

Recent Work in the Document Image Decoding Group at Xerox PARC

Thomas M. Breuel and Kris Papat

Xerox Palo Alto Research Center

3333 Coyote Hill Road

Palo Alto, CA 94304

1 Overview

Speed Enhancements to DID (Section 2)

When Document Image Decoding (DID) was proposed [15], its attractiveness lay primarily in its potential for high recognition accuracy, owing to its communications-theoretic framework, and well defined models and objective function (posterior probability). In its initial implementations it suffered from high computational cost relative to commercial OCR methods. We will summarize recent progress made on reducing its computational cost. Importantly, these speed enhancements do not come at the expense of accuracy; they are guaranteed to result in the same recognition output as DID without the enhancements.

DID with Language Models (Section 3) Until recently, DID achieved its high recognition accuracy without the benefit of linguistic knowledge. Recent work on the incorporation of linguistic knowledge in DID's search procedure will be described.

Grayscale DID (Section 4) The document image decoding framework is quite general, but for computational reasons previous work has focused primarily on binary images. The emergence of alternative image acquisition devices motivates its extension to grayscale. We consider one approach and present preliminary results.

Layout Analysis (Section 5) Layout analysis infers document structure from the arrangement of text and graphical elements in documents and uses that structure for higher-level tasks like matching, segmentation-by-example, layout based retrieval, and document indexing. Many existing systems attempt to find a single representation of document layout prior to solving the high level task. We have developed an approach to document layout analysis that is based on explicitly exploring the space of segmentation parameters as part of the overall segmentation task. In particular, the approach considers all

geometrically dissimilar layouts in tasks like layout-based retrieval and segmentation-by-example. This new approach promises to make segmentation-by-example and layout-based retrieval systems considerably more robust than previous approaches.

OCR By Clustering (Section 6) We have re-examined a well-known technique in OCR, recognition by clustering followed by cryptanalysis, from a Bayesian perspective. The advantage of such techniques is that they are font-independent, but they appear not to have offered competitive performance with other pattern recognition techniques in the past. Our analysis suggests a novel approach to OCR that is based on modeling the sample distribution as a mixture of Gaussians. Results suggest that such an approach may combine the advantages of cluster-based OCR with the performance of traditional classification algorithms.

Classification by Probabilistic Clustering (Section 7) Extending and generalizing our prior work on OCR by clustering, we have developed novel methods for probabilistic clustering. These methods promise to make OCR more robust to font variations and novel document degradation conditions, and they also have applications in other classification problems where the distribution of test samples may differ from the distribution of training samples. We describe some experiments demonstrating that the approach outperforms traditional classification methods in an OCR task.

2 Speed Enhancements to Document Image Decoding

Document image decoding involves searching a trellis for a best path that explains the observed text image, where the nodes in the trellis correspond to locations in the image, and where the edges in the trellis are labeled with a score of matching a hypothesized character beginning at that location. At each node, the number of outgoing edges is equal to the

number of characters in the font, plus special white-space characters. The best-path search has traditionally been carried out using the Viterbi algorithm, a form of dynamic programming.

In the past, the computational cost of performing the best-path search was dominated by the computation of the match scores to be assigned to the trellis edges. Three recent innovations have reduced this cost.

First, the one-pass Viterbi search has been replaced by an iterative scheme which involves repeatedly finding a best path by Viterbi, but initially using inexpensively computed upper bounds on the match scores. On each iteration, any edges labeled with upper-bound scores along the path found are re-labeled with their (expensive) true values. Eventually, a path will be found in which all of the edge labels are the true match score values; since this path has beaten all other optimistically scored paths, it must be a truly highest-score path. The savings comes about because the vast majority of true scores need never be computed; only those determined to be promising on the basis of the upper-bound scores are kept alive. The specific upper bounds used initially in this approach were best-case matches determined from counts of foreground pixels in text-line columns. This general approach, dubbed the Iterated Complete Path (ICP) algorithm, traces its roots to work in separable Markov source modeling [13] and was extended to within-line decoding more recently [17].

Second, when ICP is used as described above, portions of the path found in one iteration are re-used without re-computation in the next iteration, when it can be determined that the boundary conditions of the path segment are such that the best path in that segment cannot change. Specifically, when it is noticed that the cumulative scores attached to the nodes in the current iteration differ from those in the previous iteration by a constant value that persists over a run of pixels exceeding the maximum character width, the best path will not differ in that segment from the one found on the previous iteration, until an edge is encountered that has been re-scored. Empirically, we have noticed that most of the path doesn't change during the vast majority of the ICP iterations, so the savings thus accrued is substantial. The technique of re-using path segments in this manner is referred to as *Incremental Viterbi* and is also described in [17].

