Cursive Word Recognition Using a Novel Feature Extraction Method and a Neural Network

José Ruiz-Pinales and René Jaime-Rivas
University of Guanajuato
36730 Salamanca, Gto. Mexico
pinales@salamanca.ugto.mx

Abstract

In this paper, we present a holistic system for the recognition of cursive handwriting that utilizes a novel feature extraction method and a neural network. The Hough transform is a global line detection technique with the ability of extracting directional information presenting good tolerance to disconnections and noise and a moderate tolerance to distortion. However, its global nature is also its weakness because it does not capture well the information contained in the body of cursive words. For this reason, we have modified and reformulated this method as the correlation of the image with template line segments which is able to extract more local features while preserving the advantages of the previous method. In fact the new method is more general in the sense that it can also be used to detect complex features such as closed loops, cavities and to estimate the average stroke width. An important result is that this new method is also equivalent to a sigma-pi network closely connected with the well known feed-forward model of orientation selectivity [5]. Results obtained on a task of cursive word recognition indicate that the new method clearly outperforms the old method.

1. Introduction

The recognition of cursive handwriting is a problem with important implications for industrial applications such as the automatic processing of checks and postal mail. Though, important progress in this area has permitted the development of some practical systems, recognition rates comparable to human performance have not yet been attained. For this reason, the problem continues to be an interesting subject of research.

Extensive research has revealed that the recognition of cursive handwriting is a very complex task. This complexity is due to several factors: cursive characters present an important amount of deformation which causes ambiguous recognition, a character can be written in different ways, and the image may contain partial strokes belonging to adjacent characters, as well as ligatures and noise. For this reason, several feature extraction techniques have been proposed in order to capture the distinctive features of handwriting, [13]. These feature extraction techniques are generally classified as: global transformations and series expansions, features derived from the statistical distribution of points, and geometrical and topological features. In general, these techniques extract local and/or global features from the image.

Due to the complexity of the cursive handwriting recognition problem, the extracted features must be tolerant to the deformation of the character stroke and easy to detect with accuracy. Topological and geometrical features are very tolerant to deformation and style changes. On the other hand, the techniques based on the statistical distribution of points are capable of tolerating a moderate amount of deformation.

The Hough transform is a global technique that was originally developed for line detection in pictures. This transformation exhibits some interesting properties: information preservation, robustness to disconnections of the strokes, noise and stroke width variation.

This technique has been used in a variety of applications, [2, 9, 10]. These applications include the detection of ascenders and descenders, the estimation of the baseline angle of cursive words, the detection of the skew angle of handwritten documents, etc. Some approaches rely on this technique for the recognition of cursive handwriting. However, its use is mostly limited to certain kinds of handwriting where the strokes are line-like since, in these approaches, the strokes are extracted from a binary image by peak finding.

Experimental studies have evidenced the importance of directional features for the recognition of objects in the human visual system, [5]. Some approaches for the recognition of handwritten digits have shown the effectiveness of this kind of features. In general, these features are extracted from the contours either by using

the chain-code or the orientations of the gradient of the image, [11, 12]. However, for cursive handwriting, the deformation and noise present in the contours may degrade seriously the performance of these local methods. Thus we are forced to recur to the use of more robust global methods such as the Hough transform. In fact, the Hough transform can be used to extract global directional features having good tolerance to disconnections and noise, and moderate distortion.

This ability of the Hough transform for capturing the global directional information of the image and its robustness to disconnections of the strokes and noise has already been exploited for feature extraction [6, 7]. However, instead of searching for linear strokes in the image, in our previous approach we computed global directional information at the pixel level. This global directional information was stored into several feature maps in order to avoid assigning to each pixel a single orientation and to preserve useful spatial information. In addition, because the Hough transform does not capture well the information contained in the body of the word, a method based on the chain code of the contours was used in order to obtain a feature map containing closed loops. Then, each feature map was processed zone by zone in order to obtain an estimate of the local orientations of the strokes and to obtain a feature vector of reduced dimension suitable for recognition by means of a neural network classifier. This method has been successfully applied to the holistic recognition of cursive words.

In this paper, we present a more general method which is able to extract more directional information from the image and which can also be used to extract closed loop, cavity and stroke width information. We show that the features extracted by this new method present a good tolerance to stroke disconnections and noise and that the extracted closed loop features present a good tolerance to distortion. Unlike other methods, this new method operates on gray-level and/or binary images.

2. Description of the system

The system, see Figure 1, comprises four stages: preprocessing, feature extraction, sub-sampling and recognition.

The preprocessing stage receives as input a binary or gray-level image and yields as output a fixed size gray-scale image.

