Combining HMMs for the recognition of noisy printed characters

A. J. Elms and J. Illingworth
Department of Electronic and Electrical Engineering
University of Surrey
Guildford, GU2 5XH, U.K.

Abstract

In this paper, a novel method is proposed for the recognition of noisy printed characters. The method is based on the representation of the shape of a character by two Hidden Markov Models. Recognition is achieved by scoring these models against the test pattern and combining the results. The method has been evaluated using Baird's noise model, producing a peak performance of 99.5% on the test set in the presence of near-minimal noise. The method generalises to recognise characters with noise levels greater than those included in the training set, and an investigation of the top-k performance suggests that a much higher recognition rate could be achieved on language text using a context driven word recogniser.

1 Introduction

Methods for the classification of printed character images largely fall into two groups: those which are rely on structural analysis, and those which are based on template matching. Structural methods have the advantage of representing the pattern in an abstract form (a combination of strokes) which lends itself to polyfont recognition whereas template matching methods are generally based on a more explicit representation of the pattern. Structural methods can fail when pixels introduced by a noise process either split or connect strokes.

Many authors claim performance figures for such methods but without common databases of printed material they are difficult to compare. It is also often the case that results are reported for noise-free images which are generated on clean paper by a high-quality laser printer and are immediately and carefully scanned in. Such images rarely occur in real applications due to the use of photocopiers and fax machines, as well as the effects of handling and ageing. However, it is unusual to see a quantitative evaluation of the performance of a character recogniser in such conditions.

Baird has defined a model representing defects typical of document images [1] and the use of the model to generate artificial noisy images for the training and testing of a polyfont classifier is reported in [2].

In this paper a character recogniser is described which uses a novel combination of two Hidden Markov Models (HMMs) to capture an abstract representation of a training set of character images. This method is applicable to polyfont recognition but does not rely on segmenting each pattern into strokes and is therefore tolerant to noise. In a single-font application a peak recognition rate of 99.5% is achieved, and Baird's Defect Model is used to quantify how the performance varies with the amount of noise present in the image.

BMVC 1994 doi:10.5244/C.8.18

1.1 Previous work

Although much work on HMMs has been done in the field of Speech Recognition [3]-[6], few researchers have applied them to text recognition. Anigbogu and Belaïd [7]-[9] claim performance of between 96% and 98.65% on a range of fonts. A limitation of their method is that the characters have to be segmented before a number of features (such as number of black-white transitions and aspect ratio) are extracted. These features are ordered heuristically and it is this series of values which is used as the observation sequence which is scored for each of a set of competing models. The organisation of the features as a sequence is arbitrary and so there is no obvious explanation as to why there should be any Markov dependency in the sequence. Vlontzos and Kung [10] use a similar system of competing HMMs, with 95% accuracy, but use a parameterisation of structural primitives such as strokes and arcs as the observation sequence. These are ordered according to a traversal of the skeleton of a segmented character which is guided by an analysis of junction points in the skeleton. This method is also limited in that it requires the prior segmentation of the characters. Also, the proposed method of structural analysis is unlikely to be reliable in the presence of noise.

A method which does not rely on a prior segmentation into characters is that of Bose and Kuo [11]. In a method analogous to a recognition system for connected spoken words described by Rabiner and Levinson [12], a number of competing HMMs are used to recognise characters based on an observation sequence derived from a clustering of feature vectors from small segments of each word. A Level Building search procedure is used to produce the maximum likelihood character sequence at the same time as the segmentation, making it suitable for connected and degraded character images.

The method described here is similar to that of Bose and Kuo in that a segmentation of a printed word into its constituent characters is not required prior to recognition. However, our method does not require a complex structural analysis of the input image, and the training and recognition require no manual intervention. Each character is represented by an 8-state HMM whereas Bose and Kuo select the number of states for each character model according to its segmentation into strokes. Further, a full characterisation of the performance of the algorithm on isolated characters has been undertaken to determine its peak performance as well as its susceptibility to document image noise.

2 HMM Recogniser

A discrete HMM is a representation of the statistics of a random process which produces discrete observations $O(t)$ at sampling instants $t = 1, \cdots, T$. If such a sequence is observed, a model C can be "scored" in terms of a likelihood that the process which C represents could have produced the sequence. Given a number of sequences, each one derived from one of Q classes of pattern, and a set of models $C_q, q = 1, \cdots, Q$ where each model C_q has been trained on pattern class q, it is possible to classify the sequences by scoring all models against each sequence and choosing the model with the highest likelihood score. Thus the HMM is a natural mechanism for recognising a pattern which can be represented as a sequence of discrete observation symbols.