A final speed enhancement results by modifying the upper-bound used in ICP to group multiple columns together, which amounts to horizontal sub-sampling. The best-case match is computed not for each column individually, but rather for two, three, or four columns in the aggregate. The efficacy of

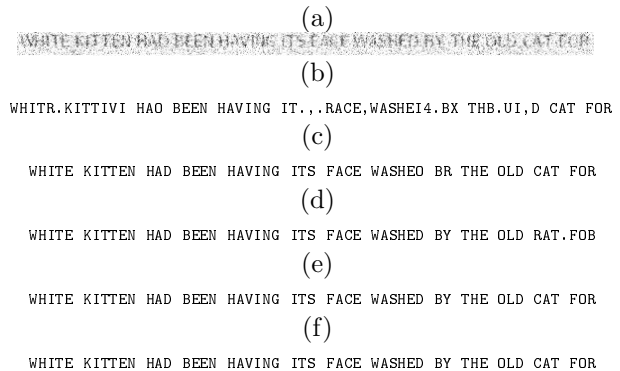


Figure 1: Example of the effect of integrating a language model into document image decoding, using several different strategies. (a) degraded, subsampled grayscale synthetic text line image; (b) decoding without a language model; (c) unigram language model via Viterbi; (d) Stack algorithm; (e) generalized ICP algorithm; (f) ground truth.

grouping multiple columns in this way depends on many factors, including the scan resolution. Details will be provided in a forthcoming publication.

Combined, the above enhancements have been found to improve the speed of DID by a factor of about forty on a small set of standard text images.

3 Document Image Decoding with Language Models

Until recently, DID had no mechanism to express prior preference for linguistically valid strings as recognized output over invalid strings. We have considered several approaches, settling on a class of approaches in which soft linguistic constraints are expressed by a sequentially predictive probability distribution over characters, conditioned on a fixed number of previous characters (typically four). This probability distribution is called a *language model*. Paths now have their edges scored with both a match component and a language model component. In principle, the trellis must be vastly expanded so that nodes can now encapsulate linguistic context in addition to position in the image. One approach is to think of the expanded trellis as a full tree, and apply an approximate search procedure to find a nearly-best path. The approximation comes about because of the practical necessity of avoiding searching the full tree of all possible messages. We have examined one such technique, the Stack algorithm, which is widely used in speech recognition [11] and in convolutional decoding [12], and found it to be promising [21].

We have also developed an iterative algorithm, much in the spirit of the ICP algorithm described in Section 2. We will refer to it here as a *general-*

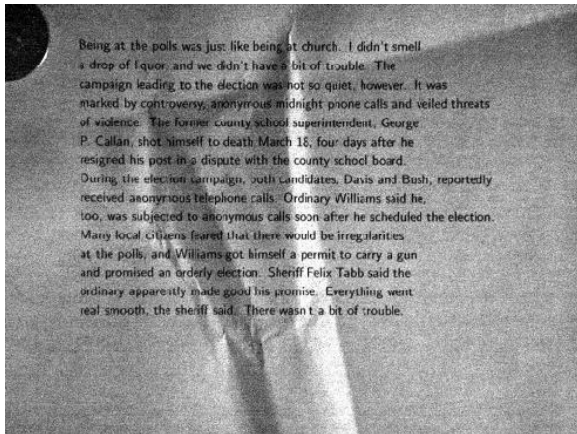


Figure 2: Deliberately wrinkled document image acquired in low-light conditions by a handheld digital camera, used to test the extension of DID to grayscale described in [19].

ized ICP algorithm. Rather than re-score edges on each iteration, nodes are added to encapsulate additional linguistic context along paths that are deemed promising, based on lower-order upper bounds on the language model scores. As the context approaches the full context exploitable by the language model, the upper bound scores approach and ultimately reach the true language model scores. When a path is found having only true language model scores on each edge, that path can be concluded to have the highest score among all paths, and the algorithm terminates. This algorithm has the advantage over the Stack algorithm and other approximate-search algorithms that it results in a true best path, but has the disadvantage that its computational complexity is strongly data-dependent. On the other hand, it can be set up to remember the best path seen so far, and to output that upon early termination. In other words, to limit computational complexity, it can be set up as an *any time* algorithm for approximate best-path search.

Figure 1 shows how language modeling in its various forms can influence DID recognition accuracy. The text line used in this example was severely corrupted by additive noise to make the error rate high enough so that the differences would be clear. The text line shown in (a) is for illustration and is actually *less* noisy than the one used for recognition, which is visually unintelligible. Both the Stack algorithm and generalized ICP yield high accuracy in this example. For more details on these approaches, see references [21] and [20].

4 Grayscale Document Image Decoding

The emergence of low-cost handheld digital cameras as a viable means of document image acquisition mo-

tivates the extension of the DID to function on relatively low-resolution grayscale images. Doing so involves significantly generalizing the channel model used by DID. One approach [19] involves carrying out the search in a high-resolution hypothesis image domain, and simulating the physical sampling and noise processes to match against the observed image. Initial results on a challenging test case are promising; the technique was found to perform favorably relative to the simple alternative of adaptive thresholding followed by application of a standard commercial OCR product. Figure 2 shows the test image used for this experiment. The edit distance between ground truth and the result of grayscale DID (with unit weighting for substitutions, insertions, and deletions) was seventy, versus ninety-one for binarization followed by commercial OCR. While preliminary, these results are felt to be encouraging.