The feature extraction stage receives as input a fixed-size gray-scale image and gives as output several feature

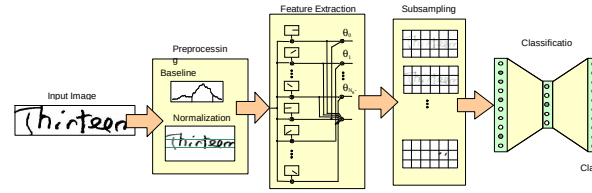

Figure 1. Architecture of the system.

maps. The sub-sampling stage processes each feature map zone by zone and yields as output a feature vector. Thus, this feature vector contains local directional features as well as closed loop features computed in several zones of the image.

The recognition stage consists of a three-layer neural network classifier. It receives as input a feature vector and yields as output a list of word class confidences. Input nodes corresponding to directional features are fully connected to a special group of neurons in the hidden layer and input nodes corresponding to closed loop features are fully connected to a distinct group of neurons in the hidden layer. The outputs of the hidden layer units are fully connected to the output layer. The number of input units is equal to the number of components of the feature vector. The number of output units is equal to the number of classes. The network is trained to approximate the posterior probabilities $P(class = i|F)$ by minimizing the cross-entropy error function.

3. Feature extraction

The Hough transform is a technique devised for detecting line segments in images, [1]. In this technique, a line is represented in its normal form by:

$$\rho = x\cos\theta + y\sin\theta$$

For a given point (x,y), this equation represents a sinusoid in parameter space. Thus, trajectories corresponding to collinear points will intersect at a common point (ρ,θ). The detection of lines in an image is then reduced to finding these intersection points.

Let $i(x,y)$ be a gray-scale image containing a cursive word. The Hough transform $h(\rho,\theta)$ of the image $i(x,y)$ can be written as:

$$h(\rho_i,\theta_j) = \sum_{x,y} i(x,y)\delta(\rho_i - x\cos\theta_j + y\sin\theta_j)$$

where $\delta(t) = 1$ for $t = 0$ and zero otherwise, $\theta_k = \frac{\pi k}{N_\theta}$, $\rho_i = \frac{iR}{N_\rho}$, $k = 0,1,\ldots,N_\theta - 1$ and $i = 0,1,\ldots,N_\rho - 1$.

3.1. Extraction of directional features

Directional feature maps can be extracted from the Hough transform by scanning the Hough transform along the same sinusoidal trajectories each black pixel $i(x,y)$ contributed to. This implicates that we are indirectly computing the inverse Hough transform. The Hough transform is a variant of Radon transform which can be computed by using efficient methods such as the fast Fourier transform [8]. In this case, the computation of the Hough transform consists in sampling the Fourier transform of the image in polar coordinates and then computing the inverse Fourier transform for each orientation. Finally, directional features can be obtained by scanning the Hough transform along sinusoidal trajectories.

For the case of grey-level images, the previous method may become time consuming because we must scan the Hough transform for each pixel of the image. A solution to this problem is to compute the directional information directly from the Fourier transform of the image. With regard to this, it can be shown that sub-sampling the Fourier transform in polar coordinates is equivalent to the correlation of the image with template lines. In fact, the correlation of the image with template lines can be used to detect lines or more complex patterns, e.g. letters [6]. Thus, the extraction of directional features can be made by computing the correlation of the image with template lines at different orientations. Our feature extraction method is as follows.

Let $i(x,y)$ be the grey-level value of a pixel of coordinates (x,y) and $I(u,v)$ be the Fourier transform of the image. We compute for each orientation $\theta_k = \frac{\pi k}{N_\theta} | k = 0,1,\ldots,N_\theta - 1$ a feature map $m_{\theta_k}(x,y)$ by using:

$$m_{\theta_k}(x,y) = \left| \Im^{-1} \left\{ I(u,v) T_{\theta_k}(u,v) \right\} \right| \qquad (1)$$

where

$$T_\theta(u,v) = \delta(-u\sin\theta + v\cos\theta) \qquad (2)$$

with

$$\delta(t) = \begin{cases} 1 & \text{if } t = 0 \\ 0 & \text{elsewhere} \end{cases}$$

is the complex conjugate Fourier transform of a line oriented at an angle θ_k and centered about the origin.

Each feature map is practically the same as the one computed directly from the Hough transform. Thus, the features extracted by using the new method also present the same tolerance to disconnections and noise. However, this method (as well as the one based on the Hough transform) is unable to extract all the directional information contained in the body of cursive words because of the high number of pixels aligned along the baseline.

