ELSEVIER

Pattern Recognition Letters 19 (1998) 365–371

Pattern Recognition Letters

Low resolution, degraded document recognition using neural networks and hidden Markov models

M. Schenkel *, M. Jabri

University of Sydney, 2006 Sydney, Australia

Received 21 February 1997; revised 13 November 1997

Abstract

We collected a large, real world database, containing degraded, old and faxed documents and present a comparison between two leading edge commercial software packages and human reading performance which shows quantitatively the huge performance gap between humans and machines, even on random character documents where no context can be used. This indicates room for possible improvements. We implemented an integrated segmentation and recognition algorithm using neural networks and hidden Markov models trained on the database and present results which show the superior performance of the algorithm. © 1998 Elsevier Science B.V. All rights reserved.

Keywords: OCR; Document recognition; Low resolution; Neural networks; Hidden Markov models

1. Introduction

OCR of machine-print document images has matured considerably during the last decade. Recognition rates as high as 99.5% have been reported on 300 dpi omnifont documents. However, for lower image resolutions (200 dpi and below), noisy images, images with blur or skew, the recognition rate degrades considerably. Low resolution may be required because of data communication bandwidth and storage limitations. The distortions in the images may be due to distortions in the original or because no interactive control of the capture device is possible. Although these kind of documents pose no problems for humans to read, all commercial products we tested produce unsatisfactory recognition rates.

In bad quality documents, character segmentation is as big a problem as the actual character recognition. In many cases, characters tend either to merge with neighboring characters (dark documents) or to break into several pieces (light documents) or both. In the literature, segmentation has only recently attracted considerable attention.

Most character segmentation techniques can be divided into two categories (Casey and Lecolinet, 1996): (1) *external* segmentation, where a segmentation algorithm precedes the recognition and (2) *internal* segmentation where segmentation and recognition are combined. The method proposed in this paper performs combined recognition/segmentation. A space displacement neural network is trained to generate character probabilities. The output of the neural network is then post-processed by a hidden

* Corresponding author. Current address: Supercomputing Systems, Technoparkstr. 1, 8005 Zürich, Switzerland. Email: schenkel@scs.ch.

PII S0167-8655(97)00176-1

Markov model which effectively searches through the recognition character candidates for optimal character identifications and boundaries.

The rest of the paper is organized as follows: in Section 2 we describe the database, its composition and collection. Section 3 presents recognition performances of leading edge commercial software packages on the database and compares them with human reading performance. In Section 4 we present our integrated recognition/segmentation system. The performance of the algorithm and some variations are described in Section 5. Finally, Section 6 concludes and discusses future work.

2. Data collection

Gathering data is a time consuming and difficult task. That is why artificial data generation, according to some predefined defect models, is such a popular method for training and evaluating recognition algorithms. Yet such models rarely capture the true nature of the degradations encountered (see (Li et al., 1996) for a validation of defect models).

This led us to make the effort of collecting a database of various sources. We selected true degraded documents, as well as normal documents, faxed them, copied them and scanned them at 200 dpi and 400 dpi resolution. Fig. 1 shows some examples of the data. The database consists of three sections. Section one contains 35 real world English documents containing about 19 000 words and 97 000 characters. Section two consists of 24 self-printed and degraded random character documents in 12 fonts (10pt) containing about 6 500 words with 104 000 characters. The documents were copied either light or dark and faxed through a standard fax machine. Section three is made up from computer generated and degraded data, using additive noise, Gaussian smoothing and thresholding, it has about 1 600 entries containing 34 000 characters. All data is binary valued, labeled and segmented on word and character level. Each document consists of a TIFF file and a segmentation description file.

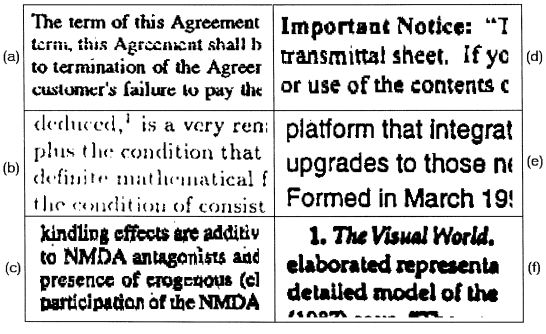

Fig. 1. Examples from the database.

