Document Image Content Inventories

Henry S. Baird, Michael A. Moll, Chang An, Matthew R. Casey

Computer Science & Engineering Department
Lehigh University, 19 Memorial Drive West
Bethlehem, Pennsylvania 18017 USA

E-mail: baird@cse.lehigh.edu
URL: www.cse.lehigh.edu/~baird

ABSTRACT

We report an investigation into strategies, algorithms, and software tools for *document image content extraction and inventory*, that is, the location and measurement of regions containing handwriting, machine-printed text, photographs, blank space, etc. We have developed automatically trainable methods, adaptable to many kinds of documents represented as bilevel, greylevel, or color images, that offer a wide range of useful tradeoffs of speed versus accuracy using methods for exact and approximate k-Nearest Neighbor classification. We have adopted a policy of classifying each pixel (rather than regions) by content type: we discuss the motivation and engineering implications of this choice. We describe experiments on a wide variety of document-image and content types, and discuss performance in detail in terms of classification speed, per-pixel classification accuracy, per-page inventory accuracy, and subjective quality of page segmentation. These show that even modest per-pixel classification accuracies (of, *e.g.*, 60–70%) support usefully high recall and precision rates (of, *e.g.*, 80–90%) for retrieval queries of document collections seeking pages that contain a given minimum fraction of a certain type of content.

Keywords: *document content extraction, content inventory, Bayes decision theory, classification, k Nearest Neighbors, k-d trees, CART, spatial data structures, computational geometry, hashing, versatility*

1. INTRODUCTION

We report on an investigation of algorithms for **document image content extraction**:

> *Given* an image of a document,
> *find* regions containing machine-printed text, handwriting, photographs, graphics, line-art, logos, noise, etc.

We approach this problem in its full generality, attempting to cope with the richest diversity of document, image, and content types that occur. We have reported preliminary results in the development of highly versatile[1] and voracious[2,3] classifiers for this problem domain. Types of document images that we accept include color, grey-level, and bilevel (black-and-white); also, any size or resolution (digitizing spatial sampling rate); and in any of a wide range of file formats (TIFF, JPEG, PNG, etc). We convert all image file formats into a PNG file in the HSL (Hue, Saturation, and Luminance) color space; bilevel and greylevel images convert to HSL images with fixed values for hue and saturation. We have gathered a database of over 9000 sample page images containing the following types of content: machine print (MP), handwriting (HW), photographs (PH), line Art (LA), math notation (MT), maps (MA), engineering drawings (ED), chemical drawings (CD), "junk" (JK, e.g. margin and gutter noise), and blank (BL). We are also gathering samples of each content type across a wide range of languages and image qualities and from several historical periods.

We have adopted the policy of classifying individual *pixels*, not *regions* as most previous R&D projects have done. This avoids the arbitrariness and restrictiveness of limited families of region shapes, as illustrated in Figures 1 and 2.

In all examples shown, each test image is on the left with the results of classification next to it on the right as a *classification image* where the content classes are show in color: machine print (MP) in dark blue, handwriting (HW) in red, photographs (PH) in green or blue-green, blank (BL) in white, and unclassified in light grey. Each image possesses a thin border of unclassified pixels (difficult to see at this resolution) due the fact that feature extraction requires a region of a minimum size. Some other pixels remain unclassified due to sparsity of training data in the k-D tree.

[As published in *Proc., IS&T/SPIE Document Recognition & Retrieval XIV Conf.* (DRR2007), San Jose, California, January 28 – February 1, 2007. Figures 3 and 4 have been corrected and reformatted to conform to a conventional confusion-matrix format.]

Figure 1. A color image of a magazine page, before classification (left) and after classification (right). Our policy of classifying pixels has the advantage of adapting to arbitrary layouts (here, regions with circular boundaries). Classification of text embedded in regions of constant color is sensitive to type size. The pink coloration in the background is due to a scattering of isolated blank pixels misclassified as handwriting. The per-pixel classification accuracy is 52.9%.