Consider the character **m** depicted in figure 1. Taking horizontal single pixel-width scan lines down the image, it is clear that the pattern is characterised by observing a few lines with long runs of ON pixels, followed by a large number of lines with three short

runs of ON pixels. When scanned vertically, a similar generalisation can be made about the pattern: three sets of lines with long single runs of ON pixels are separated by two sets of lines with short single runs. The HMM appears to be a natural way of expressing this profile of image features - the presence of a feature depends on those immediately prior to it but not on features much earlier in the profile (a low-order Markov dependency).

2.1 Feature extraction

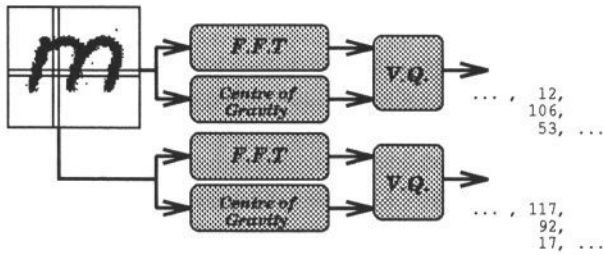


Figure 1: Feature extraction process.

The assumption outlined thus far is that a character image can be represented by the profile generated by observing a sequence of horizontal or vertical scan lines through the image. Furthermore, for a 64 × 64 pixel binary image (for example) it is unlikely that there will be 2^{64} possible scan lines observed in either the horizontal or vertical profiles generated for a set of images. The patterns observed in the scan lines therefore lend themselves to quantisation, such that a pattern is represented by one entry from a codebook of possible patterns, where the entry is chosen as the closest to the pattern, in some sense. The codebook design is discussed further in the next section.

It is not possible directly to cluster a training set of scan lines using a simple distance measure. Consider two patterns taken from scans through an image. They may be identical, save for a one pixel offset in the direction of the scan. This offset introduces a large distance between the patterns, which is counter-intuitive to the assumption of how the scan lines represent the image. It is therefore desirable to represent the scan lines in a way which is independent of the absolute location of the character within the image window. By transforming the scan line to produce the magnitude of its complex 1-dimensional Fourier Transform (using a Blackman window and a 64 point FFT), the resulting representation is shift invariant, and due to the symmetry of the spectrum has halved the number of dimensions of the vector - a 64 pixel scan line becomes a 32 dimensional spectrum. If the features extracted from the vertical scan line were completely shift-invariant, however, the characters **b** and **p** would be very similar in feature space. Therefore a measure of the centre-of-gravity (COG) relative to the running average of the COG of previous scan lines is incorporated as the 33rd dimension of the feature vector. This is designed to emphasise large changes of COG with respect to previous lines whilst reducing the effect of small differences which may be due to noise. The feature vector for scan line i is therefore:

$$\begin{aligned} F_i(k) &= \sqrt{\mathcal{R}_i(k)^2 + \mathcal{I}_i(k)^2} \qquad k = 0, \cdots, 31 \\ F_i(32) &= \left(\frac{\text{COG}_{i-2} + \text{COG}_{i-1} + \text{COG}_i}{3} - \text{COG}_i \right)^3 \end{aligned} \tag{1}$$

where $\mathcal{R}_i(k)$ and $\mathcal{I}_i(k)$ are the real and imaginary components of the complex spectrum of scan line i and cog_i is its centre of gravity.

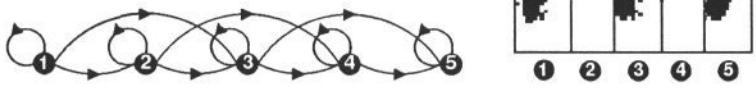
2.2 Vector Quantisation

The result of the feature extraction defined in equation 1 is a continuous 33-dimensional vector representing each vertical and horizontal scan line through the image. The HMMs are used to recognise a sequence of discrete symbols, so it is necessary to quantise the feature vectors such that each one can be represented by a symbol representing the closest vector contained in a codebook.