We have made this data publicly available, in the hope that other researchers will make use of it, helping to achieve results which can be compared (http://www.sedal.usyd.edu.au).

3. Leading edge commercial OCR performance

To assess today's best recognition rates, we tested two leading edge commercial products on our low resolution database. Throughout this paper we use single character edit distance (SCED) for error counting. This method assigns a count of 1 for each insertion, deletion or substitution between the OCR generated text and the original text. No formatting information such as tabulators or carriage returns are counted. Also word segmentation errors have been removed which results in non-white-space error counting. This is done to make error rates comparable with our experimental system which is based on word units (excluding word segmentation errors). The transition from white-space counting to non-white-space counting increased the error rate on average by 10% (e.g. from 4.9% to 5.5% SCED).

In a first experiment we tested the influence of the scanning resolution on the recognition performance. We scanned two documents once at 400 dpi and once at 200 dpi (as used in fax machines). Table 1 shows

Table 1
Performance reduction due to small dpi. The error rates of a commercial package increase significantly (factor 2–3) only due to a reduction in resolution from 400 dpi to 200 dpi

Document	Without language	With language
Good 400 dpi	1.6	1.0
Good 200 dpi	5.6	3.4
Maintenance (a) 400 dpi	16.1	13.2
Maintenance (a) 200 dpi	26.5	26.5

the results of a leading commercial software package for the two documents: the first document (called ''Good'' in Table 1) is a good quality print in Times Roman 12pt and 14pt; the second document (called ''Maintenance'' in Table 1) is a poor copy of a document in times 10pt. In both cases an increase of a factor 2–3 in error rate occurs when reducing the scanning resolution from 400 dpi to 200 dpi.

In a second experiment we ran the commercial products on six of our real world documents, scanned at 200 dpi (see Fig. 1 for views from the documents). Table 2 shows the per character error rates with and without the use of the built-in language models. Although the recognition scores differ between the packages, we found that both packages performed unsatisfactorily on most documents. The performance of both packages is significantly worse on light documents (such as ''Maintenance'' or ''Broken'') than on dark documents (such as ''Vision'' or ''Neurofax''). Most broken characters are classified as ''i'', ''l'', ''I'', ''t'' or '','' with high enough confidence to be accepted by the software. Note that the overall performance of package 1 is better although the language model of package 2 seems superior.

3.1. Comparison with human reading performance

All of the above mentioned documents could be read by humans without any errors. This is a strong indication that significant improvements of machine recognition rates are possible.

It has been argued that humans use an extremely wide range of context information, such as current topics, syntax and semantic analysis on top of simple lexical knowledge to achieve such high recognition rates.

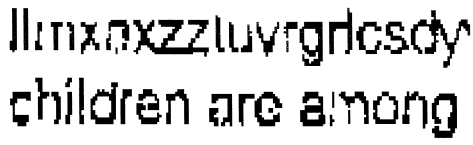

Fig. 2. Degraded writing: Example of the writing used to test human reading performance. Two variations were tested: random character sequences and legal English words.

To test the influence of higher context analysis by humans, we have conducted the following experiment: Human subjects were presented with a line of random character words and a line of English words on a screen. The words were isolated and presented in random order to prevent use of any semantic or grammatical context. The subject had no time restrictions for reading the word (or pseudo word).

The source document was printed in 10pt Arial Narrow, degraded by photocopying and faxing, and scanned with 200 dpi. Fig. 2 shows some of the random strings and words. The test was run independently for the degraded and the original document. The same document was presented to the commercial package 1.

The results are reported in Table 3. The difference between human and machine reading is striking, even when *no context can be used* (about a factor of two in the above experiment). The difference is even more striking in the case where word context can be used. Humans seem to be using word context far more efficiently than the software packages.

