Training on Severely Degraded Text-Line Images

Prateek Sarkar
Palo Alto Research Center
Palo Alto, CA 94304 USA
psarkar@parc.com

Henry S. Baird
Palo Alto Research Center
Palo Alto, CA 94304 USA
baird@parc.com

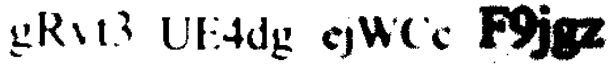
Xiaohu Zhang
Stanford University
Stanford, CA 94305 USA
zhangxh@stanford.edu

Abstract

We show that document image decoding (DID) supervised training algorithms, as a result of recent refinements, achieve high accuracy with low manual effort even under conditions of severe image degradation in both training and test data. We describe improvements in DID training of character template, set-width, and channel (noise) models. Large-scale experimental trials, using synthetically degraded images of text, have established two new and practically important advantages of DID algorithms:

1. *high accuracy (>99% chraracters correct) in decoding using models trained on even* **severely degraded images** *from the same distribution; and*
2. *greatly improved accuracy (<1/10 the error rate)* **across a wide range of image degradations** *compared to untrained (idealized) models.*

Examples of degradations for which these claims hold:



This ability to train reliably on low-quality images that suffer from massive fragmentation and merging of characters, without the need for manual segmentation and labeling of character images, significantly reduces the manual effort of DID training.

1. Introduction

It is widely agreed that severe document image degradations are important causes of OCR errors [15][16]. It is also generally accepted that training on a sufficiently large set of representative correctly labeled glyph (isolated character) images can improve accuracy, under many pattern recognition training methodologies [4][13]. 'Representative' means, of course, drawn from the same distribution — here, the same typefaces, type sizes, image degradations, language model, etc. — as the test data.

However, in practice, segmenting a sufficient number of document images into isolated glyph images, and then labeling each glyph image correctly with its ground-truth symbol, is itself a high-skill, tedious, and thus often prohibitively expensive manual effort. Even merely segmenting into glyphs, without labeling, is difficult to automate fully, especially when the images suffer from degradations severe enough to cause glyph images to split into fragments or to merge with their neighbors.

We describe algorithms for training which eliminate the need for segmenting page images into isolated glyphs before training on them. All they require is presegmentation into *images of entire lines of text* presented in the same order as the ground-truth lines of encoded text. The algorithms automatically align glyph images with their corresponding encoded characters in the ground truth. The training process is robust even when the initial choices of templates differ significantly in image quality from the final trained results. Robustness in the face of such differences reduces the skill-level, manual effort, and time required — and thus reduces training costs. These algorithms are an improvement over *supervised DID template estimation* [10].

We report large-scale experimental trials in which these algorithms proved able to learn DID models of high quality (that is, enabling recognition with accuracy greater than 99%) even when the training and test images are of low quality. The trials were carried out using pseudo-randomly degraded images permitting systematic mapping of the domain of competency of the training and decoding algorithms. We exhibit sample images at extreme points where the algorithms begin to fail.

We summarize prior art in the DID framework in the following section, and our training algorithms in Section 3. We describe our experiments and results in Section 5. Section 6 contains a brief discussion of results.

[0]Published in *Proceedings, IAPR 7th International Conference on Document Analysis and Recognition* (ICDAR03), Edinburgh, Scotland, August 3–6, 2003.

2 Background: Published DID Training Algorithms

Document image decoding (DID) is an approach to document recognition that is based on a communication theory view of the processes of document creation, transmission and recognition [8]. The mathematical framework permits the tasks of character segmentation, classification, and Markov language modeling to be combined into a single joint optimization problem. A major issue in applying DID to a particular recognition problem is developing explicit models for the glyph shapes found in the documents to be recognized, and corresponding layout parameters (set-widths) [7]. DID emphasizes the use of document-specific templates as glyph models, rather than "omni-font" features, since the error rate of a font-specific recognizer can be far less than that of a multi-font system [2]. A major advantage of DID is that there is no need to segment input text-line images into individual glyph images prior to classification; also, DID glyph templates need not be rectangles (a template's "support" region can have any shape, so long as neighboring glyphs' supports do not share any pixels).