The codebook must first have been created to closely approximate the feature vectors in a training set. The training set is clustered according to the *Post Transfer Advantage rule* of Kittler and Pairman [13]. Clusters are represented as normal distributions in 33-D and cluster sizes are taken into consideration. This gives an improved partition of the data set when compared to a simple *k-means* approach. The result of this clustering is a codebook which represents each cluster by its mean vector, covariance matrix and number of elements. During the recognition process, the feature vector calculated for each scan line of the test image is used as the key to a linear search of the codebook which produces an observation symbol according to the best match of the vector against the cluster parameters.

2.3 Model Training and Scoring

A set of example images is required for the training of the models for each of the character classes to be recognised. Features are extracted as described above, resulting in a set of sequences of symbols representing the horizontal and vertical profiles of the training images. A HMM can then be trained (using the *Baum-Welch reestimation procedure* [6]) to represent the statistics of the process which generates the training sequences. This means that each class is represented by two models - one representing possible vertical profiles and one representing horizontal profiles.



(a) State transition diagram

(b) Possible state assignment

Figure 2: State assignment of HMM

The HMM models a sequence as the output of a finite state automaton, in which a

transition from state i to state j occurs according to a probability a_{ij} at each time instant, and an output is observed based on a probability distribution $b_j(s)$ associated with the state j the model is in at that time instant. Figure 2 (a) shows a typical state transition diagram. The Markov dependence is between state transitions - the probability of a transition from state i to state j depends only on the values i and j (a first-order Markov assumption) and not on any previous transitions. The result of training on vertical scan lines can be envisaged as shown in figure 2 (b) - the model will partition the sequence of observations between its states such that in state 1 symbols representing long single runs of ON pixels are more likely to be observed, whereas in state 2 short single runs are more likely.

During recognition, a Viterbi scoring method is used to find the likelihood score $L(q)$ that each model C_q could have produced the given observation sequence. Each model is matched against the observation sequence beginning at sample number $t = 1$. Initialisation is:

$$\delta_1(1) = \left[b_1^q(O_1)\right] \tag{2}$$

Recursion is for $2 \leq t \leq T, 1 \leq j \leq N$:

$$\delta_t(j) = \max_{1 \leq i \leq N} \left[\delta_{t-1}(i) a_{ij}^q\right] . \left[b_j^q(O_t)\right] \tag{3}$$

Termination occurs when:

$$L(q) = \delta_T(N) \tag{4}$$

The matching of the observation sequence to a character model can exhibit extreme time-warping properties as it is possible for a short observation sequence to be generated by the model of a character which normally produces a long sequence. This is due to the long forward state transitions as depicted in figure 2 (a). Rabiner and Levinson reduce this effect in their spoken word recognition system [12] by heuristically incorporating the probability of word duration. It has also been possible to model printed character width as a normal distribution, deriving μ_q and σ_q (in number of pixels) from training data.

The probability of model C_q matching an observation sequence of T samples is:

$$P_D(T) = \frac{\exp - \left[\frac{(T-\mu_q)^2}{2\sigma_q^2}\right]}{\sqrt{2\pi\sigma_q}} \tag{5}$$

Thus the likelihood score $L(q)$ can be modified to account for the character width probability $P_D(T)$:

$$\tilde{L}(q) = L(q).(P_D(T))^{\gamma} \tag{6}$$

where γ is a weighting factor which is optimised experimentally. A value of $\gamma = 3.0$ was used in the experiments described in this paper, although the results were not particularly sensitive to the value of γ.

3 Combining Recognisers

By using the scoring mechanism described in section 2.3 a likelihood can be computed for each class q that the model C_q could have produced a given input pattern. Strictly

speaking there are two likelihoods, $\tilde{L}_V(q)$ and $\tilde{L}_H(q)$, relating to the models of the vertical and horizontal profiles respectively. It will be seen experimentally that choosing the maximum likelihood model from either of these "experts" produced a good recognition rate, but clearly combination of the evidence from both experts is likely to produce a better overall result. One solution to this problem is to compute:

$$\hat{L}(q) = \tilde{L}_V(q) \times \tilde{L}_H(q) \tag{7}$$

Equation 7 is only true given that $\tilde{L}_V(q)$ and $\tilde{L}_H(q)$ are statistically independent. This is clearly not the case - since either expert in isolation can achieve a good recognition rate, then there must be a significant correlation between the experts. However, it will also be shown later that in practice equation 7, an approximation to the real situation, can be used to combine the likelihoods from the two experts. This results in an improved recognition rate than either could achieve in isolation. Clearly a better combination rule, which exploits the dependence between the two experts, is a subject of future investigation.