Table 2
Error rates of two commercial products on selected low quality documents. All documents are real world documents, scanned at 200 dpi. Samples of the documents are shown in Fig. 1(a)–(f). All errors are single character edit distance and no white-space counting. The last column gives the number of characters in each document

Document	Comm. 1		Comm. 2		Document size
	Without Dict.	With Dict.	Without Dict.	With Dict.	
Maintenance (a)	26.5	26.5	69.5	53.3	2980
Broken (b)	36.9	36.8	57.8	47.9	1672
Neurofax (c)	32.7	32.7	46.3	46.0	2061
Import (d)	9.6	5.5	20.4	15.6	395
Precept (e)	2.5	1.3	9.6	5.2	1904
Vision (f)	26.3	26.3	27.7	16.6	2643

Table 3
Human versus machine reading. The character error rates (single character edit distance) of the commercial package are significantly higher than human error rates. The difference is far more prominent when word context can be used.

Reader	Document	English words	Random characters
Humans	Clean	0	0
Commerc.	Clean	0	5
Humans	Degraded	0	19
Commerc.	Degraded	30	35

We conclude from this, that there is still room for significant improvements in the field of pure character recognition, where only geometrical shape is used. This is somewhat in contrast to the common believe that context is essential for good recognition of degraded documents. Even larger improvements can be made for word recognition, where word-based context models can be used, still excluding any computationally intense higher level context models.

4. A combined segmentation and recognition system

To see by how much one can improve on the state-of-the-art, we have decided to implement a combined segmentation/recognition system, using a combination of a neural network and a hidden Markov model as used in handwriting recognition (Seiler et al., 1996).

4.1. Combined segmentation and recognition

The basic principle of combining segmentation and recognition is to generate first many alternative ''tentative segmentations'' of which ''tentative characters'' are built and presented to a recognizer. The recognition scores ultimately determine which of the alternative string segmentations proposed is used. This is usually done by using some dynamic programming algorithm. Dictionaries can be built into this process. These schemes rely on the recognizer to give low confidence scores for wrong tentative characters corresponding to a segmentation mistake. Recognizers trained only on valid characters usually perform poorly on such a task.

One way of generating tentative characters is by sliding a restricted window over the input sequence in small steps. At each step the content of the window is taken to be a tentative character. The output of the recognition engine can be interpreted as a pseudo continuous estimation of the probability of the character classes. The subsequent graph algorithm can implement for example a model for expected character durations.

This approach does not take any early decisions about segmentation points, thus it is not possible to ''loose'' the correct segmentation early on. The graph algorithm finds the globally best path through all possible segmentations and recognitions. The recognition engine can be trained without the need to segment the training data. (e.g. forward-backward algorithm (Baum and Sell, 1968) or position-invariant-learning (Keeler et al., 1991)).

4.2. System design

In the *preprocessing* step we normalize the word to a fixed height (using an estimate of the core height as in (Seiler et al., 1996)). The result is a grey-normalized pixel map of the word (Fig. 3, bottom). This pixel map is the input to a *neural network* which estimates a posteriori probabilities of occurrence for each character given the input in the sliding window whose length corresponds approximately to two characters. We use a space displacement neural network (SDNN) which is a multi-layer feed-forward network with local connections and shared weights, the layers of which perform successively higher-level feature extraction. SDNNs are derived from time delay neural networks which have been successfully used in speech recognition (Lang and Hinton, 1988) and handwriting recognition (Matan et al., 1992; Guyon et al., 1991). Thanks to its convolutional structure the computational complexity of the sliding window approach is kept tractable. Only about one eighth of the network connections are reevaluated for each new input window.

The outputs of the SDNN are processed by a *hidden Markov model* (HMM). HMMs are finite state machines, which align a state sequence (which