5 Layout Analysis using the Document Scale Space¹

The layout of elements on a printed page conveys a wealth of information about a document. Much of the research on document layout analysis attempts to recover a single representation of the layout of a document. In many commercial OCR applications, the goal of representing document layout is to be able to recover the layout sufficiently well for making the document editable and presentable in a word processor or HTML. Another important application of layout analysis is in information extraction from documents for the purpose of document retrieval, as well as appearance based retrieval (reviewed in [6]).

A large number of different approaches to layout analysis have been described in the literature ([16] contains a list of references). Many systems perform a non-probabilistic bottom-up analysis based on the distances between connected components (e.g. [18]) or based on an analysis of the whitespace (e.g., [10]). Such systems generally require a number of numerical thresholds and parameters to be picked; e.g. thresholds at which characters are merged into lines, thresholds at which lines are merged into paragraphs, etc. Sometimes, these parameters are picked globally for the whole page image, but they can also depend on the local context.

More recently, Liang [16] has described a Bayesian approach that starts with similar primitives but computes a statistically optimal segmentation of the complete page, taking into account higher order constraints among layout elements. Another approach to document layout analysis assumes that there is a known, underlying logical and/or hierarchical model that describes the document (e.g., SGML, HTML,

¹This section is based on, and contains excerpts from, a paper presented at the DAS '2000 workshop[2].

or TeX source) and attempt to match such models against the physical representation of a document [23, 5]. When the underlying physical segmentation does not correspond well to the given logical model, conflicts are resolved in some cases using backtracking search.

This work proposes an approach to document layout analysis that differs in several ways from these other approaches. The key ideas are:

- The space of all possible physical segmentations is explored and represented efficiently and completely as a *document scale space*.
- The document scale space, rather than a single documentation, is used in layout matching tasks.
- The approach integrates the exploration of different segmentations directly into tasks like layout matching or segmentation by example.
- The approach is motivated using a Bayesian analysis of the layout matching problem.

Two applications for this work are *appearance-based retrieval* and *segmentation by example*. Retrieval of documents from document databases based on their physical or logical layout has been described, for example, by Doermann *et al.* [7]. The idea is to first perform a layout analysis of the documents in the database and the query document and then to compare the layouts for the purposes of retrieval. As we will see below, layout based retrieval can benefit significantly from incorporating the segmentation step directly into the layout matching process.

Appearance-based retrieval can also be used for segmentation by example tasks. The basic idea is to match an unsegmented query document against a database of manually segmented documents in a database. The segmentation of the best match found in the database can then be used to segment the query document. Segmentation-by-example tasks occur frequently in legacy conversions of company memos, patent documents, scientific journals, and medical data sheets. For such tasks, it would be desirable if unskilled users could indicate regions of interest on a few sample pages and the system could then use those samples to identify reliably corresponding regions in the document database.

5.1 Document Scale Space

To see how we can compute a document scale space efficiently, we need to look in more detail at how traditional document layout analysis methods work.

Single-Parameter Case A common approach to document layout analysis is based on choosing a threshold θ on the minimum Euclidean distance between connected components found in a document. Connected components that are closer to one another than the given threshold θ are grouped together into *layout components*. This thresholding operation partitions the set of connected components into a collection of disjoint sets. We call this collection of disjoint sets a *segmentation* or a *physical layout analysis* of the document.

Different thresholds θ give rise to different segmentations. If we look at the the different segmentations parameterized by θ , we obtain a structure similar to the scale space widely used in computer vision. We refer to this as a *single parameter document segmentation scale space*. It should be noted that, unlike in the computer vision case, there are only a finite number of distinct segmentations in the document segmentation scale space.

It is generally assumed that the hierarchy implied by a logical layout description of the page (page > column > paragraph > line > word) is paralleled in the document segmentation scale space. In practice, this assumption seems to be fairly well satisfied for some classes of documents, but in general, it clearly is not satisfied by many document layouts. Determining the actual threshold parameters to the different levels of logical layout themselves also involves some experimentation and heuristics. The methods described in this paper address both these problems.

Multiparameter Case We can extend the notion of a document segmentation scale space that we developed above to a case where thresholds and distances are evaluated differently in the x and y directions. In particular, for each pair of connected components, let us measure two distances, their horizontal distance and their vertical distance. We define the horizontal distance between two connected components to be infinity unless their vertical extent overlaps. If their vertical extents overlap, their distance is simply the distance between their bounding boxes. We can make an analogous definition for the vertical distance of two connected components. A segmentation is now given by picking two thresholds, θ_x on the horizontal distance and θ_y on the vertical distance. We can apply the same arguments as above and see that there are at most N different choices for each θ_x and θ_y , so there are at most N^2 different two parameter segmentations of the input.