An approach to supervised maximum likelihood (ML) template estimation, in which inputs to the training procedure are images of entire text lines plus their corresponding text-line transcriptions, was described in [10]. Briefly, it is a three-phase iterative procedure. The first phase is *transcription alignment*, in which glyph origins are located and labeled using a current set of templates. The heart of the algorithm is the second phase, *aligned template estimation* (ATE), during which templates are estimated using the current labeled glyph positions. The third phase is *channel estimation* in which parameters of the assumed image degradation model are updated. The procedure is applicable to the class of multilevel glyph shape models defined in [11] and the class of Markov image sources defined in [8].

3 EM Training Algorithms

An EM algorithm for DID template and channel estimation was developed and implemented by the late Gary Kopec who partially documented it in an as-yet unpublished work [12]. He found that EM methods improved template estimation compared to [10]. By using information from all paths through the transcription image source, rather than merely the single best path, the EM algorithm makes better use of training data. Systematic trials and improvements to Kopec's implementation by the first and third authors enabled the large scale experiments described.

The heart of the new algorithm is a three-phase process, as in [10], except that explicit transcription alignment is replaced by a forward-backward algorithm similar to those used for training HMMs.

Forward-Backward Alignment: Input to this stage are the text-line image, the current "best-guess" DID models, and the ground-truth transcription. Its goal is to choose a sequence of templates whose labels match the transcription and whose glyphs match the training image, subject to the constraints of the DID models. This is achieved by a two-step procedure: (1) combine the image source model and the transcription into a 'transcription image source'; and (2) apply a forward-backward algorithm as the E-step in an EM algorithm for computing posterior probabilities over "paths" that may have generated the line-image. A description of the forward-backward algorithm for one-dimensional signals may be found in [4]. Scheduling is simplified by assuming no backward transitions; and computation is reduced by following a given baseline rather than an exhaustive 2-D image search.

Aligned Template Estimation (ATE): The inputs to ATE are a text-line image and high score paths in the transcription image source (equivalently, the set of labeled glyph positions defined by the paths along with the respective forward-backward scores). An ATE algorithm must decide which of the pixels in each glyph image region belong to the labeled glyph, and which belong to adjacent glyphs or background (the background level). Then, within the support region, each template pixel is assigned to one of (in these experiments) three other "levels" which correspond to 'write-black', 'write-white', and 'sometimes-black' (*e.g.*, edge) pixels. These assignments are guided by the combined evidence from all of the paths and corresponding images for that glyph. For this, a greedy assignment algorithm, with a few refinements, has been found sufficient.

Channel Estimation: Each of the four 'levels' of pixels in templates is described by a "channel" (or, noise) model which gives the probability that each image pixel, when templates are printed, will be black or white. Channel models for levels are inferred from evidence combined across all aligned glyphs in training data by the forward-backward algorithm.

Set-width and Origin Estimation: Set-widths are estimated from an analysis of the best transcription alignments for the training images. Since DID allows for insertion of "thin spaces" that cause a displacement in the image without adding to the message string, currently set-widths and kerning distances are estimated as the minimum observed pixel displacements between training glyph pairs. A discussion of this process may be found in [10].

Initial Conditions: Our training process begins with the alignment step which requires an initial guess at templates and channel parameters. We have found that EM-training is not sensitive to these initial models. It is, however, sensitive to initial choices of glyph size and set-width. For our

experiments, we chose a "best-guess" initial font from a library (*e.g.*, PostScript) of computer-legible bitmap or outline fonts. Another way to obtain initial templates is to excise sample images from training images using an interactive image-editing tool. Set-widths may also be estimated by solving linear equations as in [14].

4 Engineering Issues in EM-Training

A full description of the algorithms will not fit within the page limit. Here are highlights.

EM iterations: The probabilistic multilevel templates and channel models are updated using an EM algorithm. The origin and set-width for each glyph are updated, outside of the EM loops, using the methods described in [10]. We have found that running EM again after updating the set-widths, and iterating a few times, leads to better estimates of template and channel models.