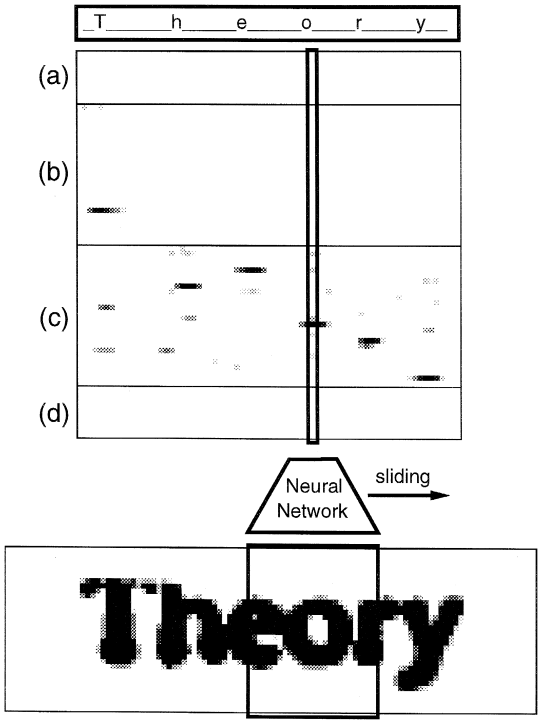

Fig. 3. Output probability map of recognition SDNN. Each row represents the probability of occurrence for a fixed class at all positions in the word. The darker a box is, the higher is its probability. (a) Numbers 0–9. (b) Uppercase characters. (c) Lowercase characters. (d) Selection of punctuation marks.

corresponds to a character category sequence in our case) with an observation sequence (the outputs of the SDNN). In our case the HMM implements character duration models. It tries to align the best scores of the SDNN with the corresponding expected character durations. The Viterbi algorithm is used for this alignment, determining simultaneously the segmentation and the recognition of the word. Finding this state sequence is equivalent to finding the most probable path through the graph which represents the HMM. Our HMM is tuned to model character durations (see (Schenkel et al., 1993) for details).

5. Experimental results

We tested a series of architectures, varying the window size, number of free parameters, number of layers and using 1-d and 2-d convolutional kernels. The system handles 72 classes: Numbers 0–9, Uppercase characters, Lowercase characters and a selection of punctuation marks (regions (a), (b), (c) and (d) in Fig. 3).

We use an image height of 28 pixels, which results in a resolution of 12 pixels for the core height (height of characters such as ''a'' or ''o''). Our window size was set to 31, which is 30% wider than the widest expected characters (being ''m'' and ''w''). Our best architecture contains 4 1-d convolutional layers with a total of 50 000 weights. (Kernel sizes in ascending layer order: 7, 11, 7, 9. Number of features in ascending layer order: 30, 30, 40, 72.) No under-sampling was used. Half of the weights are used in the last layer which performs the classification of the 72 classes.

The training set consisted of a subset of 180 000 characters from our database. To train the network we use the standard error backpropagation algorithm. For generating the targets we tested both, the forward-backward algorithm and the position-

Table 4
Error rates of the combined recognition/segmentation system. All documents are real world documents, scanned at 200 dpi. Samples of the documents are shown in Fig. 1(a)–(f). All errors are single character edit distance and no white-space counting. Three methods for the segmentation have been tested: Using the human entered segmentation positions (column ''True''), using our hidden Markov model (column ''HMM'') and using a neural network for segmentation (column ''NNSeg''). All results are without usage of context

Document	Best comm.	SDNN + True	SDNN + HMM	SDNN + NNSeg
Maintenance (a)	26.5	16.9	18.1	26.9
Broken (b)	36.9	14.1	16.4	22.9
Neurofax (c)	32.7	8.9	12.2	19.8
Import (d)	9.6	3.2	4.1	9.1
Precept (e)	2.5	9.9	13.9	17.0
Vision (f)	26.3	6.7	9.4	14.7

invariant-learning described above but found no significant difference in the performance of the resulting networks.

Fig. 3 shows the output probability map of such a SDNN. Each row represents the probabilities of occurrence for the corresponding class for all possible x-coordinate positions. The darker the box is, the higher is its probability.

We tested our combined recognition/segmentation system on the same documents used in our assessment of the commercial packages Table 4 shows the results. None of the documents were used during training. All results are without any context usage.

5.1. An alternative to the hidden Markov model

As an alternative to the hidden Markov model we have tested a sliding window approach to segmentation. Again we trained a space displacement neural network, this time to estimate probabilities for segmentation. The SDNN's receptive field is restricted to width of approximately two characters. The output unit produces scores which can be interpreted as the probability for a segmentation point in the center of the input field. For the training of the network we had to hand-segment the data to generate target values.