5.2 Computing Document Scale Space

To compute and represent the document scale space, we use the following approach. First, the bound-

ing boxes of the connected components are stored in a trie data structure, which allows us to determine quickly the nearest neighbors of each connected component in the horizontal and vertical directions. Using this neighborhood information, we construct a graph, in which the connected components are the nodes and the nearest neighbor relationships define the edges. We can now take the set of all horizontal edges and all vertical edges and sort them by their distance.

For a particular choice of horizontal and vertical thresholds, we retain all edges in the neighborhood graph corresponding to distances less than the chosen thresholds. We then compute all the connected components of the neighborhood graph efficiently using a union-find algorithm.

To speed up this process further, we can use incremental updates of the data structures; that is, if we have computed the segmentation for given horizontal and vertical thresholds, if we increase either threshold, we do not have to restart the computation from scratch but update data structures incrementally. When applied to real document images, we obtain a few thousand distinct segmentations per document image in this way. The collection of these segmentations (together with their associated threshold parameters) is a complete representation of document scale space.

While fast enough for analyzing individual documents, when searching through a large document database, performance is proportional to the size of the document scale space. To improve performance further, we would like to come up with a more compact representation. We can achieve this by decimating the document scale space and retaining only representatives that are “substantially different”. If we consider segmentations that differ by less than 5% from one another (meaning, they have approximately the same overall structure and the areas of the different segmentation components differ by less than 5%), the number of distinct segmentation is reduced from 1000 or more to under 50 for most documents.

5.3 Applications

Based on these ideas, a prototype system was implemented that allows layout-based (appearance-based) retrieval of documents from the University of Washington Database 1. There is currently no widely used benchmark for document retrieval based on layout or physical appearance, but representative examples of queries and top matches are shown in Figure 3. Searches run at the speed of about 3.7 seconds per 1000 models on a IBM ThinkPad 600E (400MHz, RedHat Linux 6.1).

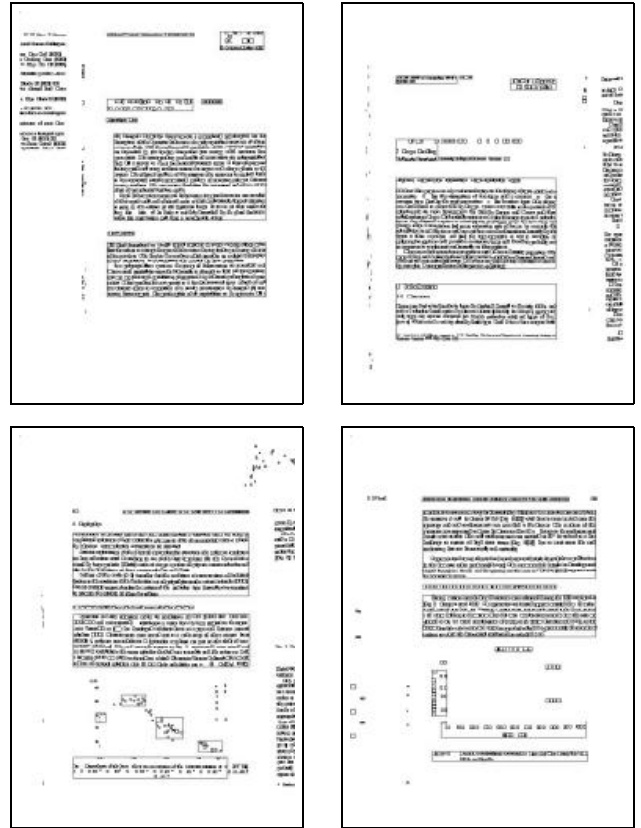


Figure 3: Layout-based retrieval.

5.4 Conclusions and Future Work

This work describes a number of ideas important for layout analysis:

- task-driven segmentation: the segmentation parameters are selected in an integrated way with the overall task (layout-based retrieval, segmentation by example, matching against a logical layout model, etc.);
- anisotropic, multi-parameter segmentations;
- document segmentation scale space;
- the use of representative sets of segmentations to speed up task-driven segmentation.

The paper motivated these ideas with geometric and Bayesian arguments.

These approaches help to address two fundamental problems in document analysis: that of unnecessary early commitment to a (possibly erroneous) bottom-up segmentation, and the dependence of many layout analysis methods on empirical, highly database dependent threshold parameters.

Further experiments need to be carried out to evaluate the performance of task-driven segmentation on layout-based retrieval and other tasks. However, the preliminary experiments presented in this

work suggest that the approach can be both efficient and gives reasonable results on a commonly used database of documents.

6 OCR By Clustering²

This work examines the effects of clustering character images prior to recognition in optical character recognition (OCR) of printed documents. This approach has a long history in OCR, and prior work has addressed the questions of how to build a clustered representation quickly [4], as well as how to label the resulting clusters. Clustering, mixture models, and mixture-based Bayesian recognition itself, of course, has a long history in statistics and pattern recognition. In this work, we make a connection between the two approaches. The key point is that the clustering of the character templates is, in effect, a mixture density estimation of the sample distribution. This connection allows us to reexamine issues of cluster validity, style adaptation [22], and cluster label assignment within a Bayesian framework.