In order to improve the extraction of directional features in the body of cursive words, we have modified equation 2. Thus, we have defined the kernel T_θ as the Fourier transform of a line segment of length λ centered about the origin. In this case we have used the following formula:

$$T_\theta(u,v) = \frac{\sin(\pi\lambda(u\cos\theta + v\sin\theta))}{\pi\lambda(u\cos\theta + v\sin\theta)} \qquad (3)$$

Figure 2(b) shows examples of the feature maps obtained by using this formula. As we can see, it is able to extract more of the directional information contained in the body of cursive words.

3.2. Extraction of closed-loop features

Another problem that we had to solve is that closed loops cannot be easily detected in gray-level images. This is because most detection algorithms require a binary image in order to perform a serial scan. For instance, the ray intersection method is based on counting the number of intersections at several orientations around a given pixel. In this case, we have found that it is not really necessary to count the number of intersections. An alternative is to consider that a pixel belonging to a closed loop is surrounded by the stroke at all directions. Thus, in order to determine if a given pixel belongs to a closed loop (and maybe partially open loop) we must verify that the sum of pixels aligned around that pixel (and within a certain distance) is not cero for all directions. In fact, computing the sum of pixel intensities aligned at a given direction is equivalent to computing the correlation of the image with a template line segment oriented at given angle and with one end located at the origin.

Let $i(x,y)$ be a grey-level image and $I(u,v)$ be its Fourier transform. We compute for each orientation $\theta_k = \frac{\pi k}{2N_\theta} | k = 0,1,\ldots,2N_\theta - 1$ a feature map $b_{\theta_k}(x,y)$ by using:

$$b_{\theta_k}(x,y) = \left| \Im^{-1} \left\{ I(u,v) H_{\theta_k}(u,v) \right\} \right| \qquad (4)$$

where

$$H_\theta(u,v) = \frac{\sin(\pi\lambda(u\cos\theta + v\sin\theta))}{\pi\lambda(u\cos\theta + v\sin\theta)} e^{j\pi\lambda(u\cos\theta + v\sin\theta)} \qquad (5)$$

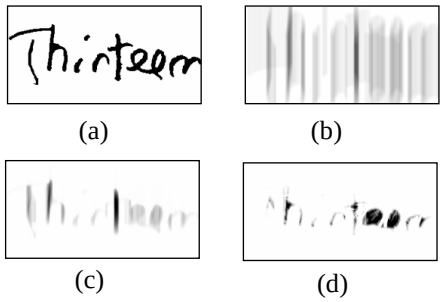

Figure 2. Examples of feature maps. (a) An input image. (b) and (c) feature maps (corresponding to vertical strokes) obtained by using the sum and the geometric mean of directional features. (d) The feature map corresponding to closed-loops for *W* = 4.

is the complex conjugate Fourier transform of a line segment of length λ oriented at an angle θ_k and with one end located at the origin.

Now, in order to determine if a given pixel belongs to a closed loop (or partially open loop) we compute a feature map $h(x,y)$ by using

$$h(x,y) = \left(\prod_{k=0}^{2N_\theta - 1} g(b_k(x,y)) \right)^{1/2N_\theta} \qquad (6)$$

where $g(\cdot)$ is a saturating nonlinearity. One expression for this function is given by:

$$g(z) = \begin{cases} 1 & \text{if } z \geq W/\lambda \\ \lambda z / W & \text{otherwise} \end{cases}$$

where *W* is the average stroke width.

Figure 2(d) shows examples of the closed-loop feature maps obtained by using this formula. As we can see, it is able to extract all closed loops. It is even capable of extracting partially open loops. This is because we use features that present a good tolerance to disconnections and noise. Loops are detected disregarding their shape which indicates that these features must be also resistant to deformation.

3.3. Generalization

Interestingly, the method for extracting directional features can be unified with the method for extracting closed-loop features because we can also obtain the feature maps $m_k(x,y)$ by using:

$$m_k(x,y) = b_k(x,y) + b_{k+N_\theta}(x,y)$$

However, in this new formulation, we have found that we can obtain even better results by using

$$m_k(x,y) = \sqrt{b_k(x,y) b_{k+N_\theta}(x,y)}$$

which corresponds to the formula of the geometric mean. Figure 2(c) shows en example of the use of this formula. As we can see, with the geometric mean the extraction of directional information is more precise than with the summation formula.