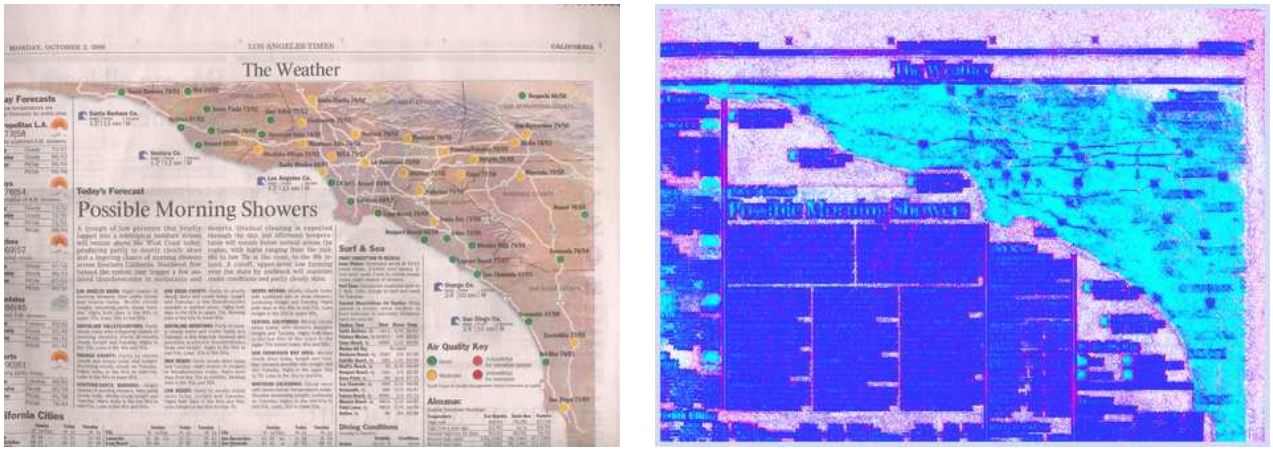

Figure 2. A color image of a newspaper weather page, before classification (left) and after classification (right). Here, in addition to a complex nonrectilinear layout, machine-print text is interspersed with small photographs and other non-textual artwork. The subjective segmentation of tightly spaced blocks of small text is remarkably good given the low print quality of the newspaper. The per-pixel classification accuracy is 60.8%.

This flexibility has another advantage, as we will show: it allows greater accuracy in *inventory* statistics, by which we mean summaries of each page estimating, for each content class, the fraction of page area dominated by that class.

Thus both training and test datasets consist of pixels labeled with their ground-truth class (one of MP, HW, PH, BL, etc). Each pixel datum is represented by scalar features extracted by image processing of a small region centered on that pixel; these features are discussed in detail in Section 3.

2. CLASSIFICATION ALGORITHMS

We are investigating a wide range of automatically trainable classification technologies, including brute-force k-Nearest Neighbors (kNN), fast approximate kNN using hashed k-d trees, classification and regression trees, and locality-sensitive hashing.

2.1. Brute-force k-Nearest Neighbors

We have implemented 5NN under the Infinity Norm using a brute-force algorithm. We regard this as our "gold standard" and compare other faster but usually less accurate methods to it.

2.2. Hashed k-D Tree Classifier

We have previously reported our non-adaptive k-D tree classifier using fixed cuts, sped up by hashing bit-interleaved addresses[2],[3] which runs up to several hundred times faster than brute-force 5NN with only a small loss in accuracy in this domain. The experimental results described here were achieved using this classifier, hashing 24 bits of bit-interleaved address. We also sped it up by a technique of "inverted classification" ("filtering" in[3]), in which test data are read first and hashed into the k-D tree; as the training data is read, data that hashes to an empty cell (*i.e.* one that contains no test data) can be discarded, while those that hash into occupied cells are of course used to "annotate" the relevant testing points with their class and distance (each testing point owning a list of up to k nearest neighbors so far). The principal advantage of this technique is that it allows us to constrain memory usage to $O(m)$, where m is the testing set size, with no sacrifice in accuracy and with the same computational cost (measured in numbers of distance computations). As test and training sets grow, inverted classification scales well since the test set can, with little or no loss in accuracy, be split into separate test sets as needed to maintain memory footprints small enough to avoid thrashing.

Since inverted classification allowed us to avoid thrashing, observed runtime was roughly proportional to the number of distance computations performed. For example, given a testing set of 3.3 million samples and a training set of 35,247 samples* a brute-force kNN classifier would perform over 110 billion computations, whereas the hashing classifier performed only 7.5 billion, a speed-up of a factor of 15.5. This allowed the classifier to run to completion in 47 CPU minutes, permitting frequent experiments which allowed a more thorough investigation of effective combinations of features. These results were typical of our experiments with the hashing inverted classifier.

2.3. Classification and Regression Trees

We have implemented one variety of Classification and Regression Trees (CARTs).[4] In preliminary trials, accuracy is comparable to Hashed K-D Trees, and classification speed is somewhat faster, but training is orders of magnitude slower, restricting the size of data sets we can experiment with.

2.4. Locality-Sensitive Hashing

We have also implemented a Locality-Sensitive Hashing classifier[5].[6] This is another fast approximation to k-NN. The key idea is to "hash the points using several hash functions so as to ensure that, for each function, the probability of collision is much higher for objects which are close to each other than for those which are far apart".[7] The method enjoys, with high confidence, sublinear time complexity as a function of the size of the training set, but it is exponential in the number of dimensions. Indyk *et al* report large speedups over various tree-search methods on specially devised training sets. It was originally designed for (single) nearest neighbor queries, so we are working to extend it to approximate kNN.