As all three likelihoods, $\hat{L}(q)$, $\tilde{L}_V(q)$ and $\tilde{L}_H(q)$ are computed for each class q, it is possible to rank the likelihood (whether from an individual expert or from the combined result) in order to produce a number of ranked alternative recognition results. A *top-k* recognition result can therefore be computed for this method. This is the percentage of test cases where the correct class label appears in the first k choices output by the recogniser. This figure indicates how well the recogniser would perform given further contextual information - if, for example, the top-3 performance is significantly better than the top-1, then a bottom-up contextual knowledge source (such as the probabilities of transitions between letters) [14] is expected to improve the performance of the recogniser on running text.

4 Experiments

The experiments reported here are intended to demonstrate the recognition performance of the algorithms described, and particularly to quantify how the performance degrades in the presence on document image noise. A single point size was used *(11pt)* and a single font *(Times-Roman)*. Although this is clearly not a complete test set, it is not unreasonable: Times-Roman is a common font for body-text and *a-priori* knowledge of the point size can be feasibly determined during the page segmentation process from the inter-line spacing. This limitation in typography was necessary due to the large amount of computation and storage required for artificially generating the noisy images required to quantify the performance in the presence of noise.

4.1 Noise Model

Baird's model of document image defects facilitates production of artificial images which are corrupted by typical document image noise. Each output image takes as its basis a noise-free input image which is then corrupted by noise characterised by a number of model parameters. These parameters can themselves vary from image to image according to a pseudo-random process. The model has ten parameters quantifying *Resolution*, *Blur*, *Threshold*, *Sensitivity*, *Jitter*, *Skew*, *Width*, *Height*, *Baseline* and *Kerning*. The Mahalanobis Distance (MD) of the model parameters from the mean of the defect distribution

can be used as a single measure of image quality - the larger the MD the more deformed the image. Note that *Baseline*, *Jitter* and *Kerning* are not included in the MD calculation.

The model was implemented to reflect the description in [1] as closely as possible. The GNU project PostScript interpreter *"GhostScript"* was used to generate a bitmap image for each character at 300DPI resolution. The *Skew*, *Width* and *Height* variations were then applied to the bitmap. This was done by means of an inverse transformation - each pixel in the output image was mapped back to its location in the original image under the inverse transform. In general this point would lie between four pixels on the original image, so the output pixel level was found by linear interpolation. The resulting image was then oversampled by a factor of 8 and the output pixel centres were computed according to the Jitter and Kerning parameters. The oversampled image was then blurred by a circularly symmetric Gaussian filter with a standard error σ_{filt} set by the *Blur* parameter. The extent of the filter was limited to $\pm\ 3\sigma_{\text{filt}}$. The filter was centred on each output pixel centre, and the result of the convolution was normalised by dividing by the sum of the weights computed for a filter with the mean *Blur* of 0.7. The output was adjusted according to the Sensitivity parameter and was finally binarised by comparison with a threshold. A value of 0.43 was selected as optimal for a *Blur* of 0.7. It was noted that for a *Blur* parameter selected from a normal distribution with ($\mu_{\text{blur}} = 0.7$, $\sigma_{\text{blur}} = 0.3$), it is necessary to truncate the distribution in order that the *Blur* value is positive. In fact it was observed that *Blur* must be ≥ 0.37 for the output image to be non-empty.

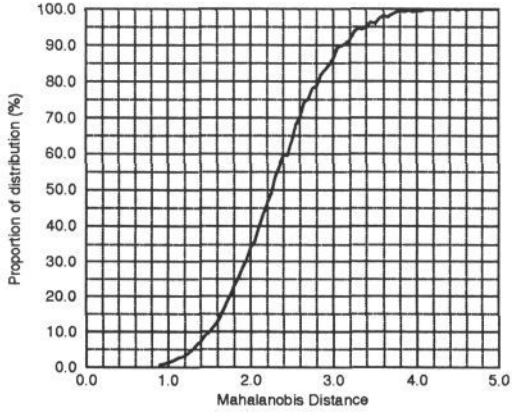


Figure 3: Coverage of Defect Model

The Mahalanobis Distance, used as an overall measure of image quality, can be related to the population upon which the model was based by the graph of figure 3. The percentage coverage of the distribution was found experimentally by successively generating model parameters, and calculating $\%_x$ as the number of generated parameter sets produced such that 1000 of the sets had a MD in the range $[0, x]$, divided into 1000. The range $[0.0, 2.0]$ can be seen to cover the best quality 35% of the population and the range $[0.0, 2.6]$ covers 70% of the population.

