasonable, this classifier may be improved by being broken into parallel sub-classifiers for each of the kinds of borders surrounding the documents above.

3.2 Skew Detection

Skew noise is often found in the digitalization of large quantity of documents overall when performed by automatic feed scanners. In

order to test the presence of skew noise in document images, the images considered as having original skew (inserted by the digitalization process) have rotation angles of less than 3 degrees in over 96% of cases. The remaining ones have rotation angles of less than 5 degrees. The synthetic images were generated by rotating straight-up images 2, 5 and 10 degrees. Some of the images were of handwritten documents such as the one on the top left hand corner of Figure 1. This although considered as being a non-skewed document, has a visible skew in the handwritten lines and so poses extra difficulty to the classifier. The confusion matrix for classifying images with skew is shown in Table 4.

Skew	With	Without	Ratio
With	9,671	329	96.71
Without	198	9,802	98.02

Table 4 – Confusion matrix of the **skew noise** classifier with sub-sampled images

As one may observe from the results presented in Table 4 the classifier correctly detected most of the images. From the 329 images with skew that were classified as being without skew, there are 100 synthetic images of historical documents that, the rotation imposed compensated the skew in the handwriting.

3.3 Wrong Orientation Detection

The mass digitalization of batches of documents very often includes incorrectly placed ones, either upside down or sideways. The images included in the test set with original orientation noise are the result of such accidental misplacement in a large batch of documents from a real-world digitalization bureau. The synthetic documents were obtained by rotating the images 90, 180, and 270 degrees. The straight-up documents are documents whose orientation is correct.

Orientation	Misplaced	Straight-up	Ratio
Misplaced	9548	452	95.48
Straight-up	200	9,800	98.00

Table 5 – Confusion matrix of the **orientation noise** classifier with sub-sampled images

The classification results shown in Table 5 present a high accuracy reaching over 96% of the documents analyzed. The detection of upside-down documents is responsible for most of the incorrect orientations found in the classifier results, as one would expect.

3.4 Back-to-Front Noise Detection

Back-to-front noise, also known as bleeding or show-through depending on its strength may make document binarization impossible. As most OCR software takes input as a binary image, this fact has as a consequence that documents with such a noise cannot be automatically transcribed. Researchers [15][16] have pointed out that no algorithm in the literature is good enough to remove bleeding noise in all sorts of documents. Depending on the strength of the noise, some algorithms may perform better than others. Unfortunately, the back-to-front noise appears more often in the digitalization of documents than one may assume to start with. The test set of documents we used with show-though had 2,027 real-world documents (no synthetic ones) which were obtained either from historical files (such as the one shown in the top-left hand corner of Figure 1) or from the scanning of printed proceedings of technical events. Images were hand labeled according to four levels of interference as: strong, medium, light and none. The classifiers for this noise were cascaded, as shown in Figure 2. The strong-classifier was trained with the images tagged as strong in the training set, against all the remaining images (Medium-Light-None) from the training set. Similarly, the medium-classifier was trained with the images labeled as medium, against the others with a lighter or no interference. The classification results obtained are shown in Table 6.

Back-to-front	Strong	Medium	Light	None	Ratio
Strong	1,073	65	3	1	93.95
Medium	91	638	15	19	83.61
Light	5	9	96	12	76.22
None	24	53	106	2,817	93.90

Table 6 – Confusion matrix of the **back-to-front noise** classifier with sub-sampled images

The analysis of the data obtained shows that the classifier was able to detect the back-to-front noise in 90.97% of the noisy images and also to classify t93.90% of the noise-less images correctly. It is also worth mentioning that the misclassification of the images without noise was in the direction that they had a light back-to front interference. If one takes into account that such images were in JPEG format and that the background of many documents was not solid white, but also encompassed other noises due to aging, stains, etc, the results obtained are quite reasonable.

One should also note that the noisy documents, whenever misclassified, tend to be placed in the group immediately below. For instance 91 of the documents labeled as having strong bleeding noise were classified as having a medium noise, an acceptable result as the tagging followed no quantitative criteria. The adoption of synthetic noisy images could be of some help in solving the aforementioned problems, but their generation is far from being a simple task as it involves not only the overlapping of two images, one of which is faded. The image in the background also presents some degree of blur and this scenario gets complicated further in the case of the simulation of aged documents, a situation very often found whenever dealing with historical documents.

4. TIME PERFORMANCE

This section presents the time performance of the feature extractor and classifier, which used as hardware platform a machine running a processor Intel(R) Core(TM)2 Duo CPU E7400 @ 2.80GHz, with 4,00 GB RAM. The feature extractor and the sub-sampler were implemented in C++. Together, they take 93 ms per image, on an average.

Table 7 presents details of the Random Forest [14]classifier time performance, which was implemented in Weka [12], using Java as an implementation language.

Classification	**Number of Trees**	**Time (ms)**	**Language**

Skew	10	3.1	Java
Orientation	10	3.1	Java
Board	10	3.1	Java
Back-to-front interference	30	5.9	Java
Total time per image		27	

Table 7 – Weka Random Forest classifier time performance

As shown in Table 7 the overall classification time per image is 27 ms. This time can be made smaller by a careful re-coding of the classifier in a lower level language such as C++.

5. DISCUSSION AND CONCLUSIONS

The automatic detection of noise in document images allows for better document filtering and enhancement. The classifier proposed herein presented a performance standard that is reliable enough to free humans of the burden of choosing which filters to use to remove the noises studied: border, skew, orientation and back-to-front interference. Besides that, it also helps avoid document degradation by blindly processing document images through a bank of filters. It is also important to mention that although the classifier takes 120 ms (93 ms for feature extraction and sub-sampling plus 27 ms for classification) to be able to decide about the presence of the four studied noises in an image this time is far less than what would be needed to run the image through the unnecessary filters.

Weka [8] has provided an excellent testbed for statistical analysis. The choice for a Random tree classifier was made after performing several experiments with a large number of alternatives offered by Weka, including a MLP neural classifier although results did not vary widely.

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. The test set used here attempted to be representative of the universe of images of interest. Every effort was made to ensure correct labeling of images and to avoid image duplication.

The extension of the classifier to recognize blurred images is in progress.

3. ACKNOWLEDGMENTS

The research reported herein was funded by….

4. 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] R.D. Lins; G.F.P. Silva; S.J. Simske; J. Fan; M. Shaw; P. Sá; M Thielo. Image Classification to Improve Printing Quality of Mixed-Type Documents. In: International Conference on Document Analysis and Recognition, 2009, Barcelona. Proceedings of ICDAR 2009. New York : IEEE Press, 2009. p. 1106-1110.

[9] G.F.P. Silva; R.D. Lins; B. Miro; S.J. Simske; M. Thielo. Scanned or Photographed? Automatically Deciding How a Document was Digitized. International Workshop on Camera-Based Document Analysis and Recognition, p. 1-10, IAPR Press, 2009.

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

[11] R.D. Lins. A Taxonomy for Noise Detection in Images of Paper Documents - The Physical Noises. International Conference on Image Analysis and Recognition, LNCS 5627, pp 844-854. Springer Verlag, 2009.

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

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

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

[15] R.D Lins; J.M.M. Silva; F.M.J. Martins. Detailing a Quantitative Method for Assessing Algorithms to Remove Back-to-Front Interference in Documents. Journal of Universal Computer Science, v. 14, pp. 299-313, 2008.

[16] P. Stathis; E. Kavallieratou; N. Papamarkos. An Evaluation Technique for Binarization Algorithms. Journal of Universal Computer Science, v. 14, pp. 3011-3030, 2008.