Local maxima: When samples of training glyphs for a template align well, the template estimation process works smoothly. But often the distribution of match positions for some templates is multimodal. This results in ghost replicas in estimated templates, which in subsequent iterations continue to yield multiple match positions (a local maximum trap for EM). The result is bad template estimates. Setting a threshold on accumulated pixel weights during the training process helps in reinforcing the dominant mode, and knocking the system out of a local maximum.

Sensitivity to size, set-width, and origin parameters: While the template "printing" and channel degradation models in DID are explicitly probabilistic, the bounding box sizes for glyphs, and set-widths are not. As a result the training algorithms often fail to converge to accurate models if the initial size and displacement parameters are far from true values. In particular, bounding box sizes should be initialized to bigger than expected, while set-widths should be initialized to smaller than expected, but not too small (DID allows for insertion of "thin spaces" during decoding). The requirement that glyphs of adjacent characters do not share support pixels makes the estimation of set-widths a difficult problem.

Given approximately correct initial set-widths, and generous initial bounding box sizes, the training algorithm is remarkably robust to the choice of initial glyph-shapes. However, the estimation of displacement parameters remains the weakest link in DID training.

5 The Experiments

The experiments use, for both training and test data, synthetic images of text lines degraded using the model of [1] (see summary in the Appendix). The output resolution `resn` was fixed at 300 pixels/inch. Most of the experiments are variations on this **nominal degradation model**:

- `size` = 10 point;
- `blur` = 1.7 output pixels (standard-error of 2D Gaussian kernel);
- `sens` = 0.025 intensity (standard-error of Gaussian w/ mean 0.0); and
- `thrs` = 0.30 intensity.

Here's an example image of Times New Roman text degraded using this nominal model:

This is roughly equivalent to a 1st or 2nd generation photocopy without careful scanner/printer adjustment.

All of the text used in the experiments was generated pseudo-randomly with a *uniform character unigram model* over the 64-letter alphabet { **A-Z a-z 0-9 .,** } with spaces added at about ten times the frequency of letters. All image data used for both training and testing consisted of synthetic images of entire text lines. The baselines of all images were the true baselines: (*i.e.* they were not estimated from analysis of the images).

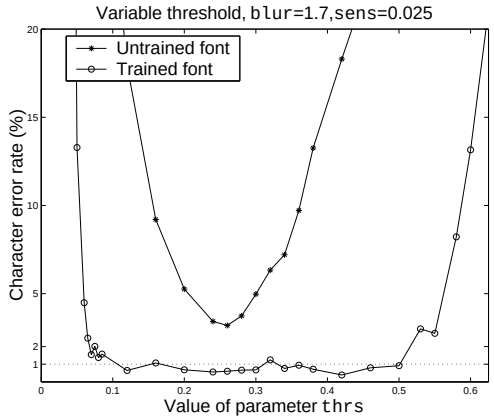
To generate the training and test images, each line of pseudo-random text was first rendered, at 972dpi 32pt Times New Roman, with a public domain program (*ftstrpnm*), from the FreeType project [19]. An implementation of Baird's degradation model (blurring, additive noise, thresholding, subsampling) was then applied to obtain 10pt size degraded text.

For each choice of image degradation parameters, we generated 200 text-line images for training and another 200 text-line images for testing. Each text line so generated contained 60 characters including spaces. Thus each set contained 12,000 characters. We compressed multiple spaces into one space everywhere, yielding about 11,400 characters: of these, about 10,000 were printable (non-space) characters. Thus each of the 64 printable characters was represented by about 150 images.

We trained DID models on the training images supervised by the ground-truth text; then we decoded the (distinct) test set (assuming a uniform character unigram language model over all characters, including spaces) using an implementation of the Iterated Complete Path algorithm [3]. Levenstein string-edit distances, between decoded text and ground-truth, were used to estimate the number of characters decoded incorrectly; this gave a *character error rate* for each text-line image.

To test the ability of our algorithms to train on severely degraded images, we varied the degradation parameters `thrs`, `blur`, and `sens` one at a time, stepping away from nominal values until we observed high decoding error rates.

