Decoder Banks: Versatility, Automation, and High Accuracy without Supervised Training

Prateek Sarkar and Henry S. Baird
Palo Alto Research Center
3333 Coyote Hill Road, Palo Alto, CA 94304 USA
{psarkar|baird}@parc.com

Abstract

A methodology using decoder banks *is proposed for high-accuracy, fully automatic recognition of machine printed text across a wide range of challenging image qualities, without requiring manual intervention or supervised training. This approach is made possible by two crucial properties of* document image decoding *(DID) technology: (1) it is trainable for high accuracy across a wide range of explicitly parameterized image degradations; and (2) decoders for arbitrary parameter settings can be generated automatically. We report the results of large–scale experiments on synthetic images which demonstrate that, when many pretrained decoders are applied in parallel to an input image with unknown parameters, the decoder that yields the highest accuracy is often the one that exhibits the highest DID* posterior *'Viterbi score'. When implemented naively, in a brute–force manner, decoder banks are computationally intensive: but we suggest ways that this cost may be reduced with no loss of versatility, automation, or accuracy.*

1. Introduction

No existing document–image understanding technology, whether experimental or commercially available, can guarantee high accuracy across the full range of documents. Research at PARC has focused for more than ten years on relieving this critical bottleneck to automatic analysis of the contents of paper-based documents, FAXes, etc. PARC's *document image decoding* (DID) technology achieves high accuracy on images of documents printed in a potentially wide variety of writing systems and typefaces, languages, and severely degraded image quality [4][6]. Until now, DID methodology has emphasized retargeting: that is, supervised training of decoders, requiring manual effort to prepare training images and synchronized ground truth [5]. In recent years we have reduced the manual effort of DID training significantly by obviating manual pre-segmentation of images of text lines into words or characters during both training and testing [3]. We have also shown that DID decoders are trainable to high accuracy across a wide range of explicitly parameterized image degradations [7].

In this paper we propose scaling up DID methodology to cope with a wide variety of documents of image quality whose parameterizations are unknown, by the use of massively parallel recognition using ensembles ('banks') of automatically pre-trained DID decoders. We report experiments using synthetic data which show that the DID decoder (among all the decoders in the bank) which achieves the highest accuracy on input generated with unknown parameters is very often the one exhibiting the highest DID *posterior* 'Viterbi score.' By choosing this decoder, the need for manual document–specific training is eliminated with little or no loss in accuracy. Our *decoder banks* method is, in its present implementation, computationally intensive during both off-line generation of the decoders and on-line recognition of the document image – but we suggest, as topics for future work, approaches that may reduce this cost fully automatically, with no loss of accuracy, and no restriction on the range of application.

2. Training on Severely Degraded Text-Line Images

In [7] we showed that DID supervised training algorithms, can achieve high accuracy with low manual effort even under conditions of severe image degradation in both training and test data. Large-scale experimental trials, using synthetically degraded images of text, established two new

[0]Published in *Proceedings, IAPR 17th International Conference on Pattern Recognition*, Cambridge, UK, Vol. II, pp. 646–649, August 23–26, 2004.

and practically important advantages of DID algorithms: (a) 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.

These experiments (and the ones we now report) use, for both training and test data, synthetic images of text lines degraded using the model of [1]. 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.
- `skew` = 0; `xscl` = `yscl` = 1.0.

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 (*ft-strpnm*), from the FreeType project [8]. 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 [2]. Levenstein string-edit distances between decoded text and ground-truth were used to estimate the number of characters decoded incorrectly; this yielded a *character error rate* for each text-line image.

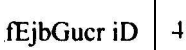
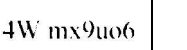
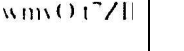
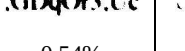
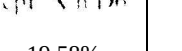
	`thrs`=0.08	`thrs`=0.30	`thrs`=0.50
`blur`=1.3	 0.18%	 1.09%	 0.73 %
`blur`=1.9	 2.45%	 0.65%	 12.06%
`blur`=2.5	 60.10%	 0.54%	 19.58%

Figure 1. Examples of degraded images for various parameter settings.

3. Training the Decoder Bank

We trained 21 decoders on datasets generated for the image-degradation parameters in the cross product of these values:

`thrs` = 0.08, 0.16, 0.24, 0.30, 0.38, 0.46, 0.50
`blur` = 1.3, 1.9, 2.5

Figure 1 illustrates the range of image quality expressed by these parameters. Note that it includes extreme cases of severe image degradations on which conventional OCR technology is well known to fail catastrophically.