3. CHOOSING THE FEATURE SET

Each pixel (the "target pixel") is represented by scalar features extracted by image processing of a small region centered on that pixel. We have investigated more than 60 features, all extracted from the luminosity channel (ignoring the hue and saturation channels): we selected twenty-six of these for the experiments reported here, for reasons summarized below. All feature values are scaled to lie within the (convenient but otherwise arbitrary) integer range 0-255.

*This training set is small due to the decimation described in Section 4.2.

3.1. Average Region Luminosity

A group of four features: the average luminosity values of NxN-subregions centered on the target pixel (for N=1,3,9,27). The algorithm makes five successive passes. In the first pass, it simply copies all the pixel luminosity values in an array. In the second through fourth passes, it calculates the sum of the luminosity values needed for the successively larger boxes by taking the sums of the values held in the smaller boxes (so 9 smaller boxes are added to create an larger one, at each pass). A final pass outputs the values to the feature array, dividing the values in each of the boxes by the number of pixels summed in that box.

On small, relatively specialized, training and test datasets, these features discriminated handwriting from machine print well, but their effectiveness lessened as the training set grew and diversified. Unsurprisingly perhaps, the larger the NxN region the less discriminating they were.

3.2. Region Luminosity Difference

A group of sixteen features: each is the difference in total luminosity between halves of NxN regions cut in four directions: horizontal, vertical, and the two diagonals.

These are effective in discriminating between BL (blank) and other content classes, with the (still somewhat mysterious) exception of HW (handwriting).

The following five groups of features extract features from straight lines of pixels centered on the target pixel, at each of the four directions. The length of these lines (in pixels) is an essential parameter of course: we'll give specifics of these choices at the end of this section.

3.3. Average Line Luminosity

The average of luminosity values along the line.

These assist in discriminating between handwriting and machine print. However, the diagonal features proved less effective than the horizontal and vertical features and were discarded.

3.4. Line Luminosity Average Difference

The average of absolute differences of luminosity between adjacent pairs of pixels along the line.

The diagonal variants of these proved to be effective especially in combination with the average line luminosity features. But the horizontal and vertical variants were less effective and were discarded,

3.5. Line Luminosity Max Difference

The maximum among absolute differences in luminosity between each pair of adjacent pixels along the line.

These are effective especially in combination with 3.3 and 3.4: particularly, they help discriminate BL (blank) from other classes. We use all of these.

3.6. Distance to Max-difference Pair

The distance from the target pixel to the closest pair of pixels that possess a maximum luminosity difference.

3.7. Distance to Max-difference Pixel

The distance from the target pixel to the closest one with a maximum absolute luminosity difference with the target pixel.

Early experiments suggested that the two groups of features (immediately above) were not helpful. But when we revised them (as described below), they improved....

3.8. Revised Distance to Max-difference Pair

These are features 3.6 computed in eight directions radiating out from the target pixel (rather than in four directions centered on it).

3.9. Revised Distance to Max-difference Pixel

These are features 3.7 computed in eight directions radiating out from the target pixel.

These two groups 3.8 and 3.9 proved not to be effective unless they were used together: then they were revealed to be the best features for discriminating between PH (photographs) and the other classes.

3.10. Difference Between Two Distances

These are differences between corresponding features 3.8 and 3.9. They did not assist classification (and in fact increased the error rate). We tried other ways to combine features 3.8 and 3.9, including encoding the luminosity max-difference into the distance by multiplication: but we saw no improvement.

3.11. Feature Combination

Having tested many (but, of course, not all possible) combinations and variations of the features described above, we gradually converged on the following twenty-six:

Region luminosity average: 1x1 region;

Line luminosity average: horizontal and vertical, line-length 25 pixels;

Line average difference: line-length 25;

Line luminosity average difference: diagonals only; line-length 25;

Line luminosity max difference: four directions, line-length 41;

Revised distance to max-difference pair: eight directions, line-length 41; and

Revised distance to max-difference pixel: eight directions, line-length 41.

The search for better sets of features is an open-ended engineering exploration. We still hope to identify a smaller yet more effective set. We are presently implementing a family of Haar features.

4. EXPERIMENTAL DESIGN

We have experimented with two data sets of page images: a development set (A) containing 28 images; and a benchmarking set (B) containing 117 images. For data set (A), thirteen images were placed in the training set, and the rest in the test set; then set (A) was used to drive our choice of twenty-six features. For data set (B), 31 images were placed in the training set, and the rest in the test set; then set (B) was used to train and test classifiers using these features; the results of these tests are reported here. Together the two sets contain MP, HW, PH, and BL content. Their text includes English, Arabic and Chinese characters each represented by bilevel, greylevel, and color examples. The selection of test and training pages was random except that for each test image there was at least one similar, but not identical, training image. Thus these experiments test the discriminating power of the features and weak generalization (to similar data) of the classifiers, but they do not test strong generalization to substantially different cases.