6.1 The OCR Problem

For the purposes of this work, we will define the OCR problem in the following simplified manner. We assume that there is a fixed, finite set of characters (digits, lower case letters, upper case letters, special characters). Furthermore, we assume that there is an open-ended set of possible styles, where the notion of style encompasses character properties like font, size, and idiosyncracies of the particular rendering engine used. Picking a character and a style uniquely determines an idealized image (bitmap) for the character. During document creation, this idealized bitmap is printed on a piece of paper. When the document is scanned back in again, a degraded bitmap of the character, is obtained, usually by the addition of noise, blurring, thresholding, sampling error, and various forms of geometric distortions [14]. The core function of an OCR system is (roughly) to find the most likely character and style corresponding to such a degraded character image.

Traditionally, OCR systems perform this task by estimating posterior probabilities like $P(\text{char, style}|\text{bitmap})$, say, using a neural network or a Gaussian mixture model. However, estimating such probability distributions requires a large number of example characters. In practice, however, training data for many styles (fonts, degradation parameters) is not available at all.

²This section is based on, and contains excerpts from, a paper presented at SPIE '2001[1].

6.2 OCR by Solving a Cryptogram

An alternative approach proposed in the literature [4] is based on the idea of clustering similar character shapes and then assigning character labels to the resulting clusters. Such an approach is attractive because the clustering process itself is font independent, and cluster labels can, ideally, be assigned independent of the actual bitmap representation of the characters. This, on ideal data, such an approach is completely font independent and automatically generalizes to arbitrary unknown fonts. Clustering is also attractive because of the emergence of token-based compression methods that already represent documents as a collection of tokens. If we can carry out recognition directly on these clusters, we can perform OCR directly on token-compressed data.

However, in practice, such methods for carrying out OCR by clustering have not been very successful. The reason is, this paper argues, that the clustering methods used have modeled the actual statistical nature of the recognition problem poorly. This work describes how to begin combining the advantages of font independence of clustering OCR systems with the robustness of statistical methods used in current commercial OCR systems.

6.3 Gaussian Mixture Models

Let us assume, for the purpose of illustration, that each character image in the input document is represented by a feature vector \tilde{v} that is derived from a prototype feature vector $v_{c,s}$ representing character c and style s corrupted by an additive error G with zero mean and Gaussian distribution. In such a framework, the class conditional densities are $P(v|c,s)$ are then Gaussians. If we take a supervised pattern recognition approach, we estimate the class conditional densities (or, equivalently, priors and posteriors) from training data, derive discriminant functions, and use those to classify each unknown feature vector \tilde{v} in a Bayes-optimal sense as one of the different classes c, s .

The problem with this approach is that estimating the class conditional densities depends on a representative sample of degraded feature vectors over character classes c and styles s . If we do not have such a representative sample, our class conditional density estimates are going to be poor and recognition accuracy suffers.

We can, however, take a different approach using a partially unsupervised method involving the sample distribution. In a real OCR problem, we are usually given not a single character to classify, but many thousands of samples, one for each character in the document. Of course, these characters are not labeled, so we cannot derive the class conditional densities from this sample. However, what

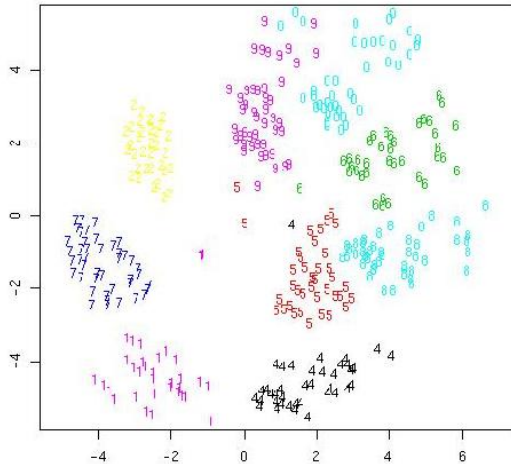


Figure 4: Low-dimensional representation of character feature vectors.

we can do is model the sample distribution, that is, the distribution of degraded feature vectors \tilde{v} , ignoring their class labels. If we assume that the class conditional densities are Gaussians, then the sample distribution is going to be a Gaussian mixture component (the frequency of the classes, c, s , are the mixture parameters). If we then try to recover the mixture components of this Gaussian mixture, we recover the individual class conditional distributions. We still do not necessarily know the classes corresponding to each such class conditional distribution, but we have a lot more information at our disposal to assign such labels than if we tried to classify characters one-by-one.

This idea is illustrated in Figure 4, which shows a two-dimensional representation (a Sammon mapping) of the feature vectors representing severely degraded samples of digits from a single font. In the figure, class assignments of the different samples are indicated by colors and labels. Looking at clusters with this information corresponds to estimating the class conditional density given labeled training data. But it is clear that even if we do not have labels available, the data still falls into fairly distinct clusters; recovering these clusters without using class labels corresponds to the clustering OCR described in this paper. (In their original, high dimensional space, these clusters are considerably better separated than in the low-dimensional non-linear mapping shown in the figure.)