Varying the Threshold Parameter: Lowering the binarization threshold monotonically increases coverage by black pixels causing thickened black strokes. At low thresholds holes close, corners get rounded off, and adjoining characters touch, while at high threshold serifs and thin strokes disappear, and characters are split. We varied the `thrs` parameter from 0.04 to 0.80; the results are shown in Figure 1. The images are examples of thresholding at 0.08 and 0.46. Error rates remain low (0.39-1.25%) across a wide range of values, from `thrs` = 0.12-0.50. It hovers below 2% at `thrs` as low as 0.07. Slightly outside of this range, error rates rise rapidly. Confusions such as "."/",", b/h, s/z characterize errors in low `thrs`, whereas the other extreme is characterized by a wide range of confusions resulting from obliterated glyphs.

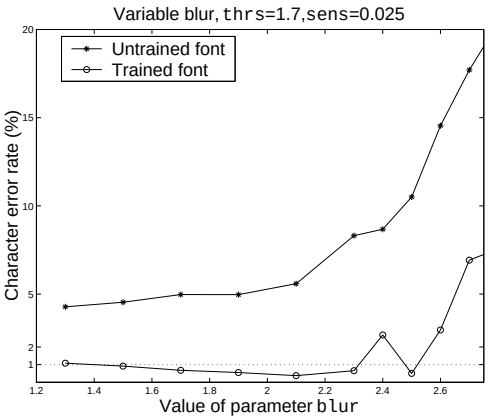


0.08:

0.46:

Figure 1. Character error rate (per cent) of decoding, for values of the threshold parameter `thrs`. Also shown are examples from the fringes of stable performance region.

Varying the Blur Parameter: We varied the `blur` parameter from 1.3 to 3.3 pixels; the results are shown in Figure 2. At low `blur` digitization approaches ideal sampling. With wider blurring kernels (also called point spread kernels) thin strokes get thinner, while wide strokes get wider, as shown in the example corresponding to `blur`=2.5 output pixels. Error rates remain low hover around or below 1% for `blur` values up to 2.3. Above this range, errors climb but remain below 10%. Discounting space insertion errors, caused by bad estimates of the width of a space character, errors remain under 3.5%. Substitution of "." by a space dominates errors, presumably because periods disappear due to heavy blurring.



1.5:

2.5:

Figure 2. Character error rate (per cent) of decoding, for values of the blur parameter `blur`. Also shown are examples from the fringes of stable performance region.

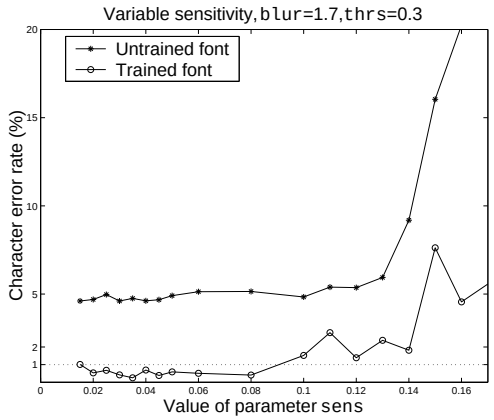
Varying the Sensitivity-Error Parameter: We varied the `sens` parameter from 0.015 to 0.220 (units of intensity); the results are shown in Figure 3. Higher values of `sens` result in higher "salt and pepper" noise. Error rates remain low (0.38-1.52%) across a wide range of values from `sens` = 0.015 through 0.100. Above 0.140, error rates rise rapidly: to 13.3% at 0.220. Most common confusions are between characters with differences in a few pixels, such as "."/"," and I/1/l.

Varying the training sample size: A training set size of 200 text-line images, on which we have reported, corresponds roughly to four full text-page images, perhaps not an excessive requirement in practice. We have repeated all experiments with as few as 40 lines of training data (30 samples per printable character), and have observed no significant deterioration in training and decoding performance. We intend to investigate lower limits of training sample size required.