Each content type was zoned manually (using closely cropped isothetic rectangles) and the zones were ground-truthed. The training data was decimated randomly by selecting only one out of every 3000th training sample. This policy was suggested by experiments described later in Section 4.2.

We evaluated performance in three ways:

Per-pixel accuracy: the fraction of all pixels in the document image that are correctly classified: that is, whose class label matches the class specified by the ground truth labels of the zones. Unclassified pixels are counted as incorrect. This is an objective and quantitative measure, but it is somewhat arbitrary due to the variety of ways that content can be zoned. Some content—notably handwriting—often cannot be described by rectangular zones. This in some cases will lead to a per-pixel accuracy score being worse than an image may subjectively appear to be. However, this metric does provide a simple generalization of how well the classifier is performing: for the test set for data set B, the average per-pixel accuracy score was 62.4%. The complete confusion matrix is given in Table 3.

Our classifier is best at recognizing MP and PH, has some difficulty with BL, and has a lot of trouble with HW, misclassifying 43% of HW pixels as BL. An analysis of the raw data, given in Table 4, reveals that, in spite of the high overall error rate, each content type is represented by a roughly similar number of pixels whether by ground truth or as a result of classification. As mentioned before, we do not necessarily expect the per-pixel accuracy score to be extremely high due to arbitrariness and even inconsistency in zoning. However, it seems to be reasonable to expect zoning to reflect the overall amount of each content type found in an image, and we hope the classifier will do the same.

It can be instructive to analyze the raw pixel-count data for a particular page image (shown in Figure 1), as given in Table 5.

This image had an overall per-pixel accuracy score of 52.9%, which might suggest that classifier performance was mediocre. However, the raw pixel-counts reveal that a large number of pixels that were zoned as machine print

	BL	HW	MP	PH	Type1
BL	0.159	0.028	0.032	0.005	0.065
HW	0.028	0.023	0.014	0.001	0.043
MP	0.046	0.029	0.353	0.039	0.114
PH	0.023	0.007	0.046	0.167	0.076
Type2	0.097	0.064	0.092	0.045	0.298

Figure 3. Confusion matrix for per-pixel classification of the test set of data set B, which contained 178 million test pixels (40M BL; 12M HW; 83M MP; and 43M PH). The rows label ground truth content types; the columns label the content types assigned by the classifier. The Type1 column entries summarize error rates for each true class (that is, the frequency with which that true class is misclassified). The Type2 row entries summarize error rates for the classes resulting from classification (that is, the frequency with which that classifier decision is incorrect). The bottom right entry gives the overall error rate: 29.8%.

	BL	HW	MP	PH	Type1
BL	28385810	4992050	5678089	963871	11634010
HW	5054560	4128702	2422925	228225	7705710
MP	8150037	5196000	63137187	6968964	20315001
PH	4080809	1321661	8310500	29773773	13712970
Type2	17285406	11509711	16411514	8161060	53367691

Figure 4. Table of raw pixel counts for per-pixel classification of the test set of data set B. The rows label ground truth content types; columns label the content types assigned by the classifier. The Type1 column gives totals of erroneously classified samples for each true class. The Type2 row gives totals of erroneously classified samples for each classifier-reported class. The bottom right entry gives the total number of erroneously classified samples, out of a total of 178793163 samples.

	BL	HW	MP	PH	Type1
BL	160494	47200	5267	2869	55336
HW	0	0	0	0	0
MP	209381	146351	1011957	112532	468264
PH	28938	20577	177956	502363	227471
Type2	238319	214128	183223	115401	751071

Content	*True*	*Classifier*	*Accuracy*
BL	6.817	24.18	20.96
HW	0	11.3	0
MP	46.75	42.14	75.85
PH	23.06	22.38	70.91

Figure 5. Results on the page shown in Figure 1. For this image the per-pixel accuracy is 52.9%. The table on top gives the raw pixel-count classifier results (there are no HW pixels in the ground-truthed zoning). Derived from this is the table below which gives the page inventory—that is, for each *Content* class: the *True* fraction of its pixels classified as that class; the *Classifier*-reported fraction of that class; and the per-pixel *Accuracy* of the classifier on that class. Note that although the per-pixel accuracies on MP and PH are below 80%, the classifier-reported fraction is very close to the true fraction for both of them. In this way we have often seen that retrieval based on inventory scores is superior to per-pixel classification accuracy.

