2012 10th IAPR International Workshop on Document Analysis Systems

Scanning Neural Network for Text Line Recognition

Sheikh Faisal Rashid*, Faisal Shafait† and Thomas M. Breuel*
*Department of Computer Science
Technical University Kaiserslautern, Germany
Email: rashid@iupr.com, tmb@iupr.com
†German Research Center for Artificial Intelligence (DFKI), Kaiserslautern, Germany
Email: faisal.shafait@dfki.de

***Abstract*—Optical character recognition (OCR) of machine printed Latin script documents is ubiquitously claimed as a solved problem. However, error free OCR of degraded or noisy text is still challenging for modern OCR systems. Most recent approaches perform segmentation based character recognition. This is tricky because segmentation of degraded text is itself problematic. This paper describes a segmentation free text line recognition approach using multi layer perceptron (MLP) and hidden markov models (HMMs). A line scanning neural network –trained with character level contextual information and a special garbage class– is used to extract class probabilities at every pixel succession. The output of this scanning neural network is decoded by HMMs to provide character level recognition. In evaluations on a subset of UNLV-ISRI document collection, we achieve 98.4% character recognition accuracy that is statistically significantly better in comparison with character recognition accuracies obtained from state-of-the-art open source OCR systems.**

***Keywords*-Scanning Neural Network; Multilayer Perceptron; AutoMLP; Hidden Markov Models; Optical Character Recognition; Segmentation free OCR**

I. Introduction

Optical character recognition (OCR) has been an interesting application of pattern classification and computer vision from last three decades. Recent advances in OCR research make it possible to provide high recognition accuracies for machine printed Latin script documents, but error free recognition is still not possible under moderate degradations, variable fonts, noise and broken or touching characters. Moreover, character recognition rate further decreases in case of handwritten or cursive script text. Broadly, OCR approaches can be divided into segmentation based and segmentation free approaches. Segmentation based approaches work by segmenting the text into individual characters and recognition is performed at character level. However, in case of degraded, handwritten or cursive script text, segmentation of text into characters is problematic and the performance of character segmentation significantly affects character recognition accuracies. In this paper, we present a novel segmentation free OCR approach using artificial neural networks (ANNs) and Hidden Markov Models (HMMs). We primarily focus on recognition of entire text line instead of isolated words or characters with the help of a line scanning mechanism. We train an auto-tunable multilayer perceptron (AutoMLP) [1] on possible character and non-character positions over complete text line using standard back propagation algorithm. This trained MLP model is used as a tool for predicting the class probabilities at successive positions on a given text line. The output of this line scanning neural network is a time series signal generated at each pixel transition. This time series is finally passed to a trained Hidden Markov Models (HMMs) decoder to obtain the most likely character sequence. Figure 1 outlines our approach for line scanning neural network. The system is trained and evaluated on subsets of UNLV-ISRI document collection [2]. Figure 2 presents some sample text lines taken from this document collection. We achieve significantly better character recognition accuracies in comparison to state-of-the-art open source OCR systems.

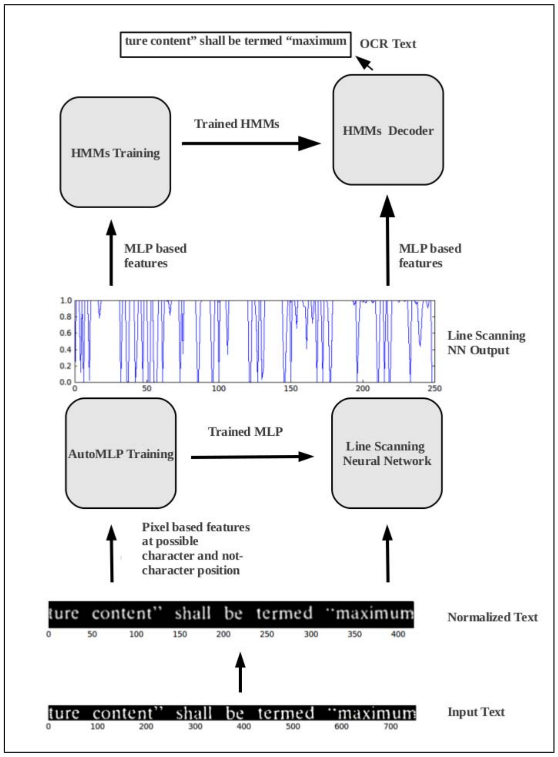


Figure 1. Line scanning neural network architecture.