The feature maps $b_k(x,y)$ capture the metric properties of the cursive word because their value indicates the position of a given pixel with respect to the line segment where it belongs to. In fact, we can also utilize them to extract other complex features such as cavities and the average stroke width. For instance, the detection of East cavities can be done by using

$$c_E(x,y) = (1 - g(b_{N_\theta}(x,y))) \prod_{\substack{k=0 \\ k \neq N_\theta}}^{2N_\theta - 1} g(b_k(x,y))$$

The average stroke width can be estimated by finding the highest peak of the global histogram of intensities of all feature maps $b_k' = (1 - i(x,y)) b_k(x,y)$. The count can be restricted to only those pixels lying in the neighborhood of the stroke by eliminating those for which $b_k(x,y) < 1/\lambda$.

3.4. Neural Network Formulation

One of the aims of our work is to find connectionist architectures for modeling the human reading process, including models of how information is extracted from the visual scene. In this regard, we have found that our feature extraction method has the form of a feed-forward neural network. Figure 3 shows an example of the architecture of our feature extraction network for the case of vertical strokes. The entire three layer neural network extracts at each pixel $N_\theta + 1$ features: N_θ directional features plus one closed-loop feature. In the input layer there are only pixel values. In the second layer, there are $2N_\theta$ summation neurons. Thus, each neuron sums the intensities of pixels aligned at a given direction. In the

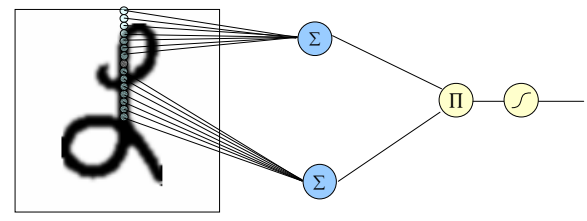

Figure 3. Equivalent neural network for a vertical stroke detector.

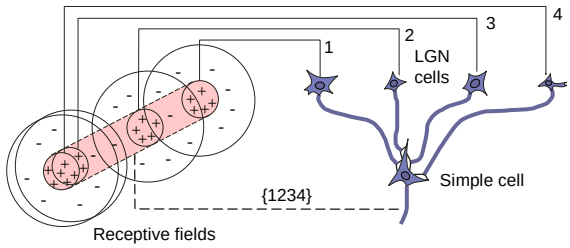

Figure 4. The feedforward model of orientation selectivity [5].

third layer, there are $N_\theta + 1$ product neurons. Since the same network is used for every pixel position (replicated in space), it can be considered as a feature extraction Sigma-Pi TDNN architecture.

We have noticed that our neural network presents an astonishing resemblance with the well known feed-forward model of orientation selectivity (see Figure 4) proposed by Hubel and Wiesel [5]. As we can see, simple cells in this model compute the sum of intensities of all pixels aligned at a given orientation and in a given neighborhood. This resemblance may not be surprising if we consider that we took motivation from the features presumably extracted by the human visual system. Somehow, the architecture of our system might be seen as a simple attempt of how visual information might be processed in the human visual system.

4. Results

We have implemented the system for performing the task of the recognition of cursive words. We have used a database that contains 3140 cursive word images of English check amounts. Each word image has been normalized to a fixed height image by using an estimation of the position of the baseline. The method used for detecting the baseline is based on the histogram of horizontal projections. After smoothing the histogram by means of an averaging filter, the optimal position of a rectangular window is determined by fitting the histogram to a rectangular function.

After preprocessing each image, we have extracted 7 feature maps: 6 directional feature maps plus one closed-loop feature map. Then, we have sub-sampled each feature map by using an anti-aliasing technique and 12 horizontal and 3 vertical zones.

Each database has been partitioned into two sets: one tenth for testing and the rest for training.

After initialization, the neural network was trained for several epochs using the method of gradient descent. We have experimented with several values of the parameters. In order to determine the needed complexity of the neural network, we have experimented using several numbers of hidden units as well as different architectures. After several essays, we have obtained better results when directional features were connected to one group of hidden units and when closed-loop features were connected only to a distinct group of neurons.

For all the tests, $N_\theta = 6$, window size is 96×48, the number of classes is 32 and the number of zones is $12 \times 3 = 36$. Thus, the total number of features is 252.

Table 1 shows the performance of the system with respect to other systems. The top result for our new method is shown in the second row. In the first row, we show the performance of the system by using the previous method. In the third row, we show the performance of a system based on global features and the first letter [4]. In the forth row, we show the performance of a system based on ascenders, descenders, loops and the interactive activation model [3]. It is clear that our new method has outperformed all the other systems.

Table 2 shows the performance of our system for different network sizes. The top result has been achieved by using 96 hidden neurons for processing directional features and 32 hidden neurons for processing closed-loop features.