Threshold	Recall	Precision
0.0	1.0000	1.0000
0.1	1.0000	0.9104
0.2	0.9643	0.9152
0.3	0.8800	0.9362
0.4	0.8409	0.9250
0.5	0.7586	0.8148
0.6	0.6000	0.7500
0.7	0.5714	0.8000
0.8	0.6667	0.6667
0.9	—	0.0000
1.0	—	—

Figure 6. Recall and precision scores for the query "Find all pages with at least the fraction T of machine-print (MP) pixels," over a range of thresholds T from 0.0 to 1.0, on the test set of data set B. Values left blank reflect queries which do not return any images.

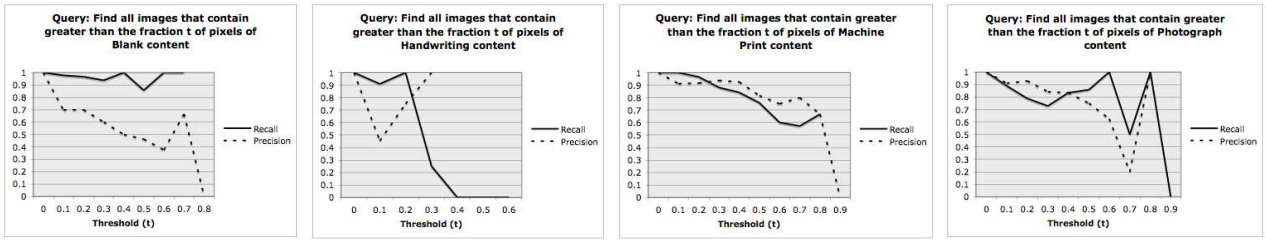

Figure 7. Precision and recall curves for blank (BL), handwriting (HW), machine print (MP), and photograph (PH) content classes, as measured on the test set of data set B. The x-axis represents the threshold value T and the y-axis the precision (or recall) scores for finding all images with at least the fraction T of each content class, compared to the ground truth results.

were misclassified approximately uniformly across the other three classes, which is not usually the case. Thus a better interpretation is that zoning on this page was problematic.

The raw pixel count data is useful for identifying exactly what type of errors the classifier is making relative to the ground truth. We see that BL, MP, and PH pixels are all classified correctly nearly 70% of the time. As mentioned before, we see that HW is being recognized very poorly, and actually more pixels are being classified as BL than HW. This also may point to problems with zoning methodology.

Per-page inventory accuracy: for each content class, we measure the the fraction of each page area that is classified as that class. That is, each page is assigned four numbers—one for each of BL, HW, MP, and PH—which sum to one. This description allows a user to query a data base of page images in a variety of natural and useful ways, For example, in an attempt to retrieve all page images with large photographs with captions, she might ask for all pages containing least 70% photograph and 10% machine print.

We have analyzed the performance of queries of this form: "find all images that contain at least the fraction T of pixels of content class C." This is of course an information retrieval problem for which precision and recall are natural measures of performance:[8–10] precision is the fraction of page images returned which are relevant; and recall is the fraction of relevant documents that are returned.

We issued queries, for every content class, over the full range of threshold values, and summarized the results with precision and recall curves as a function of threshold. For example, the precision and recall scores for MP are shown in Table 6.

Thus the query "Find every image containing at least 30% machine print" can be answered with 88% recall and 94% precision.

The complete set of four recall and precision curves are shown plotted in Figure 7.

Each curve must of course start at 100% at threshold 0. As we saw from the confusion matrix before, performance on handwriting is poor. On blank we see a much higher recall rate; but precision is substantially lower and decreases steadily. This implies that more content is being classified as blank than was zoned. This conclusion is not surprising as we use rectangles in zoning which cannot accurately capture the layout of a page.

Both machine print and photographs, which appeared to perform best from analyzing the confusion matrix, also enjoy high precision and recall rates, which do not decline as quickly as the other classes. Interestingly, both curves for each class cross each other multiple times; and they are often not monotonic.

If we assume that all threshold values (from 0.0 through 1.0) are equally likely, we can compute expected recall and precision scores for each class:

	Recall	**Precision**
BL	0.967	0.556
HW	0.451	0.801
MP	0.809	0.772
PH	0.760	0.788

It is interesting that even at this early stage of development of these document inventory methods, MP and PH enjoy usefully high expected recall and precision, far higher than the per-pixel classification accuracy scores would suggest. This good performance persists up to a threshold of about 60%; the fall off after that can be attributed to the rarity of such images in the test set. Most images in the test set were of mixed content type and do not contain high percentages of any single content class.

Document content inventory precision and recall curves allow a richer and, arguably, more realistic analysis of performance than per-pixel recognition rates and confusion matrices. Given the confusion matrix for a content class, it seems to be possible to infer only crude generalizations of what the average precision and recall curves might look like for that class.

Furthermore, these per-page inventory scores seem to be significantly less sensitive to the arbitrariness of zoning methodologies than per-pixel accuracies are.