978-0-7695-4661-2/12 $26.00 © 2012 IEEE
DOI 10.1109/DAS.2012.77

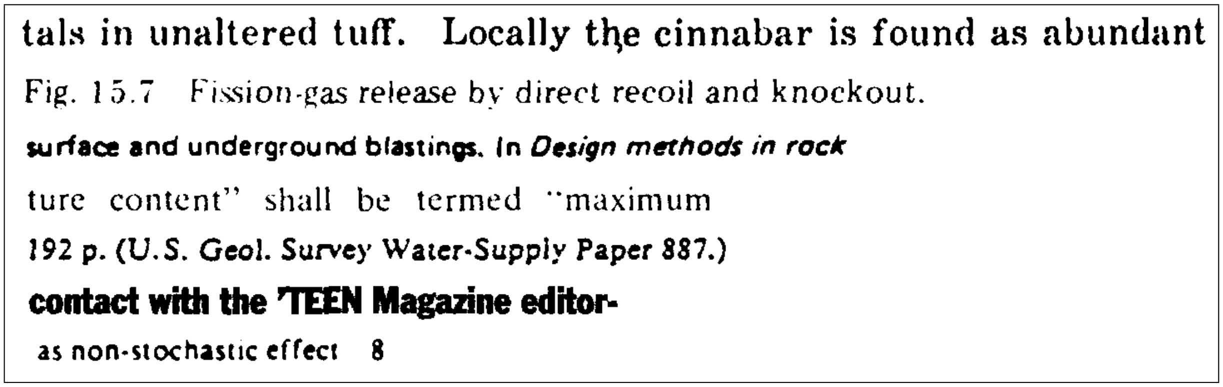

tals in unaltered tuff. Locally the cinnabar is found as abundant

Fig. 15.7 Fission-gas release by direct recoil and knockout.

surface and underground blastings. In *Design methods in rock*

ture content" shall be termed "maximum

192 p. (*U.S. Geol. Survey* Water-Supply Paper 887.)

contact with the 'TEEN Magazine editor-

as non-stochastic effect 8

Figure 2. Sample text lines from UNLV-ISRI document collection.

II. Related Work

Due to the inherent problem of segmentation in speech recognition, most of the segmentation free approaches in OCR are employed from speech recognition research. Hidden Markov Models (HMMs) [3] are very popular and are extensively applied to recognize unconstrained handwritten text or cursive scripts [4], [5], [6]. However, HMMs have drawbacks like having independent observation assumption and being generative in nature. Recurrent neural networks can be considered as alternative to HMMs but are limited to isolated character recognition due to segmentation problem [7]. Some efforts are made by Graves et. al [8] to combine the RNN with connection temporal classification (CTC) for segmentation free recognition of off-line and on-line handwritten text. Hybrid approaches, based on combination of various neural networks and HMMs have also been proposed in application to handwriting, cursive script and speech recognition. In most of the hybrid approaches [9], [10], [11], [12] a neural network is used to augment the HMM either as an approximation of the probability density function or as a neural vector quantizer. Other hybrid approaches [13], [14], [15] use the neural networks as part of feature extraction process or to obtain the observation probabilities for HMMs. These hybrid approaches either require combined NN/HMM training criteria or they use complex neural network architecture like time delay or space displacement neural networks. Recently, Dreuw et. al [16] presented a confidence- and margin-based discriminative training approach for model adaptation of a hidden Markov model (HMM)-based handwriting recognition system. Kae et. al [17] proposed an OCR approach for degraded text using language statistics and appearance features without using any character models or training data.

