A DOCUMENT PAGE CLASSIFICATION ALGORITHM IN COPY PIPELINE

X. Dong†, P. Majewicz‡, G. McNutt‡, C. Bouman†, J. Allebach†, and I. Pollak†

† Purdue University
School of ECE
West Lafayette, IN 47907

‡ Hewlett-Packard Company
Boise, ID 83714

ABSTRACT

This paper describes a real-time, strip-based, low-complexity document page classification algorithm, which can be used as a copy mode selector in the copy pipeline. The benefits of such a copy mode selector include improving copy quality, simplifying user interaction, and increasing copy rate.

Index Terms— document classification, page classification.

1. INTRODUCTION

The general workflow of a digital copier is to take scanned images from its scanner, process these images, and send them to its printer for the physical reproduction. A copier must be able to process many different kinds of originals. These originals may have different types of content, such as text, line art, graphics, and natural photos; they may be printed on different kinds of media, for example, on papers of various levels of quality and brightness; they may be created using different rendering techniques such as halftone or continuous tone. These different kinds of originals may interact in many different ways with various limitations of copy pipeline such as streaks, stray light, color fringing, drift, gamut, moiré, etc. Fixed settings of the copy pipeline would therefore produce varying levels of reproduction quality, depending on the type of the original. To resolve this issue and generate user-preferred reproductions, different configurations of the copy pipeline are required. We call these different configurations "copy modes".

As illustrated in Fig. 1, a much better reproduction is obtained when the original matches its optimal configuration. Consider a photograph and a fax. The optimal configuration for a photo (i.e., the photo mode) has smoothing and a wide tone curve range to achieve noise reduction and color accuracy. The optimal configuration for a fax (i.e., the text mode) has sharpening and a nearly bilevel curve to achieve an enhanced reproduction. The left panel of Fig. 1 is the original, and the other two panels are the results of processing in the photo and text modes, respectively. Users prefer the cleaner background, increased contrast, and sharpness of the text mode and do not like the over-smoothed, low-density text and the dirty background in the photo mode. Therefore it is essential to match the originals with the most appropriate copy modes.

Three methods exist to match the originals to the copy modes. The "institutional copier" method prompts the user to describe attributes of the original to determine a matching mode. It requires a trained user. The "defaults to a single mode" method yields poor quality for originals that do not match that single mode. The "sub-menus" method provides users the opportunities of selecting modes through optional sub-menus. Users mostly ignore or are unware of such options. None of the above three methods satisfy user interaction and copy quality expectations.

Fig. 1: Processing a fax in photo and text modes. Left to right: original, photo mode, text mode.

In this paper, we propose a novel document page classification algorithm which can be used as a copy mode selector in the copy pipeline. Many existing algorithms segment the page and label each segmented region with an appropriate class label [1–5]. These algorithms require simultaneous access to the entire page image, and visit each pixel multiple times. On the other hand, any algorithm applicable to the copy pipeline must process image data one strip at a time, and never revisit previously processed strips. This makes it impossible to apply the existing approaches [1–5]. Our proposed algorithm operates on strip-based data and makes one pass over each strip. The computational complexity and memory requirements of our algorithm are both very low. These advantages make our algorithm an ideal candidate to be implemented in the existing copy pipeline without altering the hardware.

In the rest of the paper, we describe and illustrate our algorithm: Section 2 provides its general overview; Section 3 describes its specifics; Section 4 contains experimental results.

2. ALGORITHM OVERVIEW

Figure 2 illustrates a possible implementation of our classification algorithm. The classifier resides between the front end and the back end of the pipeline, and directs the image data to the most appropriate processing mode. The revised copy pipeline not only improves copy quality and simplifies user interaction, but also significantly increases copy rate in at least one case, as follows. The pipeline reconfigures for each page in a multi-page copy job from an automatic document feeder

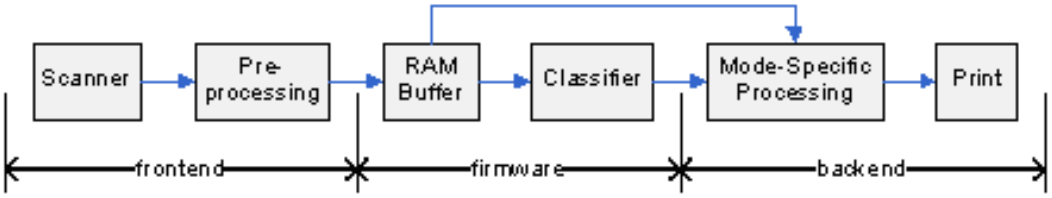

Fig. 2: One possible implementation of our page classification algorithm in the existing copy pipeline.