If we compare this to previous methods for OCR by clustering, what it means practically is that we replace the ad-hoc, non-statistical clustering methods used in the literature [4] with a Gaussian mix-

ture estimation algorithm. Furthermore, statistics can also help us with additional questions that the original OCR approaches left unanswered, most importantly: how many clusters should there be?

6.4 Experiments

To study the feasibility of this approach to OCR, a simple prototype system based on the ideas described in this paper was implemented and applied to 1500 images of digits in the `cmr6` font from the Bell Labs database of severely degraded character images found on the University of Washington Database I. The input data was divided into 1000 training samples and 500 test samples. The images were centered, convolved with a Gaussian of $\sigma = 1$ and subsampled to a size of 10×10 . The resulting image was treated as a raw feature vector and 7 principal components were extracted. These PCA feature vectors were then used as input to a sample-distribution based classifier, as well as a mixture discriminant analysis-based classifier (MDA; [9]).

The clustering OCR system performs its unsupervised clustering using the method described by Frelley and Raftery[8] and implemented by the `mclust` package for the R statistical system. Clusters in this approach are represented as Gaussian distributions. The method first performs hierarchical clustering and follows it by Expectation-Maximization (EM) steps to optimize the cluster shapes. In these experiments, the cluster shapes considered by the algorithm were “spherical” (all clusters have spherical covariance matrices), “uniform” (all clusters have the same covariance matrix), and “unconstrained”.

In this way, the clustering OCR represents the unlabelled sample distribution as a mixture of 15 Gaussians. By assumption of the method, each Gaussian corresponds to a single digit label. When the assignment of labels to clusters is correct (either based on cryptanalysis or based on a non-specific classifier), the error rate of the sample distribution based recognizer on test data is 0.7% ($N=500$) in these experiments.

To compare the performance of the clustering OCR with a traditional approach to character recognition, a Mixture Discriminant Analysis (MDA) model[9] was trained. An MDA model represents likelihood functions as mixtures of Gaussians and uses Bayes rule to perform classification. The Gaussian mixtures are estimated using the Expectation Maximization (EM) algorithm. In the experience of the author, as well as based on results reported in the literature[9], MDA performance is roughly comparable to the performance of other, commonly used classifiers like neural networks and radial basis function methods. The R implementation of MDA (available from the R web site) was used for the experi-

ment. When the MDA classifier was trained on the training set (N=1000), its error rate on the test set was 0.6% (N=500)

6.5 Discussion

Ideas of clustering and style in OCR are not new and have been explored by a number of authors explicitly or implicitly. What this work contributes is a re-examination of clustering OCR methods from the point of Bayesian statistics, Gaussian mixtures, and mixture density estimation of the sample distribution. This helps both understand why and how clustering OCR methods work, and helps us improve them. The long term promise of this work is to arrive at classification methods that are considerably more robust to statistical differences between training and test data than traditional pattern recognition methods. The initial experiments presented above suggest that such an approach is feasible; more sophisticated implementations are needed to demonstrate that it delivers superior performance in real-world situations. For OCR systems in particular, this translates into much more robust recognition when novel fonts or document degradation conditions are encountered.

7 Classification by Probabilistic Clustering³

7.1 Introduction

Current classifiers for the recognition of handwriting, printed characters, phonemes, and similar signals can achieve very high performance (often exceeding that of humans) when given sufficiently large and representative training sets. Techniques have also been used to synthesize additional training examples from a given training set to further increase the effective training set (and ability to generalize) for the classifier. A key limitation of such approaches is still that they can be sensitive to novel data whose distribution is significantly outside the training set.

In the work described above, we have seen how modeling the sample distribution using Gaussian mixtures can achieve comparable font independent performance to existing classification methods. That work was also motivated by the application of clustering OCR methods to OCR applied in the compressed domain for document images compressed using token-based methods. This work describes a technique that builds on those ideas but uses a novel, non-parametric probabilistic clustering technique. The approach is based on modeling, using a multilayer perceptron (MLP), the probability

³This section is based on, and contains excerpts from, a paper to be presented at ICASSP '2001[3].

that two given images represent the same character. These probabilities are then integrated into an overall interpretation of a document using the maximum likelihood assignment of character identities to the individual images in the maximum entropy distribution compatible with the pairwise probability estimates derived from the MLP. Some experimental results are presented that demonstrate superior performance on a font-independent recognition task compared to traditional pattern recognition problems.

7.2 Pairwise Probabilities

In recognition by probabilistic clustering, rather than estimating $P(c, f|v)$, we estimate the pairwise probabilities $P(c = c', f = f'|v, v')$, i.e., the probabilities that two feature vectors represent the same character. The motivation for this approach is that we can imagine that determining whether two character images are similar or different may be considerably easier to perform in a font-independent manner than determining whether a given character image actually represents a particular character. For example, empirically, a simple but already fairly good statistic for determining the identity of two bilevel characters is to look at the minimum of the total area of their symmetric difference under arbitrary translations, normalized by the area of the larger of the two characters. This statistic can be computed completely independently of the font and distinguishes characters in a wide variety of fonts well.