4. Testing the Decoder Bank

We tested these decoders on 84 datasets generated for the image-degradation parameters in the cross-product of these values:

`thrs` = 0.08, 0.12, 0.16, 0.20, 0.24, 0.26,
0.30, 0.34, 0.38, 0.42, 0.46, 0.50
`blur` = 1.3, 1.5, 1.7, 1.9, 2.1, 2.3, 2.5

Figure 2 shows the result of testing a single decoder, trained for `thrs`=0.30 & `blur`==1.9, on all 84 test sets: it cannot perform well across the entire range of image qualities.

This is in stark contrast to the performance that is achievable when the image degradation parameters of the test data are known so that the 'tuned' decoder (trained for exactly the same parameters) can be applied (Figure ??): in this case, error rates remain low across a far broader region of the parameter space.

5. Automatically Selecting the Best Decoder

How close can we come to picking the highest-accuracy decoder without relying on prior knowledge of the degra-

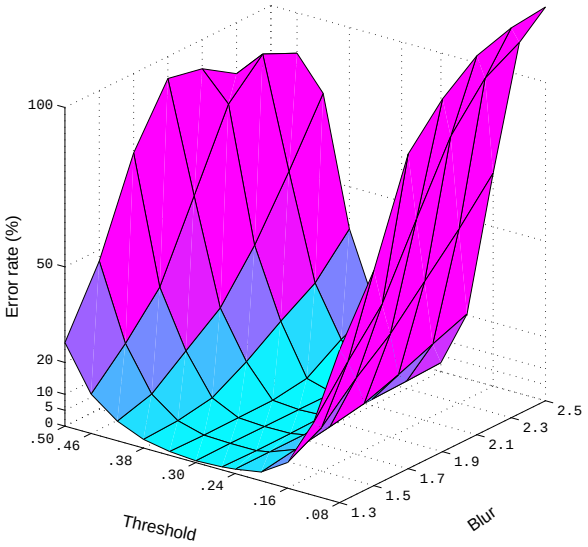


Figure 2. Character error rate (per cent) of decoding, as a function of the threshold (`thrs`) and the blurring `blur` parameters, for a single decoder (trained on `thrs`=0.30 and `blur`=1.9) showing the narrow range of its competence.

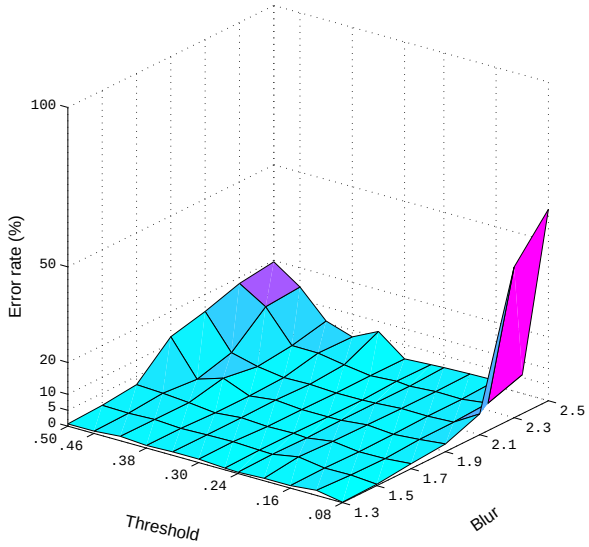


Figure 3. Character error rate (per cent) of decoding, as a function of the theshold (`thrs`) and the blurring (`blur`) parameters, for 'tuned' decoders trained on the same parameters as the test data.

dation parameters of the test data? We experimented with a brute-force technique in which all of the decoders are run and then the one whose 'Viterbi score' is highest (among all the decoders) is selected and its recognition result returned. The Viterbi score is the automatically computed estimate of posterior probability of correct match for the interpretation selected by the Viteri algorithm on a text-line image. The results of this policy are illustrated in Figure4: in the great majority of cases, the automatically selected decoder performed as well as the 'tuned' decoder.

Details of the remarkable improvement achieved are illustated in Figure ??? for four decoders: all were trained on `blur`==1.9, and each was trained on `thrs` = 0.16, 0.24, 0.30, 0.38 respectively. The 'domain of competence' achieved by each of the four decoders is narrow, but the 'highest Viterbi score' policy broadens it, nearly equalling the performance of tuned decoders which require prior knowledge of degradation parameters.

6. Discussion

The experiments suggest that the best document image decoder can reliably be chosen automatically from among a set ('bank') of pre-trained decoders, simply by picking the one whose 'Viterbi score' (an estimate of the posterior probablibility of correct match) is the highest. Thus the need for prior knowledge of image degradation parameters is eliminated, with little loss in accuracy. This advantage holds across a wide range of image qualities, including cases on which conventional OCR technology fails. Thus the proposed method of 'decoder banks' is promising for applications in which high accuracy across a wide range of severe image degradtions is required.