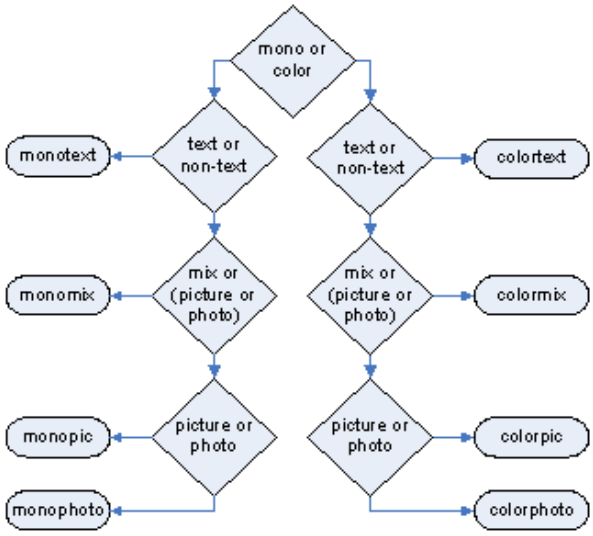

Fig. 3: Classification decision tree.

(ADF). When it detects a black-and-white page inside the job, it implements 3-to-1 channel reduction and enters the faster mono pipeline mode.

We work with a specific copy pipeline equipped with eight different copy modes which are all possible combinations of mono/color and text/mix/picture/photo. Mono mode is a configuration specially optimized for mono originals while color mode is optimized for color originals. Text mode is optimized for text, line arts, graphics, handwritten text, and faxes; picture mode is for high dynamic range halftoned originals such as glossy magazine pages; photo mode is for continuous tone natural scenes on photographic paper; mix mode is for all other types of originals. Our goal is to classify the scanned image of the original into the following eight distinct classes: color-text, color-mix, color-picture, color-photo, mono-text, mono-mix, mono-picture, and mono-photo. For simplicity, we just use the names of the copy modes as the names of candidate classes. Color-mix is the default class when the classification is hard to make.

We follow a tree-like decision structure illustrated in Figure 3. A mono vs. color classifier is placed at the root level of the decision tree. Then text vs. nontext classifiers, text/mix vs. picture/photo classifiers, and picture vs. photo classifiers are deployed at the second, third, and fourth level of the decision tree. Note that starting from the second level of the decision tree, the two classifiers on the same level share the same methodology and the only difference between them is some tunable parameters. In the following section, we discuss each of these classifiers.

3. ALGORITHM DETAILS

3.1. Color vs. Mono Classifier

We assume that all image data are from 300 dot-per-inch (DPI) scans, gamma-corrected, and represented in RGB color space with 8 bits per channel.

The essence of our color vs. mono classification algorithm is detecting image patches which are close to gray and declaring an image to be mono only if every patch is close to gray. Basing the decision on patches that are too small would not be robust to noise; on the other hand, small color regions may be missed if the patches are too large. We have experimentally found that partitioning an image into 32×32 blocks works well.

The specifics of our algorithm are as follows. If $R(p) = G(p) = B(p)$ for a pixel p, we say the pixel is gray, where $R(p)$, $G(p)$, and $B(p)$ are the pixel's red, green, and blue intensities, respectively. We define the *colorfulness* C(p) of a pixel p as the following quantity:

$$C(p) = \max[R(p), G(p), B(p)] - \min[R(p), G(p), B(p)].$$

Note that if $C(p) = 0$ the pixel is gray. Note also that the quantity $C(p)$ may be interpreted as the "Manhattan distance" from the point $(R(p), G(p), B(p))$ to the line $R = G = B$ in the RGB space.

The colorfulness $C(b)$ of a block of pixels b is defined as the sum of $C(p)$ over all the pixels p that belong to the block b. We partition the image into 32×32 blocks and define the colorfulness of the image, C_{image} as the maximum of $C(b)$ over all 32×32 blocks b. We classify an image as color if C_{image} is larger than a threshold, and we classify it as mono otherwise. The threshold is determined from training data.

3.2. Text vs. Nontext Classifier

Here nontext refers to all classes other than text, i.e., mix, picture, and photo. We use two properties of text documents in order to distinguish them from documents that contain pictures or photos in addition to (or instead of) text. First, the histogram for a typical text region has peaks that are more narrow and tall than the peaks in a typical picture or photo histogram. For example, the histogram for a patch of sharp mono text would have two large and narrow peaks, one near white and one near black. Second, nontext areas of a text document typically contain only

a few colors. (If, in addition to text, a document contains many colors, we would want to classify it as mix.)