Subjective segmentation quality. This is a subjective assessment of the quality—expressed as 'good', 'fair', and 'poor'—of the classification as a guide for geometric segmentation of the page area into regions of different content classes. Eventually, automatic methods may be developed to convert a pixel-based classification into one of many possible region-based segmentations: this measure attempts to predict how well that could be done. For lack of space we will not attempt to quantify this here, but we will discuss a wide range of illustrative examples, in the next Section.

4.1. Discussion of Experimental Results

The next four Figures (8, 9, 10, and 11) show classification results for selected images from the development data set A.

Figure 8 shows results on two magazine pages. These illustrate one of the methodological problems with using comparison to ground-truth files as our evaluation metric, inasmuch as our ground-truth labels rectangular zones, while obviously the text blocks here are not laid out in rectangles. Arguably these classification results are in some ways better than the per-pixel accuracy scores suggest: we feel justified in assigning a "good" rating for its subjective segmentation quality.

Figure 9 shows results on three pages containing handwriting, in bilevel, greylevel, and color images. The classifier locates the handwriting in detail, not merely approximated as rectangles (as the ground-truth zoning does). In the last example the lines of the legal pad are misclassified as machine print, suggesting that better features are needed.

Figure 10 shows results on clippings from two newspaper pages annotated with handwriting. In both samples the handwriting blocks are indicated by a higher concentration of pink, but the classification remains ambiguous since some pixels are classified as machine print. Note that, in the photograph in the first image, the text on the football player's jersey is correctly identified as machine print.

Figure 11 shows three magazine-article pages with more complex layouts. Foreground text occurs on backgrounds of contrasting color, and there is both black-on-white and white-on-black text. The layouts are partly non-rectangular, with photographs intruding past the vertical margins of text areas. This illustrates a strength of our policy of classifying pixels rather than regions.

Figures 12, 13, and 14 show classification results from data set B. The number of sources and images used in the training set were roughly doubled from data set A. As a result of some of these engineering decisions, performance on things like machine print and photographs appears to have improved, while performance on blank space has become noticeably worse.

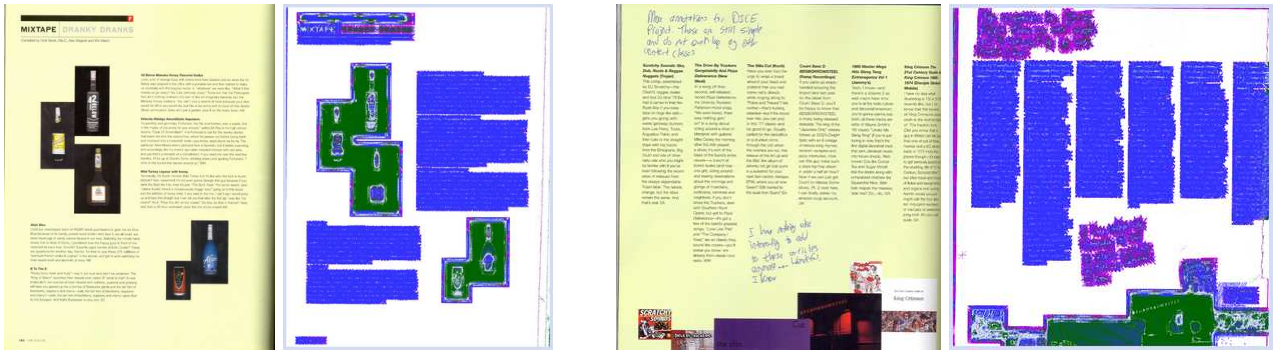

Figure 8. Test page images containing relatively simple magazine article layout. The first image contains blank (shown as white), machine print (blue), and photograph (green) content and the second also contains handwriting annotations (pink). In the first image 86.72% of the pixels were correctly classified and 70.4% in the second. We subjectively rank the segmentation quality as good except perhaps for a fraction—less than 20%—of the photograph regions.