Fig. 4 shows an input word pixel map and the corresponding output of the SDNN. To obtain the final tentative segmentation points, we threshold the output and segment at the closest local minima of number of black pixels to be cut through.

We use the output of the network to segment the output map of the character recognition network as a replacement of the hidden Markov model. The corresponding recognition rate is given in Table 4. As can be seen, the HMM is clearly superior in its performance. We suspect the reason for this to lie in the ability of the HMM to *globally* optimize segmentation and recognition scores, whereas the segmentation network only *locally* decides on segmentation points, independently of the corresponding recognition scores.

By treating the proposed segmentation points as tentative and using them in a recombination or search scheme, we expect improvement of this approach. However, visual inspection showed that the proposed segmentation points do not always include the true segmentation.

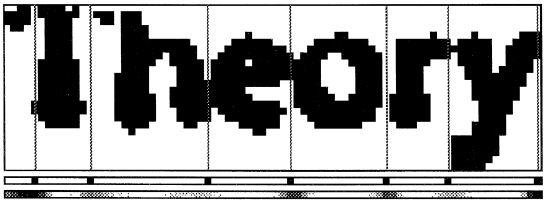

Fig. 4. Sliding window segmentation: After a normalization step, a sliding window neural network generates probability scores for segmentation points. After applying a threshold and fine adjustments the word is segmented.

6. Conclusions

Tests with two leading edge commercial software packages confirm that today's best OCR still performs poorly compared to human reading capabilities, even without the use of context. Low resolution and degraded documents are still a challenge for the OCR field. We have proposed a combined segmentation/recognition approach that makes use of space displacement neural networks and hidden Markov models. Our approach reduces by the errors produced by leading edge commercial OCR systems on low resolution and low quality documents by 50%. We believe the better performance of our system is mainly due to the combined segmentation/recognition architecture as well as for the real world training and testing data we have collected and used. In order to facilitate benchmarking with other systems and methods, we are releasing this data on the internet (http://www.sedal.usyd.edu.au).

Acknowledgements

This research is supported by a grant from the Australian Research Council (grant No A49530190).

References

Baum, L.E., Sell, G.R., 1968. Growth functions for transformations on manifolds. Pacific J. Math. 27 (2), 211–227.

Casey, R.G., Lecolinet, E., 1996. Survey of methods and strategies in character segmentation. IEEE Trans. Pattern Anal. Machine Intell. 18, 690–706.

Guyon, I., Albrecht, P., Cun, Y.L., Denker, J., Hubbard, W., 1991. Design of a neural network character recognizer for a touch terminal. Pattern Recognition 24 (2), 105–119.

Keeler, J., Rumelhart, D.E., Leow, W.-K., 1991. Integrated segmentation and recognition of hand-printed numerals. In: Advances in Neural Information Processing Systems, vol. 3. Morgan Kaufmann, Denver, pp. 557–563.

Lang, K.J., Hinton, G.E., 1988. A time delay neural network architecture for speech recognition. Tech. Rept. CMU-cs-88-152. Carnegie-Mellon University, Pittsburgh, PA.

Li, Y., Lopresti, D., Nagy, G., Tomkins, A., 1996. Validation of image defect models for optical character recognition. IEEE Trans. Pattern Anal. Machine Intell. 18 (2), 99–107.

Matan, O., Burges, C.J.C., Le Cun, Y., Denker, J., 1992. Multi-digit recognition using a space displacement neural network. In: Moody, J.E. (Ed.), Advances in Neural Information Processing Systems, vol. 4. Morgan Kaufmann, Denver, pp. 488–495.

Schenkel, M., Weissman, H., Guyon, I., Nohl, C., Henderson, D., 1993. Recognition-based segmentation of on-line hand-printed words. In: Advances in Neural Information Processing Systems, vol. 5. Morgan Kaufmann, Denver, pp. 723–730.

Seiler, R., Schenkel, M., Eggimann, F., 1996. Off-line cursive handwriting recognition compared with on-line recognition. In: Proc. ICPR-96, Vienna.