III. Scanning Neural Network

This section briefly describes the architecture of line scanning neural network. The system proceeds in several stages:

1) Text line normalization
2) Features extraction
3) Neural network training
4) Text line scanning
5) Hidden Markov Models decoding

A. Text Line Normalization

The first step is text line normalization. This is important because the MLP classifier takes a fixed dimensional input and text lines differ significantly with respect to skew, height and width of the characters. Printed documents originally have zero skew, but when a page is scanned or photocopied, nonzero skew may be introduced. Skew can be corrected at page level [18] but as we are working with text lines, we need to correct any possible skew for every text line before further processing. A skew angle is determined and corrected as described in [6]. After skew angle correction text lines are normalized to a height of 30 pixels. In order to normalize the text line, we first divide the text line into ascender, descender and middle or x-height[1] regions. This division is performed while estimating the base line, and x-line using linear regression. Figures 3(a) and 3(b) show the original text line and its separated regions. The height of ascender and descender regions are made equal to x-height by cropping or padding. These three regions are then rescaled separately to a height of 10 pixels (calculated as $\frac{desiredheight}{3}$) and are shown in figure 3(c). A normalized text line, as shown in figure 3(d), is obtained by combining these three rescaled regions. This kind of normalization is performed because we want to rescale the x-height of all characters to a specific height without affecting the ratio of the x-height to the body height (one of the major characteristics that defines the appearance of a typeface).

B. Features Extraction

Pixel based features are extracted from normalized text lines at possible character and non-character positions to provide positive and negative examples from training data. The

[1] http://en.wikipedia.org/wiki/X-height

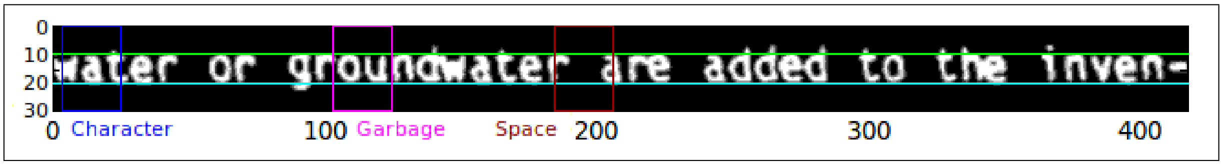
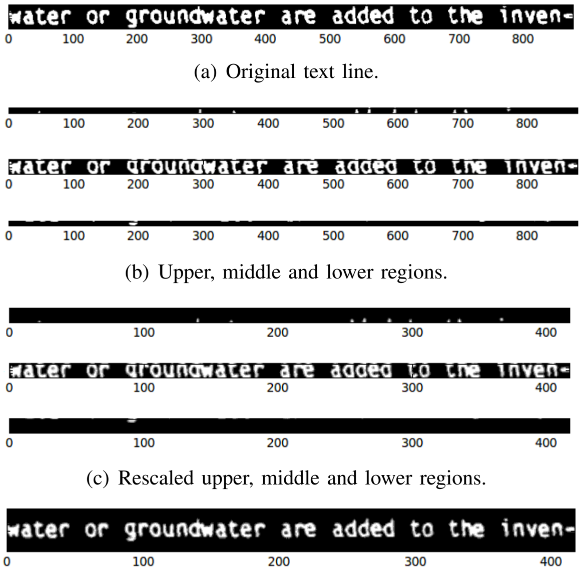


Figure 4. Example window positions for character, non-character/garbage and space. x-height is normalized to 10 pixels.



(a) Original text line.

(b) Upper, middle and lower regions.

(c) Rescaled upper, middle and lower regions.

(d) Normalized text line.

Figure 3. Text line normalization steps.

possible character positions are obtained using a dynamic programming algorithm as proposed by Breuel [19]. A mapping function β is used to provide correspondence between characters in normalized text line to the possible character position in original text line. A 30×20 $(height \times width)$ window is placed at each possible character so that the baseline is at $y = 20$ and x-height is at $y = 10$ and the character is at the center of the window. The width of the window is set to 20 pixels to incorporate the neighboring context as shown in Figure 4. This contextual window is moved from one possible character to another possible character to extract feature vectors for valid characters. Feature vectors for non-character/garbage class are obtained by placing the window at center of two consecutive characters as shown in Figure 4. Spaces are considered as valid characters and distinction between space and garbage class is made by computing the distance between two consecutive characters. If the distance is less than a specific threshold value then it is considered as a garbage, otherwise it is considered as a space. Due to variations in inter-character spaces this threshold is computed for every text line. A mean distance between all the characters in a text line is computed and a standard deviation is added to that mean. This sum of mean and standard deviation provides the threshold value for spaces. At each 30×20 contextual window, gray scale pixel values are used to construct the feature vector $x_i \in R^{600}$.