7. Future Directions

The present implementation of 'decoder banks' is brute force, running all the decoders to completion in parallel before selecting the best one: this policy multiplies the computation cost of recognition by a factor equal to the number of decoders in the bank. Future work will examine in detail the tradeoff between the number of decoders in the bank and expected loss of accuracy. We beleive that it is also possible, in principle, to combine all the decoders into a single one and then search among them simultaneously, applying well-studied search optimization strategies – *e.g.* iterated complete path search [2] – to reduce the total runtime without loss of optimality or range of application.

References

[1] H. S. Baird. Document image defect models, 1992.

[2] D. Bloomberg, T. Minka, and K. Popat. Document image decoding using iterated complete path search with subsampled heuristic scoring. In *Proceedings of the IAPR 2001 International Conference Document Analysis and Recognition (ICDAR 2001)*, Seattle, WA, September 2001.

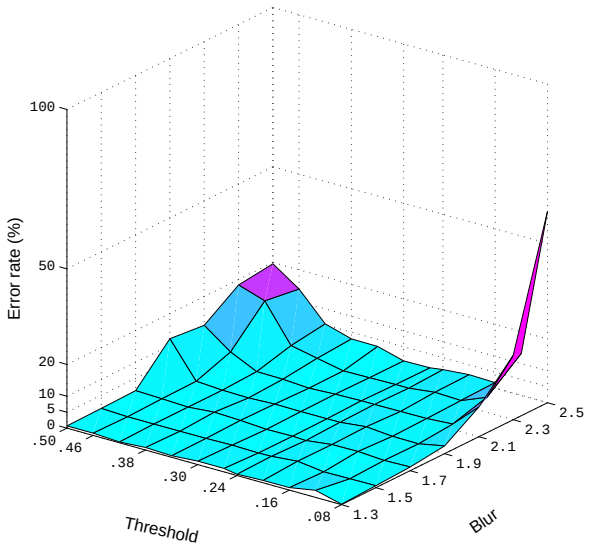


Figure 4. Character error rate (per cent) of decoding, as a function of the theshold (`thrs`) and the blurring (`blur`) parameters, for the 'highest Viterbi score' decoder selected automatically from among all the decoders in the bank. Note that performance is only slightly worse than for 'tuned' decoders for which prior knowledge of the degradation parameters is required.

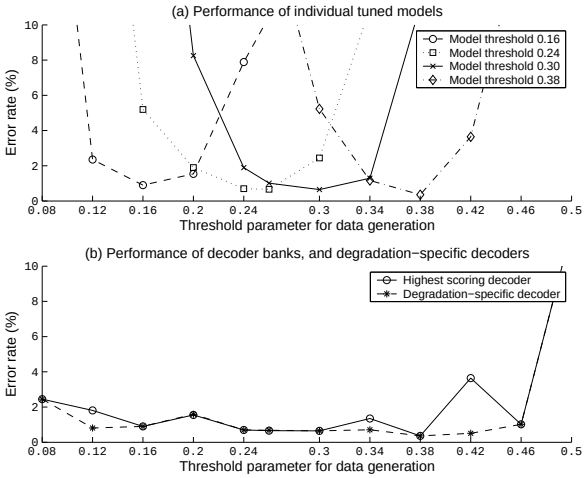


Figure 5. (a) Top: character error rate (per cent) of decoding, as a function of the theshold (`thrs`) parameter, for three decoders, showing the narrow domain of competence of each. (b) Bottom: error rate (1) for the "highest Viterbi score" decoder, among all decoders in the bank, and (2) for the degradation-specific decoder trained on ('tuned to') the parameter value of the test data.

[3] G. Kopec. An em algorithm for character template estimation. submitted March 1997; returned for revision, but not revised due to the author's death; available from PARC by request.

[4] G. Kopec and P. Chou. Document image decoding using markov source models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-16:602–617, June 1994.

[5] G. Kopec and M. Lomelin. Supervised template estimation for document image decoding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-19(12):1313–1324, December 1997.

[6] G. Kopec, M. Said, and K. Popat. N-gram language models for document image decoding. In *IS&T/SPIE Electronic Imaging 2002 Proceedings of Document Recognition and Retrieval IV*, San Jose, California, January 2002.

[7] P. Sarkar, H. S. Baird, and X. Zhang. Training on severely degraded text–line images. [submitted to] IAPR Int'l Conf. on Document Analysis & Recognition, Edinburgh, August, 2003.

[8] D. Turner. The freetype project. `http://www.freetype.org`.