System	Performance
HT features + closed loops + avg. transitions	79.18
Our new system	**85.63**
Global and 1st Letter [4]	73.50
PERCEPTO [3]	73.60

Table 1. Summary of results.

Architecture	Performance	MSE
72, 32	83.87	0.2160
80, 16	84.75	0.2203
80, 32	84.45	0.2181
88, 32	85.04	**0.2095**
96, 32	**85.63**	0.2142
104, 32	84.75	0.2256

Table 2. Results for different network sizes. Notation: number of hidden neurons for directional features, number of hidden neurons for closed-loop features.

5. Conclusions

We have presented and tested a new feature extraction method for the recognition of cursive words by using a neural network. Unlike other approaches, the new method operates on gray-level images. It extracts the directional information of the image and presents a good tolerance to disconnections of the strokes and noise. For instance, the closed-loop features obtained by using this method are resistant to deformation because loops are detected even when they are not closed.

We have tested and compared our system with other systems. Our system obtains the best performance, 85.63%. It is also important to mention that we have employed a reduced version of the same database used in [4].

We have formulated our feature extraction method as a feed-forward sigma-Pi neural network. We have seen that this network is connected to the well known feed-forward model of orientation selectivity [5] and constitutes a simple attempt to explain how visual information might be processed in the human visual system (at least during the pre-attentive stage).

We have implemented this system by using the neural network simulator described in [14].

Further work would concentrate on the comparison of our neural feature extraction network with other neural networks. For example, it would be interesting to determine the optimal architecture for extracting loops and directional features. At least in this work, our network seems to be among best ones. It would also be interesting to determine the optimal architecture for the extraction of other features presumably used by the human visual system such as end-points.

Acknowledgements

This work was supported by CONACYT and PROMEP under project number 103.5/02/2338.

References

[1] D. H. Ballard. Generalizing the Hough transform to detect arbitrary shapes, *Pattern Recognition*, 13(2), pp. 111—122, 1981.

[2] F. Cheng, W. Hsu, and M. Chen. Recognition of Handwritten Chinese Characters by Modified Hough Transform Techniques, *IEEE Trans. On Patt. Anal. and Mach. Intell.*, 11(4), 1989.

[3] M. Côté, E. Lecolinet, M. Cheriet, Y. C. Suen. Automatic Reading of Cursive Scripts Using a Reading Model and Perceptual Concepts, *International Journal on Document Analysis and Recognition*, 1(1), pp. 3—17, 1998.

[4] D. Guillevic. Unconstrained Handwriting Recognition Applied to the Processing of Bank Cheques, *PhD thesis*, Computer Science Department, Concordia University, Montreal, 1995.

[5] D. H. Hubel & T. N. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex, *Journal of Physiology*, 160, pp. 106—154, 1962.

[6] A. V. Lugt. Signal Detection by Complex Spatial Filtering, *IEEE Trans. On Information Theory*, IT-10, pp. 139—145, 1964.

[7] A. D. Mandalia. A. S. Pandya, and R. Sudhakar, A Hybrid Approach to Recognize Handwritten Alphanumeric Characters, *Proc. IEEE Int. Conf. on Systems Man and Cybernetics*, 1, pp. 723—726, 1992.

[8] L. M. Murphy. Linear Feature Detection and Enhancement in Noisy Images Via the Radon Transform, *Pattern Recognition Letters*, 4, pp. 279—284, 1986.

[9] J. Ruiz-Pinales & E. Lecolinet. Cursive Handwriting Recognition Using the Hough Transform and a Neural Network, Proc. Int. Conf. on Patt. Recog., volume 2, pp. 231—234, 2000.

[10] J. Ruiz-Pinales. Reconnaissance Hors-ligne de L'écriture Cursive par L'utilisation de Modèles Perceptifs et Neuronaux, *PhD thesis*, Computer Science and Networks Department, ENST Paris, 2001.

[11] G. Srikantan, S. W. Lam, and S. N. Srihari. Gradient-Based Contour Encoding for Character Recognition, *Pattern Recognition*, 29(7), pp. 1147—1160, 1996.

[12] H. Takahashi. A neural net OCR using geometrical and zonal pattern features, *Proc. Int. Conf. on Doc. Anal. and Recog.*, pp. 821—828, 1991.

[13] O. D. Trier, A. K. Jain, and T. Taxt. Feature Extraction Methods for Character Recognition - A Survey, *Pattern Recognition*, 29(4), pp. 641—662, 1996.

[14] A. Zell, G. Mamier et al. Stuttgart Neural Network Simulator User Manual, Version 4.2, University of Stuttgart, 1997.