6 Discussion

6.1 Practically Important Features

The document image decoding (DID) approach is designed for document-specific training, where a large batch of images of similar document must be decoded with high accuracy. DID allows high-performance models to be



0.015:

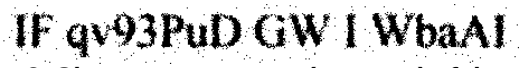
0.120: 

Figure 3. Character error rate (per cent) of decoding, for values of the sensitivity-error parameter `sens`**. Also shown are examples from the fringes of stable performance region.**

trained using only a small fraction of the images. Training is least expensive, and accuracy is highest, when the documents in the batch share the largest number of characteristics: the same language, typefaces, type sizes, and image quality.

We have described algorithmic refinements which enhance one of DID's most promising features: that training, although supervised, requires a *minimum of manual effort* since the training images need not be presegmented into isolated glyphs and words and labeled with their ground-truth. Among competing techniques, HMM models based on vertical image slices also offer this — but DID offers yet more: since its templates are explicitly 2-D and can have non-rectangular support, DID can handle complex text such as italicized fonts and mathematical notation.

We have shown that this important property holds even when training and test data are severely degraded. It is encouraging to note that the DID 'channel' (noise) model is not the same as the Baird model which was used for training and test data generation — and yet DID performance held up well.

6.2 Interpreting the Error Rates

The meaning of error rates always depends on the underlying distribution of characters in both training and test samples. While we assumed uniform distribution over characters, in English text the distribution follows Zipf's law. This would, of course, also affect training because more data may be needed to provide enough training samples for rare characters such as "Q".

In the cases where we measured error rates less than 1%, about 65% of errors were confusions among $\{$**I,l,1**$\}$ ('eye', 'el', and 'one'). In practice, given a language model, most of these can be disambiguated.

In several cases, especially on severely degraded images where the error rate is greater than 1%, a large fraction of the errors (often $> 50\%$) are inserted spaces. This is triggered by a too-narrow set-width being estimated for the space template. Since DID accommodates extra spacing by inserting thin spaces, space set-width is set to the minimum observed spacing in training samples. This, coupled with the lack of anchoring foreground pixels in a space character template, make estimation of space set-widths unreliable in degraded images.

Compared to untrained models, post-training error rates decreased by an average factor of 17, the average being over all cases where pre-training error was below 20%.

6.3 Design of the Experiments

We assumed a stationary degradation model, *i.e.*, each training and test sample is generated using the same values of degradation parameters; only the seed for the pseudo-random number generator changed from training data to test data. While this methodology allowed a systematic and replicable trial, a more realistic test of robustness would allow the degradation parameters to vary over small ranges, within both training and test samples.

We used the baseline location defined by the ideal (undegraded) font properties during data generation, rather than through image analysis. Thus our results do not include errors that may result in practice from running baseline finding algorithms. However, DID software, both for training and decoding, can accommodate a few pixels of positional variation of characters away from the baseline.

We have not explored geometric deformations that arise from the physical positioning of the paper during scanning, such as skew, non-linear baselines, and gutter and page-edge defects due to uneven lighting, incorrect focus, and perspective. The DID software is, however, equipped to handle skewed and nonlinear baselines.

7 Conclusion

Experiments on synthetic degraded images have demonstrated that training of DID models, and their application to decoding is robust and yields very high OCR accuracy over a wide range of image quality. DID models for character templates, character set-widths, degradation channel are trained automatically from page images presegmented

into text lines, and corresponding ground-truth. This obviates expensive and error prone steps of character segmentation and labeling. Training on representative data resulted in OCR error rates under 1% (under a uniform character unigram distribution) over a wide range of image qualities, including severely degraded (eroded or darkened) text. Examples of such images have been included in Section 5.

Acknowledgments

Our experiments are an attempt to finish and publish unfinished work of the late Gary Kopec. The code base was inherited from him, and minor engineering changes implemented by the authors to make the programs workable and robust. We are grateful to Tom Breuel for a fresh implementation of the generative image degradation model of [1].