C. Neural Network Training

Artificial neural networks (ANNs) have been successfully applied for character recognition. One of the long-standing problems related to ANNs is parameter optimization, such as selection of learning rate, numbers of hidden units and epochs. To avoid these problems, we use an auto tunable multilayer perceptron (AutoMLP) [1] for training and recognition. The AutoMLP works by combining the ideas from genetic algorithms and stochastic optimization. It maintains a small ensemble of networks that are trained in parallel with different learning rates and different numbers of hidden units using gradient based optimization algorithms. After a small, fixed number of epochs, the error rate is determined on a validation set. The worst performer neural networks are replaced with copies of the best networks, modified to have different numbers of hidden units and learning rates.

The extracted features are used to train AutoMLP for 94 character classes–upper and lower case Latin characters, numerals, punctuation marks and white space– along with one extra garbage class. Hence the network has 95 output units. The activations of the output layer can be now interpreted as the probabilities of observing the valid character classes as well as the probability of observing garbage at a particular position on a text line. This leads us to the idea of line scanning neural network.

D. Text Line Scanning

The line scanning neural network works by moving a contextual window, from left to right, centered at each pixel position on a normalized text line. The output of the line scanning neural network is a vector of posterior probabilities (one element for each character class). A character sequence can also be generated by picking the most probable class from these output probabilities by detecting the local maximum (peak). Figure 5 shows an example text line and some detected peaks that correspond to specific character classes at that point. This kind of output is similar to the output generated by Graves et al. [20], [8] using RNN and CTC architecture.

E. Hidden Markov Models Decoding

Hidden Markov Models (HMMs) have been successfully applied to continuous speech, handwritten and cursive script

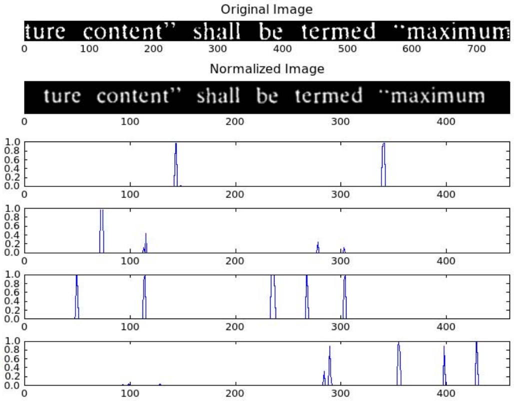


Figure 5. Local maximum (peak) detected at some character positions.

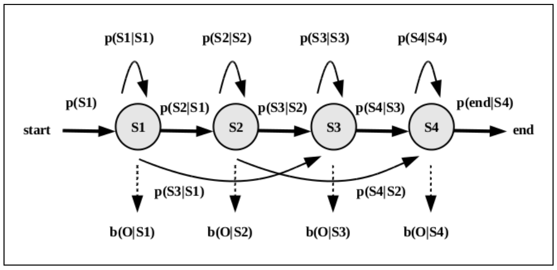


Figure 6. Example four states left to right HMM topology.

text recognition [21], [22], [6]. The basic idea is that the output of line scanning neural network can be interpreted as a left-to-right sequence of signals that are analogous to the temporal sequence of acoustic signals in speech. Therefore, the output vector generated by scanning neural network is treated as the observations for Gaussian mixture based HMMs. As the output probabilities have very skewed distribution, the probabilities are smoothed with a Gaussian kernel ($\sigma = 0.5$) and are converted to negative logs before passing them as feature inputs to HMMs.