We extract *histogram peakiness* features via the following procedure. We partition the image into 8×64 blocks and calculate 64-bin R, G, and B histograms for each block. The k-span of a histogram is defined as the largest number of consecutive bins in the histogram whose values are greater than or equal to k. The k-span of an image is defined as the largest k-span over all 64-bin R, G, and B histograms of its 8×64 blocks. For each image, we form a ten-dimensional feature vector consisting of k-spans of the image for $k = 3, 6, \ldots, 30$.

This feature extraction procedure is used to calculate the *histogram flatness score* for any image as follows. We estimate the means of text and nontext feature vectors as well as a common covariance matrix, based on labeled training data. For any image whose feature vector is $\mathbf{x}$, we then define the histogram flatness score

$$F = (\mathbf{m}_{nontext} - \mathbf{m}_{text})^T \Sigma_F \mathbf{x},$$

where $m_{nontext}$ and m_{text} stand for the mean of the nontext and text feature vectors, respectively, Σ_F stands for the common inverse covariance matrix of the text and nontext feature vectors, and T denotes the transpose of a vector. A small histogram flatness score implies peakiness of the histogram and suggests that the image is text, whereas a large histogram flatness score indicates that the image is nontext.

In addition to the histogram flatness score, we compute the *color variability score* which is small if there are only a few colors in the nontext regions of the image, and large if there are many colors. In order to identify nontext regions we define a *text edge* to be three consecutive pixels in either horizontal or vertical direction such that their values are monotonically increasing or decreasing and the absolute value of the difference between the first and the third exceeds a threshold. We take the threshold to be 100. A *nontext region* is any region that does not contain any text edges.

The computation of the color variability score starts with partitioning the image into 8×8 blocks, detecting nontext blocks, and computing the mean R, G, and B values for every nontext block. We then construct 256-bin R, G, and B histograms of these mean values taken from all nontext blocks and count the largest number of nonzero bins among these three histograms. This number is our color variability score. If it is low, every histogram has only a few nonzero entries, suggesting that non-text regions in the image are cartoon-like. If this score is high, there is at least one histogram with many nonzero bins, indicating the presence of many colors in the image. In this case, copying in the text mode would be inappropriate.

Finally, in order to classify an image, we set thresholds for both the histogram flatness score and the color variability score, based on the training data. We classify an image as text if both scores are less than their corresponding thresholds. Otherwise, we classify it as nontext, i.e., mix, picture, or photo.

3.3. Text/Mix vs. Picture/Photo Classifier

Photos and pictures contain natural scenes and no text. Our strategy for distinguishing such images from mix and text documents is to detect text and other regions that do not look like natural scenes.

We partition the image into 64×64 blocks and count the number of text edges in each block. We use the same definition for a text edge as in the previous section. The text edge count for an image is then defined as the maximum text edge count among all the 64×64 blocks. A large text edge count suggests that the image may be in the mix or text category.

In addition, we reuse histograms of the R, G, and B averages of 8×8 nontext blocks which, as described in the previous section, have been computed for the text vs. nontext classification task. As in the previous section, we extract the number of nonzero bins for each of the three histograms. In addition, we extract the k-spans for each of the three histograms, for three values of k: $k = M/8, M/4, M/2$, where M is the maximum of the histogram over its first 230 bins. These twelve features (the number of nonzero bins and the three k-spans for each of the three histograms) form a feature vector. We estimate the means of the feature vectors for the two classes from the training data, and we also estimate a common covariance matrix. For any image whose feature vector is $\mathbf{y}$, we then define the *unnaturalness score*:

$$U = (\mathbf{m}_{mix/text} - \mathbf{m}_{photo/pic})^T \Sigma_U \mathbf{y},$$

where $\mathbf{m}_{photo/pic}$ and $\mathbf{m}_{mix/text}$ are the mean vectors for the two classes, and Σ_U is the common inverse covariance matrix. The larger the score U, the more likely it is that the image is not a photo or a picture.

Finally, in order to classify an image, we set thresholds for both the edge count score and the unnaturalness score, based on the training data. We classify the image as photo or picture if both scores are less than their corresponding thresholds. Otherwise, we classify it as mix or text.

3.4. Picture vs. Photo Classifier

Pictures typically contain halftone noise. Smooth regions that are near midtone are most affected by the halftone noise. We use these regions to distinguish between a picture and a photo.

We partition an image into 8×8 blocks and define a block b to be a midtone block if at least one of its mean R, G, and B values is within some interval centered at 128. In our experiments, we take this interval to be $[80, 176]$. The *roughness* $S(b)$ of a block b is then defined as follows. For a block that is not a midtone block, $S(b)$ is infinite. For a midtone block, $S(b)$ is the minimum of $S_R(b)$, $S_G(b)$, and $S_B(b)$ where $S_R(b)$ is the sum of the absolute values of the first differences in the horizontal and vertical directions within the block b for the red channel, and $S_G(b)$ and $S_B(b)$ are defined analogously for the G and B channels, respectively. We finally define the roughness of the image as the minimum $S(b)$ over all the blocks b.