If characters were perfectly distinguishable from their feature vectors, so that this probability only assumes values of 0 or 1, this would allow us to divide the set of feature vectors corresponding to characters on a page into equivalence classes. Each such equivalence class would then correspond to a single character class. Of course, we would still have to determine the identity of this equivalence class using some other means.

If $P(c = c', f = f'|v, v')$ can assume values other than zero or one, then the interpretation is more complex. An optimal interpretation of the whole document would be based on the joint conditional probability $\prod_i P(c_i, f_i|v_i)$ for all characters in the document. The conditional probability $P(c_i = c_j, f_i = f_j|v_i, v_j)$ is a marginal probability of this distribution, and we can use this to compute the maximum entropy joint conditional probability. In practice, however, since we are only using estimates of the pairwise probabilities, there is almost always no probability distribution that is consistent with the estimates for the pairwise probabilities. In order to address this problem, we need to formulate the problem of assigning classes to the different feature vectors as an optimization problem. A

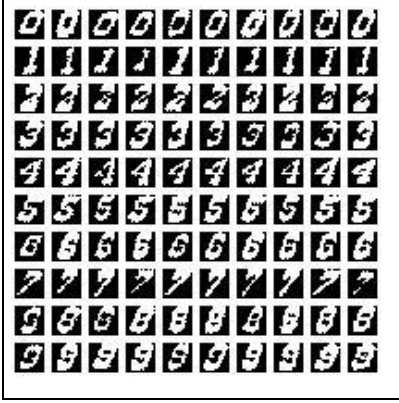


Figure 5: Examples of degraded characters used in the experiments.

simple method that suggests itself is to find an assignment of class labels to feature vectors that maximizes the product of all the pairwise probability estimates (we will not attempt a formal justification in this work). We can solve this optimization problem simply by simulated annealing, which appears to converge quickly in our experiments.

7.3 The Method

Recognition by probabilistic clustering therefore can be described as follows:

- estimate $P(c = c' \hat{f} = f' | v, v')$ based on a set of training examples $\{(c = c' \hat{f} = f', v, v'), \dots\}$ (for many different fonts)
- when faced with the problem of recognizing a new collection of feature vectors v_i , compute $\hat{P}(c_i = c_j \hat{f}_i = f_j | v_i, v_j)$ for each pair of feature vectors v_i, v_j
- assign cluster labels χ_i to the feature vectors v_i such as to maximize $\prod \hat{P}(\chi_i = \chi_j | v_i, v_j)$, for example using simulated annealing
- determine the correspondence between the cluster labels χ_i and the actual classes (and fonts, if desired)

7.4 Experiments

The dataset used in these experiments consisted of 71700 images of digits derived from 717 TrueType fonts from a commercial collection of type fonts (examples are shown in Figure 5). This dataset was split into 64600 training images representing 10 degraded samples of each digit from each of 646 fonts, and 7100 test images representing 10 degraded samples of each digit from each of 71 fonts. The character images were rendered using the Freetype engine, which performed antialiased rendering of greyscale

images of characters under affine transformation. Character images were degraded using the Baird character degradation model[14] with its standard settings, a widely used and studied model for modeling degradation of printed text under a variety of common document imaging conditions. Characters were rescaled to fit into a 16×16 square, giving rise to a 256 dimensional feature vector.

In a first step, to characterize the dataset, this feature vector was used as input to a multilayer perceptron. The MLP had 256 input units, 15 hidden units, and 10 output units. The test set error of the MLP was 9.46%. This may appear like a high error rate for OCR, but it is important to keep in mind that this test is different from most traditional OCR tests, since it (deliberately) involves a very wide diversity of fonts that are severely degraded and represented at low resolution.

For classification based on probabilistic clustering, the probability $P(c = c' | v, v')$ was estimated as follows. For each font in the dataset, the 4950 pairs of non-identical character images representing the same digits, as well as a random set of 4950 pairs of non-identical character images representing different digits were selected. Two 16×16 images were computed for each pair: the absolute difference between the two images, and the sum of the two images (at an offset that minimized the difference). These two images were used as a 512 dimensional feature vector and input into a MLP. The MLP had 512 input units, 15 hidden units, and 1 output unit. Training proceeded by training the output unit to “1” for pairs of character images representing the same digit and to “0” for pairs of character images representing different digits. It is well known that this training procedure will asymptotically converge to an estimate of the conditional probability that $c = c'$ given the input feature vector.

For testing, the input to the system consisted of 100 digit images from each font. For each pair of digit images, the pairwise probabilities $P(c = c' | v, v')$ was computed. For the simulated annealing step, a classification derived from the “traditional” classifier $\hat{P}(c | v)$ (modeled by the MLP described above) was used as the starting configuration. When this procedure was carried out for the test set, the performance of the system improved from 9.46% for the traditional MLP-based classifier to 7.66% for the clustering classifier.