The presented method models the character classes with multi-state, left to right, continuous density HMMs. Each character model has 10 states with 256 Gaussian mixture densities, self loops and transition to adjacent states with one skip. The number of states and mixture densities are determined empirically on a small set of validation data. Figure 6 shows an exemplary four state, left to right HMM topology. The "start" and "end" are non-emitting states and are used to provide transitions from one character model to the other character model. The text lines are modeled by concatenating these character models in ergodic structure. Training or estimating the HMM parameters is performed using Baum-Welch re-estimation algorithm [23], which iteratively aligns the feature vectors with the character models in a maximum likelihood sense.

Table I
RECOGNITION ACCURACIES FOR UNLV-ISRI SUBSET.

Algorithms	**Character Recognition Accuracies**
Line scanning NN + HMMs	**98.41%**
HMMs - Pixels	91.98%
HMMs - Intensity Features	91.62%
OCRopus	97.17%
Tesseract	97.66%
ABBYY	99.30%

IV. EXPERIMENTAL RESULTS AND EVALUATIONS

The proposed line scanning architecture is trained and tested on two different randomly selected subsets of UNLV-ISRI document collection[2]. We also evaluate the state-of-the-art open-source/commercial OCR engines and HMM based segmentation free OCR strategies [24] on the same test set. The test set consists of 1060 text lines, having 51,261 characters. The participating OCR engines are ABBYY FineReader 10 professional [25], Tesseract 3.1 OCR engine [26] and OCRopus 0.4 [27]. The performance evaluation is carried out by computing character recognition accuracy percentage (CRA%) with the help of following formula

$$CRA\% = \frac{N - ED}{N} * 100 \qquad (1)$$

where N = Total number of characters and
ED = Edit Distance = Nos. of deletions + Nos. of insertions + Nos. of substitutions (with equal cost).
The recognition results are presented in Table I. We achieve significantly better recognition accuracies in comparison to state-of-the-art open source OCR systems and HMM based techniques. ABBYY provides good result and one of the reasons could be the built-in language modeling facility. All the other systems are evaluated without language modeling support.

V. CONCLUSION

We have introduced a novel OCR approach for Latin printed text recognition using multilayer perceptron. The key features of the network are the line scanning architecture and HMMs decoding. This provides the mechanism to generate class posterior probabilities at each pixel succession, while incorporating the contextual information in discriminative learning. The output of the architecture is a time signal that is decoded by HMMs to provide character level classification of entire text line. In experiments on a subset of UNLV-ISRI document collection, the new approach outperformed state-of-the-art open source OCR systems and HMM-based systems without using any language modeling or lexicon.

[2] The dataset can be obtained by contacting the authors.

References

[1] T. Breuel and F. Shafait, “AutoMLP: Simple, Effective, Fully Automated Learning Rate and Size Adjustment,” in *The Learning Workshop*, April 2010, extended Abstract.

[2] K. Taghva, T. Nartker, J. Borsack, and A. Condit, “UNLV-ISRI document collection for research in OCR and information retrieval,” in *Proc. of SPIE Document Recognition and Retrieval VII*, vol. 3967, December 1999, pp. 157–164.

[3] L. R. Rabiner, “A tutorial on Hidden Markov Models and selected applications in speech recognition,” *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, February 1989.

[4] Jianying Hu, M. K. Brown, and W. Turin, “HMM based on-line handwriting recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 10, pp. 1039–1045, October 1996.

[5] M. S. Khorsheed, “Offline recognition of omnifont Arabic text using the HMM ToolKit (HTK),” *Pattern Recognition Letters*, vol. 28, no. 12, pp. 1563–1571, September 2007.

[6] U.-V. Marti and H. Bunke, “Using a Statistical Language Model to Improve the Performance of an HMM-Based Cursive Handwriting Recognition System,” *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 15, no. 1, pp. 65–90, February 2001.

[7] N. G. Bourbakis, “Handwriting recognition using a reduced character method and neural nets,” in *Proc. of SPIE Nonlinear Image Processing VI*, vol. 2424, February 1995, pp. 592–601.