4.2 Results

The experimental procedure consists of three main stages. A codebook must first be generated - this is the most time-consuming stage. Then, the codebook is used to extract feature sequences from a set of training images, and these sequences are used to train the HMMs representing each class. Finally features are extracted from a test set of images using the same codebook, and the models are scored against the test sequences in order to produce a ranked recognition result. The details of the experiments can be summarised as follows:

Training Data Set: 200 examples of each character **a-zA-Z0-9** were generated by the noise model with MD limited to the range $[0.0, 2.0]$.

Test Data Set: 100 examples of each character were generated by the noise model for each of the 13 MD ranges $[0.2, 0.4], [0.4, 0.6], \cdots, [2.6, 2.8]$.

Codebook: Generating a codebook from the entire training set would be far too computationally expensive, so the noise model was used to produce 10 images of each character in each of the MD ranges $[0.0, 0.5], [0.5, 1.0], [1.0, 1.5], [1.5, 2.0]$. A composite image of these 2480 characters was used to produce a codebook of 150 clusters in 33 dimensions. Separate codebooks were created for horizontal and vertical scan line quantisation.

HMMs: Models were trained over a maximum of 30 iterations using 8 states. (These figures have been optimised experimentally). Two models per class represented horizontal and vertical profiles respectively.

Recognition was attempted for both the training and test sets, using 3 methods: horizontal profile model only, vertical profile model only and the combined result of these experts according to equation 7. As the training set was generated with a wide range of noise parameters, its recognition rate represents an "average" figure whereas recognition rates for the test sets apply to narrow bands of noise parameters. The results are summarised in figure 4.

In all cases, the peak recognition rate of the test set corresponds to a MD of the noise parameters in the range $[0.4, 0.6]$. It is interesting that this does not correspond with the absolute lowest noise levels - this reflects the fact that the models are trained over a wide range of noise. It can clearly be seen that the result of combining the horizontal and vertical profile recognisers is consistently better than either in isolation. It is also interesting to note that performance of the horizontal profile recogniser (which takes vertical scan lines through the image) is considerably better than that of the vertical profile recogniser. It could be suggested that as we read text in a left/right direction, the discriminant information in our alphabet is greater in the horizontal profile than the vertical.

It is possible to draw some conclusions on the generalising power of the recogniser. Although trained on the noise range $[0.0, 2.0]$ which covers the top 35% of the population, when tested over the range $[0.2, 2.8]$, with over twice the population coverage, the recognition rate declined gracefully, suggesting good generalisation.

The recogniser also exhibits promising behaviour for the recognition of language text using contextual information - if the top choice is not correct then the correct answer is likely to be ranked highly. Figure 5 shows the top-k results. Given the font under test, it is not surprising that confusions such as those between the characters **l I 1** (ell, EYE, one)

Scan Direction	Test Set (Peak % correct)	Training Set (% correct)
Horizontal	92.9	86.6
Vertical	97.8	95.4
Combined	99.5	98.0

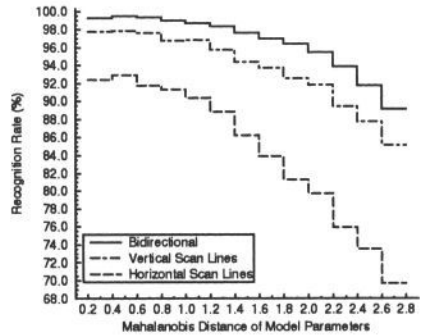


Figure 4: Performance of Individual and Combined models

occur. The top-k performance suggests that a context driven recogniser would be able to recognise printed words with a much higher success rate than that of the isolated character recogniser.