7.5 Discussion

This work describes an approach to classification based on the estimation of a class-independent probabilistic model of the similarity of two feature vectors, followed by a probabilistic clustering method. Future work will include better cluster assignment

methods, a more formal analysis and better parametric models of character similarity, and automatic ways of assessing cluster validity. Perhaps most importantly, the assignment of labels to clusters by initializing the simulated annealing process is sub-optimal because its performance is limited intrinsically by the quality of the traditional classifier (significantly incorrect initial assignments will result in permuted label assignments in the output). Several better methods offer themselves: use of the traditional classifier as a prior, greedy assignment of cluster labels based on predominant classifications of the members of each cluster, and the use of statistical language models.

While it will be desirable to design experiments more specifically to explore and demonstrate the ability of the approach to handle variations in font, degradation, and robustness to samples outside the training set, the results presented in this work, generalization of classification to a varied and difficult test set of novel degraded fonts, already suggest classification by clustering holds the promise of being a general approach to addressing problems that are hard for traditional classifiers: coping with stylistic variations and generalization to samples outside the training set.

References

- [1] T. M. Breuel. Modeling the Sample Distribution for Clustering OCR. In *SPIE Conference on Document Recognition and Retrieval VIII*, 2001.
- [2] Thomas M. Breuel. Layout analysis by exploring the space of segmentation parameters. In *Proceedings of the 4th IAPR Workshop on Document Analysis Systems (DAS 2000)*, December 2000.
- [3] Thomas M. Breuel. Classification by probabilistic clustering. In *Proceedings of the 2001 International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2001)*, Salt Lake City, Utah, May 2001. IEEE. To appear.
- [4] R. Casey, S. K. Chai, and K. Y. Wong. Unsupervised construction of decision networks for pattern classification. In *Proc. ICPR-7*, July 1984.
- [5] A. Dengel. About the logical partitioning of document images. In *3rd Symposium on Document Analysis and Information Retrieval, Las Vegas*, pages 209–218, 1994.
- [6] D. Doermann. The indexing and retrieval of document images: A survey. Technical Report CS-TR-3876, University of Maryland CS Department, 1998.
- [7] D. Doermann, C. Shin, A. Rosenfeld, H. Kainiskangas, J. Sauvola, and M. Pietikainen. The development of a general framework for intelligent document image retrieval. In *Document Analysis Systems*, pages 605–632, 1996.
- [8] C. Fraley and A. E. Raftery. How many clusters? which clustering method? answers via model-based cluster analysis. Technical Report No. 329, Dept. of Statistics, U. of Washington, February 1998.
- [9] T Hastie and R Tibshirani. Discriminant analysis by gaussian mixtures. Technical report, AT&T Bell Laboratories, 1994.
- [10] D. Ittner and H. Baird. Language-free layout analysis, 1993.
- [11] Frederick Jelinek. *Statistical Methods for Speech Recognition*. MIT Press, Cambridge, Massachusetts, 1997.
- [12] Rolf Johannesson and Kamil Sh. Zigangirov. *Fundamentals of Convolutional Coding*. IEEE Press, 1999.
- [13] Anthony C. Kam and Gary E. Kopec. Document image decoding by heuristic search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(9):945–950, September 1996.
- [14] T Kanungo, H Baird, and R Haralick. Estimation and validation of document degradation models. In *Proc. 4th Annual Symposium on Document Analysis and Information Retrieval*, Las Vegas, NV, April 1995.
- [15] Gary E. Kopec and Philip A. Chou. Document image decoding using Markov source models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(6):602–617, June 1994.
- [16] Jishen Liang. *Document Structure Analysis and Performance Evaluation*. PhD thesis, University of Washington, 1999.
- [17] Thomas P. Minka, Dan S. Bloomberg, and Kris Popat. Document image decoding using iterated complete path heuristic. In *Proceedings of IS&T/SPIE Electronic Imaging 2001: Document Recognition and Retrieval VIII*, January 2001.
- [18] L. O’Gorman. The document spectrum for page layout analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1162–1173, 1993.
- [19] Kris Popat. Decoding of text lines in grayscale document images. In *Proceedings of the 2001 International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2001)*, Salt Lake City, Utah, May 2001. IEEE. To appear.
- [20] Kris Popat, Dan Bloomberg, and Dan Greene. Adding linguistic constraints to document image decoding. In *Proceedings of the 4th international workshop on document analysis systems*. International Association of Pattern Recogni-

- tion, December 2000.
- [21] Kris Popat, Dan Greene, Justin Romberg, and Dan S. Bloomberg. Adding linguistic constraints to document image decoding: Comparing the iterated complete path and stack algorithms. In *Proceedings of IS&T/SPIE Electronic Imaging 2001: Document Recognition and Retrieval VIII*, January 2001.
 - [22] P Sarkar. *Style Consistency in Pattern Fields*. PhD thesis, Rensselaer Polytechnic Institute, May 2000.
 - [23] A. L. Spitz. Style-directed document recognition. In *Workshop on Document Layout Interpretation and its Applications (DLIA)*, 1999.