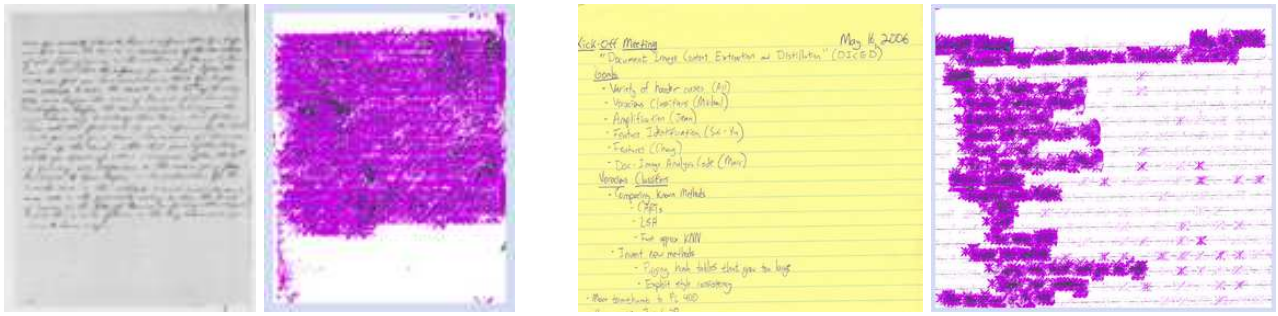

Figure 9. Test page images containing handwriting. From left to right their per-pixel accuracies are 79.0%, 64.0% and 70.1%. We subjectively rank the segmentation quality as good with the exception of a small region in the first image near the left bottom margin.

Figure 10. Test page images of newspaper clippings containing machine print annotated with handwriting, both from the same publishere but of different image qualities. From left to right their per-pixel accuracies are 52.9% and 71.3%. The segmentation quality of machine print is generally good, but there are confusions with handwriting especially within areas of large print. The segmentation quality of handwriting areas is also good.

Figure 11. Test pages images containing complex magazine-article layouts. Per-pixel accuracies are, left to right: 66.9%, 61.1%, and 58.8%. Body-text machine print segmentation quality is generally good. Most of the photographs are largely classified correctly. Confusions between machine print and photographs are not rare, but on close inspection (difficult in this Proceedings) many are arguably right. A few areas are spuriously classified as handwriting.

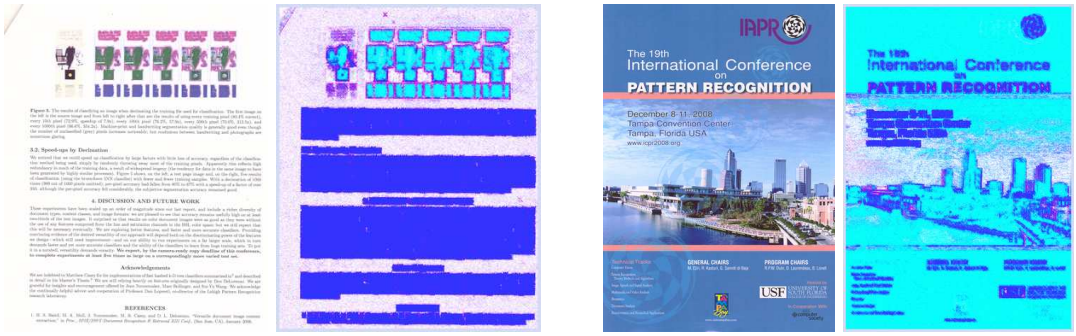

Figure 12. The page on the left illustrates excellent subjective segmentation accuracy for MP and PH, but also the high rate of confusion between BL and HW. The page on the right illustrates a methodological quandary in the zoning of large uniformly pale regions in photographs (here, the sky behind a cityscape) which are statistically indistinguishable from blank areas in other pages: still, the classifier performs fairly well. Per-pixel accuracies: left 70.8%, right 52.5%.

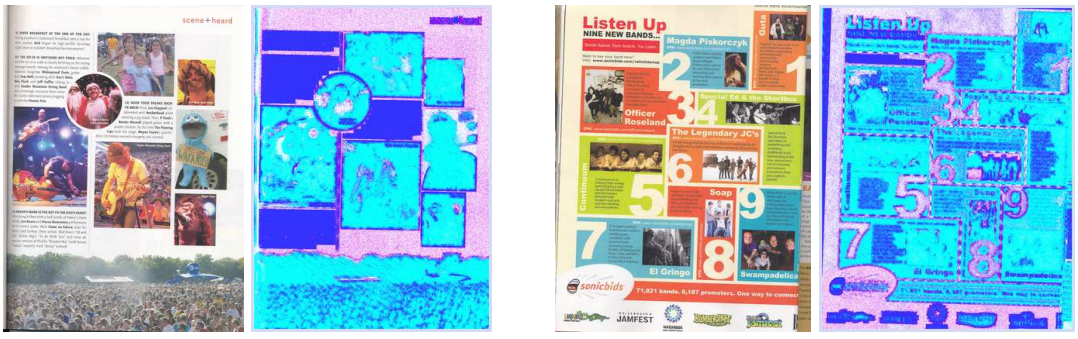

Figure 13. The page on the left illustrates success on a difficult case where the blank region that forms the background of most of the page bleeds imperceptibly into a photograph at the bottom. The page on the right illustrates an unusually complex nonrectilinear layout: when this is studied closely, it reveals many cases of successful discrimination between machine print and photographs on small irregularly shaped regions. Per-pixel accuracies: left 69.3%, right 35.0%.