Top-k performance	$k = 1$	$k = 2$	$k = 3$
Test Set (Peak)	99.50	99.97	100
Training Set	98.01	99.34	99.65

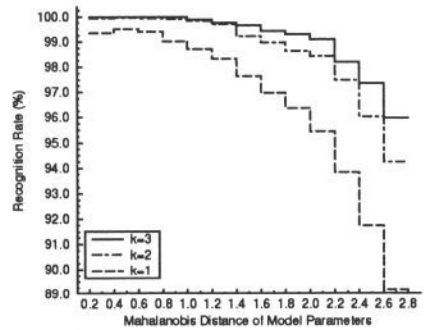


Figure 5: Top-k Performance

5 Conclusions

In this paper, a novel method has been outlined for the recognition of printed characters. The method is based on finding "profiles" of an image by approximating raster scans of the image to a sequence of discrete symbols. It has been shown how two orthogonal profiles taken from a training image can represent the shape of a character, using Hidden Markov Models.

Our method, although not optimised for speed, can recognise characters at a rate of approximately one per second. The main computational requirement is for the Vector Quantisation. The VQ process and also the scoring of the HMMs are highly parallel in nature. A commercially feasible implementation would therefore be possible through the exploitation of such parallelism or through the use of hardware accelerators for the computationally expensive tasks.

The use of the models for classifying test images has been studied in the presence of typical document imaging defects, enabling a performance figure to be measured given set noise conditions. A model of such imaging defects has been used such that the level of noise in the test condition is itself quantifiable.

Acknowledgements

Thanks to J. Kittler for help with the Vector Quantisation. Also to P. Hoad, M. Petrou, and G. Matas for advice on the implementation of Baird's Defect Model.

References

[1] H. Baird, "Document Image Defect Models," in *Structured Document Image Analysis* (H. Baird, H. Bunke, and K. Yamamoto, eds.), Springer-Verlag, 1992.

[2] H. Baird and R. Fossey, "A 100-Font Classifier," in *Proceedings of 1st International Conference on Document Analysis and Recognition*, pp. 332–340, 1991.

[3] L. Rabiner and B. Juang, "An Introduction to Hidden Markov Models," *ASSP Magazine*, Vol. 3 (1), pp. 4–16, 1986.

[4] S. Levinson, L. Rabiner, and M. Sondhi, "An Introduction to the Application of the Theory of Probabilistic Functions of a Markov Process to Automatic Speech Recognition," *AT&T Technical Journal*, Vol. 62 (4), pp. 1035–1074, 1983.

[5] L. Rabiner, "Mathematical Foundations of Hidden Markov Models," *NATO ASI Series*, Vol. F46, pp. 183–205, 1988.

[6] L. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proceedings of the IEEE*, Vol. 77 (2), pp. 257–286, 1989.

[7] A. Belaïd and J. Anigbogu, "Text Recognition Using Stochastic Models," in *Proceedings of 5th International Symposium on Applied Stochastic Models and Data Analysis*, pp. 87–98, 1991.

[8] J. Anigbogu and A. Belaïd, "Application of hidden Markov models to multifont text recognition," in *Proceedings of 1st International Conference on Document Analysis and Recognition*, pp. 785–793, 1991.

[9] J. Anigbogu and A. Belaïd, "Performance Evaluation of an HMM Based OCR System," in *Proceedings of 11th International Conference on Pattern Recognition*, pp. 565–568, 1992.

[10] J. Vlontzos and S. Kung, "Hidden Markov Models for Character Recognition," *IEEE Transactions on Image Processing*, Vol. 1 (4), pp. 539–543, 1992.

[11] C. Bose and S. Kuo, "Connected and Degraded Text Recognition Using Hidden Markov Model," in *Proceedings of 11th International Conference on Pattern Recognition*, pp. 116–119, 1992.

[12] L. Rabiner and S. Levinson, "A Speaker-Independent, Syntax-Directed, Connected Word Recognition System Based on Hidden Markov Models and Level Building," *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol. 33 (3), pp. 561–573, 1985.

[13] J. Kittler and D. Pairman, "Optimality of reassignment rules in dynamic clustering," *Pattern Recognition*, Vol. 21 (2), pp. 169–174, 1988.

[14] R. Shinghal and G. Toussaint, "Experiments in Text Recognition with the Modified Viterbi Algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 1 (2), pp. 184–193, 1979.