[8] A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, and J. Schmidhuber, “A Novel Connectionist System for Unconstrained Handwriting Recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 5, pp. 855–868, May 2008.

[9] A. Brakensiek, A. Kosmala, D. Willett, W. Wang, and G. Rigoll, “Performance evaluation of a new hybrid modeling technique for handwriting recognition using identical on-line and off-line data,” in *Proc. of Fifth International Conference on Document Analysis and Recognition*, September 1999, pp. 446–449.

[10] S. Marukatat, T. Artires, B. Dorizzi, and P. Gallinari, “Sentence recognition through hybrid neuro-Markovian modeling,” in *Proc. of Sixth International Conference on Document Analysis and Recognition*, September 2001, pp. 731–735.

[11] J. H. Kim, K. K. Kim, and C. Y. Suen, “An HMM-MLP Hybrid Model for Cursive Script Recognition,” *Pattern Analysis & Applications*, vol. 3, no. 4, pp. 314–324, 2000.

[12] S. España-Boquera, M. J. Castro-Bleda, J. Gorbe-Moya, and F. Zamora-Martinez, “Improving Offline Handwritten Text Recognition with Hybrid HMM/ANN Models,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 4, pp. 767–779, April 2011.

[13] M. Schenkel, I. Guyon, and D. Henderson, “On-line cursive script recognition using time delay neural networks and hidden Markov models,” *Machine Vision and Applications*, vol. 8, no. 4, pp. 215–223, 1995.

[14] Y. Bengio, Y. Lecun, C. Nohl, and C. Burges, “LeRec: A NN/HMM hybrid for on-line handwriting recognition,” *Neural Computation*, vol. 7, no. 6, pp. 1289–1303, November 1995.

[15] S. Knerr and E. Augustin, “A neural network-hidden Markov model hybrid for cursive word recognition,” in *Proc. of Fourteenth International Conference on Pattern Recognition*, vol. 2, August 1998, pp. 1518–1520.

[16] P. Dreuw, G. Heigold, and H. Ney, “Confidence- and margin-based MMI/MPE discriminative training for off-line handwriting recognition,” *International Journal on Document Analysis and Recognition*, vol. 14, no. 3, pp. 273–288, April 2011.

[17] A. Kae, D. Smith, and E. Learned-Miller, “Learning on the fly: a font-free approach toward multilingual ocr,” *International Journal on Document Analysis and Recognition*, vol. 14, no. 3, pp. 289–301, April 2011.

[18] J. V. Beusekom, F. Shafait, and T. M. Breuel, “Combined orientation and skew detection using geometric text-line modeling,” *International Journal on Document Analysis and Recognition*, vol. 13, no. 2, pp. 79–92, June 2010.

[19] T. M. Breuel, “Segmentation of handprinted letter strings using a dynamic programming algorithm,” in *Proc. of Sixth International Conference on Document Analysis and Recognition*, September 2001, pp. 821–826.

[20] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, “Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks,” in *Proc. of the 23rd international conference on Machine learning*, 2006, pp. 369–376.

[21] F. Jelinek, *Statistical methods for speech recognition*. Cambridge, MA, USA: MIT Press, 1997.

[22] I. Bazzi, R. Schwartz, and J. Makhoul, “An omnifont open-vocabulary OCR system for English and Arabic,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 6, pp. 495–504, June 1999.

[23] S. J. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev, and P. Woodland, *The HTK Book Version 3.4*. Cambridge University Press, 2006.

[24] S. F. Rashid, F. Shafait, and T. M. Breuel, “An Evaluation of HMM-Based Techniques for the Recognition of Screen Rendered Text,” in *Proc. of Eleventh International Conference on Document Analysis and Recognition*, September 2011, pp. 1260–1264.

[25] ABBYY. http://finereader.abbyy.com/.

[26] R. Smith, “An Overview of the Tesseract OCR Engine,” in *Proc. of Ninth International Conference on Document Analysis and Recognition*, 2007, pp. 629–633.

[27] T. M. Breuel, “The OCRopus open source OCR system,” in *Proc. of SPIE Document Recognition and Retrieval XV*, vol. 6815, January 2008, p. 68150F.