Appendix: the Image Degradation Model

A ten-parameter model proposed in [1] approximates some aspects of the known physics of machine-printing and imaging of text, including symbol size, spatial sampling rate and error, affine spatial deformations, speckle noise, blurring, and thresholding. We have used a subset of this model:

`size:` type size in points;

`resn:` spatial sampling rate (resolution), in pixels/inch – the 'output' resolution; (dimensionless);

`blur:` standard error of a circularly symmetric Gaussian blurring point-spread function kernel, in pixels;

`sens:` pixel sensitivity: standard error a Gaussian distribution with 0 mean, randomized per pixel and added to intensity after blurring and before thresholding, in units of intensity;

`thrs:` binarization threshold, in units of intensity.

The model is used to generate synthetically degraded images of text as follows. First an "idealized" bilevel image of the text is rendered, at about ten times the output resolution. Spatial sampling error is added by shifting in X and Y randomly uniformly within [0,1] in output pixel units. This is then blurred and subsampled giving a grey-level image at the output resolution. To each pixel's intensity value a sensitivity error is added, randomly for each; then the pixel's intensity is thresholded by `thrs` giving a black (0) or white (1) value.

References

[1] H. S. Baird, "Document Image Defect Models," in H. S. Baird, H. Bunke, and K. Yamamoto (Eds.), *Structured Document Image Analysis*, Springer–Verlag: New York, 1992, pp. 546-556.

[2] H. S. Baird and G. Nagy, "A self-correcting 100-font classifier", in *Document Recognition*, L. Vincent and T. Pavlidis, editors, Proc. SPIE vol. 2181, pp. 106–115, 1994.

[3] D. S. Bloomberg, T. P. Minka, and K. Popat. "Document image decoding using iterated complete path search with subsampled heuristic scoring," in Proceedings of the sixth ICDAR, September 2001.

[4] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, John Wiley & Sons, New York, 2001.

[5] A. Kam and G. Kopec, "Separable source models for document image decoding", *Document Recognition II*, L. Vincent and H. Baird, editors, Proc. SPIE vol. 2422, pp. 84–97, 1995.

[6] A. Kam and G. Kopec, "Document image decoding by heuristic search", *IEEE. Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 9, Sept. 1996, pp. 945–950.

[7] G. Kopec and P. Chou, "Automatic generation of custom document image decoders", *Proc. Second Intl. Conf. on Document Analysis and Recognition*, Tsukuba Science City, Japan, Oct. 20–22, 1993.

[8] G. Kopec and P. Chou, "Document image decoding using Markov source models", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 16, no. 6, June, 1994, pp. 602–617.

[9] G. Kopec and M. Lomelin, "Document-specific glyph template estimation", in *Document Recognition III*, L. Vincent and J. Hull, editors, Proc. SPIE vol. 2660, pp. 14–26, 1996.

[10] G. E. Kopec and M. Lomelin, "Supervised Template Estimation for Document Image Decoding," *IEEE Trans. on PAMI*, Vol. 19, No. 12, pp. 1313–1324, December 1997.

[11] G. Kopec, "Multilevel character templates for document image decoding", in *Document Recognition IV*, L. Vincent and J. Hull, editors, Proc. SPIE vol. 3027, 1997.

[12] G. Kopec. "EM estimation of templates." Unpublished manuscript. Xerox PARC, 1997.

[13] G. Nagy and Seth, "Modern optical character recognition." in *The Froehlich/Kent Encyclopedia of Telecommunications*, Vol. 11, pp. 473-531, Marcel Dekker, NY 1996.

[14] Y. Xu and G. Nagy. Prototype extraction and adaptive OCR. *IEEE Transactions on PAMI*, 21(12):1280–1296, 1999.

[15] S. V. Rice, G. Nagy, and T. A. Nartker, *OCR: An Illustrated Guide to the Frontier*, Kluwer Academic Publishers, 1999.

[16] S. V. Rice, F. R. Jenkins, and T. A. Nartker, "The Fifth Annual Test of OCR Accuracy," ISRI TR-96-01, Univ. of Nevada, Las Vegas, 1996.

[17] R. Rubenstein, *Digital Typography*, 1988, Reading: Addison-Wesley.

[18] H. Stabler, "Experiences with high-volume, high-accuracy document capture", in *Document Analysis Systems*, L. Spitz and A. Dengel, eds., 1995, Singapore: World Scientific Publishing.

[19] D. Turner, "The FreeType Project". http://www.freetype.org.