We classify an image as photo if its roughness is below a threshold, and we classify it as picture otherwise. The threshold

	Classification results							
ground-truth	color-text	color-mix	color-picture	color-photo	mono-text	mono-mix	mono-picture	mono-photo
color-text	60%	40%	-	-	-	-	-	-
color-mix	-	99%	**1%**	-	-	-	-	-
color-picture	-	66%	34%	-	-	-	-	-
color-photo	-	29%	-	71%	-	-	-	-
mono-text	18%	4%	-	-	61%	17%	-	-
mono-mix	**1%**	11%	-	-	**2%**	83%	**3%**	-
mono-picture	**1%**	4%	1%	-	-	66%	29%	-
mono-photo	-	1%	-	-	-	41%	-	58%

Table 1: Classification rates for the training suite. Harmful misclassifications are indicated in boldface.

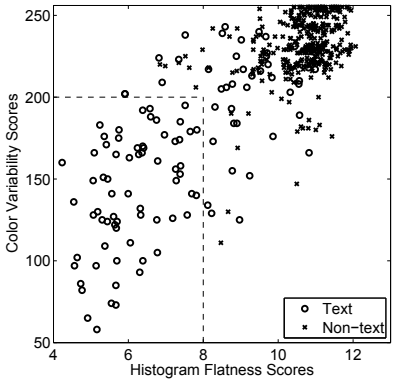

Fig. 4: Color text vs. nontext: final feature decision map. The decision boundary (dashed lines) results in no harmful misclassifications of nontext into text.

is determined from training data.

4. EXPERIMENT RESULTS

We have a training suite containing 891 images with at least 100 images for each class. The ground-truth labels of all images in the training suite are selected by hand. Since not all errors are equally costly, some are considered benign and others harmful. We accordingly select the decision thresholds discussed in Section 3. For example, mislabeling a mono image as color is acceptable whereas mislabeling a color image as mono is not. By conservatively setting the decision thresholds, we are able to correctly identify all color images in our dataset as color images.

Another example is illustrated in Figure 4. This is a 2-D feature decision plot of color-text vs. color-nontext classification. The x-axis is the histogram flatness score and the y-axis is the color variability score. All color nontext images are shown in "X"s and color text images are shown in "O"s. Recall that we classify a color image as color-text if both its histogram flatness score and its color variability score are less than the corresponding thresholds. Color-text misclassified as color-nontext is considered benign and the opposite way of misclassification is considered harmful. Therefore we choose two conservative thresholds: 8 for the histogram flatness scores and 200 for the color variability scores. The thresholds are shown by the dashed lines in Figure 4. Note that there are no "X"s in the lower-left rectangle defined by the dashed lines—i.e., there are no harmful misclassifications of nontext into text.

Table 1 summarizes the final classification results for all eight modes. It contains the empirical conditional probabilities P(classification result | ground truth) for the training suite. Depending on the ground truth, accuracy ranges from 29% to 99%. These numbers are shown in the main diagonal of Table 1. We consider the accuracy to be satisfactory because only very few images falls into harmful modes. These harmful misclassifications are indicated in boldface numbers in Table 1. All other misclassification are considered benign.

5. CONCLUSIONS

We have introduced a real-time, strip-based, low-complexity document page classification algorithm and used it as the mode selector in the copy pipeline. The revised pipeline improves copy quality, simplifies user interactions, and increases the copy rate. The classification algorithm analyzes the scan image and classifies it into one of eight classes in the copy pipeline. Modes are the combinations of mono/color and text/mix/picture/photo. Mode classification is 29% to 99% accurate with misclassifications tending towards benign modes.

6. REFERENCES

[1] H. Cheng and C.A. Bouman. Document compression using rate-distortion optimized segmentation. *J. Elec. Im.,* 10(2):460-474, Apr. 2001.

[2] R. de Queiroz. Compression of compound documents. *Proc. ICIP*, vol. 1, pg. 209-213, Kobe, Japan, Oct. 1999.

[3] S. Revankar and Z. Fan. Picture, graphics and text classification of document image regions. *Proc. SPIE*, vol. 4300, pg. 224-228, 2001.

[4] S. Simske and S. Baggs. Digital capture for automated scanner workflows. *Proc. 2004 ACM Symposium on Document Engineering* pg. 171-177, Milwaukee, WI, 2004.

[5] W. Wang, I. Pollak, T.-S. Wong, C.A. Bouman, M.P. Harper, and J.M. Siskind. Hierarchical Stochastic Image Grammars for Classification and Segmentation. *IEEE Trans. Im. Proc.*, 15(10):3033-3052, Oct. 2006.