Figure 14. The magazine page image on the left is remarkably well segmented: almost all text, including text embedded within photographs, is correctly labeled. The page on the right illustrates success on related, challenging cases: text overprinted on regions of arbirary solid color. Per-pixel accuracies: left 70.9%, right 73.8%.

Figure 15. The results of classifying an image when decimating the training file used for classification. The first image on the left is the source image and from left to right after that are the results of using every training pixel (80.4% correct), every 10th pixel (72.9%, speedup of 7.9x), every 100th pixel (76.2%, 57.9x), every 500th pixel (70.0%, 212.5x), and every 1000th pixel (66.6%, 354.2x). Machine-print and handwriting segmentation quality is generally good even though the number of unclassified (grey) pixels increases noticeably; but confusions between handwriting and photographs are sometimes glaring.

4.2. Speed-ups by Decimation

We noticed that we could speed up classification by large factors with little loss of accuracy, regardless of the classification method being used, simply by randomly throwing away most of the training pixels. Apparently this reflects high redundancy in much of the training data, a result of widespread isogeny (the tendency for data in the same image to have been generated by highly similar processes). Figure 15 shows, on the left, a test page image and, on the right, five results of classification (using the brute-force 5NN classifier) with fewer and fewer training samples. With a decimation of 1000 times (999 out of 1000 pixels omitted), per-pixel accuracy had fallen from 80% to 67% with a speed-up of a factor of over 350: although the per-pixel accuracy fell considerably, the subjective segmentation accuracy remained good.

5. DISCUSSION AND FUTURE WORK

Experiments have been scaled up by an order of magnitude since our last report, and include a richer diversity of document types, content classes, and image formats: we are pleased to see that per-pixel accuracy remains usefully high on at least two-thirds of the test images, and we are happy and surprised to see that our classifiers already support usefully high recall and precision performance on natural queries. We remain surprised and intrigued by the fact that results on color document images remain good without the use of any features computed from the hue and saturation channels in the HSL color space; we suspect that this will be necessary eventually. We are of course exploring better features, and faster and

more accurate classifiers. Providing convincing evidence of the desired versatility of our approach will depend both on the discriminating power of the features we design—which still need improvement—and on our ability to run experiments on a far larger scale, which in turn demands faster and yet more accurate classifiers and the ability of the classifiers to learn from huge training sets. We are increasingly dependent on large-scale GRID computing to drive development of these algorithms. To put it in a nutshell: versatility demands voracity.

Acknowledgements

Matthew Casey's implementations of fast hashed k-D tree classifiers are summarized in[2] and described in detail in his Master's Thesis.[3] We are grateful for insights and encouragement offered by Jean Nonnemaker, Marc Bollinger, and Sui-Yu Wang. We acknowledge the continually helpful advice and cooperation of Professor Dan Lopresti, co-director of the Lehigh Pattern Recognition Research laboratory.

REFERENCES

1. H. S. Baird, M. A. Moll, J. Nonnemaker, M. R. Casey, and D. L. Delorenzo, "Versatile document image content extraction," in *Proc., SPIE/IS&T Document Recognition & Retrieval XIII Conf.*, (San Jose, CA), January 2006.
2. M. R. Casey and H. S. Baird, "Towards versatile document analysis systems," in *Proceedings., 7th IAPR Document Analysis Workshop (DAS'06)*, (Nelson, New Zealand), February 2006.
3. M. R. Casey, *Fast Approximate Nearest Neighbors*, Computer Science & Engineering Dept, Lehigh University, Bethlehem, Pennsylvania, May 2006. M.S. Thesis; PDF available at `www.cse.lehigh.edu/~baird/students.html`.
4. L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*, Wadsworth& Brooks/Cole, Pacific Grove, CA, 1984.
5. M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, *Locality-Sensitive Hashing using Stable Distributions*, ch. 4. MIT Press, 2007.
6. P. Indyk and R. Motwani, "Approximate nearest neighbors: Towards removing the curse of dimensionality," in *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pp. 604–613, ACM, (New York), 1998.
7. M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions," in *Proc., 20th Annual ACM Symposium Computational Geometry*, pp. 253–262, ACM Press, 2004.
8. J. F. Cullen, J. J. Hull, and P. E. Hart, "Document image database retrieval and browsing using texture analysis," in *Proc., Int'l Conf. on Document Analysis and Recognition (ICDAR97)*, pp. 718–721, August 1997.
9. D. Doermann, "The indexing and retrieval of document images: A survey," *Computer Vision and Image Understanding* **70**, June 1998. Special Issue on "Document Image Understanding and Retrieval," J. Kanai and H. S. Baird (Eds.).
10. H. S. Baird and F. Chen, "Document image retrieval," *Information Retrieval journal (Special Issue)* **2**, May 2000.