Mandana Ebadian Dehkordi received her B.Sc. degree in Applied Mathematics from Isfahan University of Technology in Iran in 1988. She is recently a final year student in Ph.D course in hand writing recognition at the Nottingham Trent University, England. Her current research interest includes off-line recognition and classification of unconstrained cursive script handwriting, shape description, feature extraction, data reduction, Neural Network and pattern recognition. She also works as part time researcher in Medicsight Plc Company in UK where she is developing several classification techniques to detect abnormality in medical images. As a result of her research, She has published a number of scientific papers.

Dr Sherkat received a B.Sc Honours degree in Mechanical Engineering from the University of Nottingham in 1985. He received a Ph.D., in high speed geometric processing for continuous path generation, from the Nottingham Trent University in 1989. He is now a Reader in real-time pattern recognition in the Department of Computing and Mathematics at The Nottingham Trent University. His interests include cursive handwriting and poor quality optical character recognition and image recognition algorithms. Dr. Sherkat is the leader of the Intelligent Recognition and Interactive Systems (**IRIS**) group of the Department of Computing. He is also a director of Axiomatic Technology Limited.

Tony Allen received a first class honours degree in Physics/Chemistry from the Open University in 1990. Subsequent to this, he studied part-time at the University of Nottingham where he received an MSc in Electronic Engineering in 1992 and a PhD in Optoelectronic Neural Networks in 1997.

Tony has worked as an engineer with British Telecom PLC and as a lecturer at the Peoples College, Nottingham. Currently he is a Senior Lecturer in the Department of Computing at the Nottingham Trent University where his research interests include Collaborative and Distributed Neural Network Systems.

Style classification of handwriting

Mandana Ebadian Dehkordi, Nasser Sherkat and Tony Allen

IRIS, Department of computing, The Nottingham Trent University
Burton Street, Nottingham, NG1 4BU, U.K.
E-mail: {mand1, ns, tja}@doc.ntu.ac.uk

Abstract

This paper describes an independent handwriting style classifier that has been designed to select the best recognizer for a given style of writing. For this purpose a definition of handwriting legibility has been defined and a method has been implemented that can predict this legibility. The technique consists of two phases. In the feature extraction phase, a set of 36 features is extracted from the image contour. In the classification phase: Two non-parametric classification techniques are applied to the extracted features in order to compare their effectiveness in classifying words into legible, illegible and middle classes. In the first method a Multiple-linear Discriminant Analysis (MDA) is used to transform the space of extracted features (36 dimensions) into an optimal discriminant space for a nearest mean based classifier. In the second method, a Probabilistic Neural Network (PNN) based on the Bayes strategy and non-parametric estimation of probability density function is used. The experimental results show that the PNN method gives superior classification results when compared to the MDA method. For the legible, illegible and middle (between legible and illegible) handwriting provides 86.5% (legible/illegible), 65.5% (legible/middle) and 90.5% (middle/illegible) correct classification for two classes. For the three-classes legibility classification the rate of correct classification is 67.33% using PNN classifier.

Keywords: Style definition, Style classification, Handwriting legibility, Discriminant analysis, Probabilistic Neural Network.

1. Introduction

Nowadays very high recognition rates are possible for Optical Character Recognition (OCR); particularly for text printed in clear, easy to read fonts [20,26,37]. However, even here there are problems. The difficulties of style characterisation are even worse when handwriting is to be handled [18,30,33]. The problem of handwriting recognition is far from being solved due to the vast variability in human handwriting both between different writer (inter-writers) and within the same writer (intra-writer) [4,22,25].

Previous research has shown that writing style can vary significantly with geographical location, cultural background, age, sex, etc. [5,29]. Indeed people often completely redefine their style of writing as they age. The characteristic of cursive handwriting such as height of ascenders or descenders, word length, letter concavities etc. make the different style of writing. In cursive handwriting, letters can be connected in a variety of ways and the letter standards can differ greatly; sometimes to the point where they can be totally illegible. Cursive handwriting variability is not only due to writer's style but also to geometric factors determined by the writing conditions. There is little or no control in most off-line scenarios on the type and instrument used. The artefacts of the complex interactions between, instrument and subsequent operations such as scanning and binerization present additional challenge to algorithms for off-line handwriting recognition. Low-quality images, which lead to poor image quality such as broken lines, due to the machine printers or fax machines, pose serious challenge to current pattern recognition techniques.

Experience with analysis of word recognition systems shows that it is unlikely that a system based on a single pattern recognition approach would be capable of handling the large variability in human handwriting. The 'correction' of this variability, prior to recognition, can be helpful in reducing the variability and can lead to an important improvement in recognition performance. Hence, in current handwriting recognition systems, a pre-processing stage is normally included. The aim is to remove unwanted variation and present, to the recogniser, characters that are close

as possible to the model templates. The main functions of such pre-processing steps are usually the correction of slant [7], the deskewing of hand-written words [3], normalisation [27] etc. The use of these pre-processing steps has been shown to improve the image quality and correct the character string recognition. However, as part of this process, some of the original information may be lost.

Many attempts have been reported to deal directly with poorly written text [15,17,38]. Unfortunately, these improvements tend to result in a decrease in the system's ability to recognise clearly written characters. Also in segmentation-free recognition systems using holistic (high-level) features will perform better with lower-case or mixed case lexicons than with an uppercase word lexicon due to the fact that upper case words have less features (no ascender or descender) than lower and mixed case words. Currently, ambiguity of handwriting is considered by taking the context into consideration by using natural language processing to select words from the recognition list to improve recognition performance. For instance, post processing ways of helping cursive script recognition aim to overcome style variation. These approaches do give limited success for improving the recognition performance but do eventually fail when the handwriting becomes highly illegible as far as the recogniser is concerned. Consequently, the recognition algorithm must deal with a variety of author-specific idiosyncrasies.

Coates, Baird and Fateman [1] have shown that there are a variety of images, which although legible to human readers, are illegible to several of the best, present day, OCR systems. It has therefore been hypothesised that one way of helping cursive script recognition systems would be to detect writing style prior to the recognition stage in order to choose the best recogniser for the given writing style. In this work the concept of style classification is introduced and the various aspects of its definition in quantitative terms are discussed. To provide a starting point, style has been defined in terms of recogniser specific legibility. In this way the best recogniser could be selected for a given style of writing using a prediction of legibility based on a given recogniser's previous performance. This research therefore focuses on the problem of classifying word images

as legible, illegible or middling prior to the recognition stage. An independent handwriting style classifier has been designed that, in principle, can be used to select the best recognizer for a given style of writing. For this purpose a definition of recogniser specific handwriting legibility has been defined and a method has been implemented that can predict this legibility. Multiple Discriminant Analysis (MDA) and Probabilistic Neural Network (PNN), based on the Bayes strategy and non-parametric estimation of probability density function, techniques are proposed. Both methods are applied to the task of classifying words into legible, illegible or middling prior to the recognition stage. A comparison between the two classification techniques is thus given.

2. Definition of legibility

Up until now handwriting legibility has been defined purely in human terms. However, since the ability of a machine-based recogniser differs significantly from that of a human being [1], any definition of legibility should be based on the recognition system. Of course, similar to that of a human being, the definition of legibility is a debatable issue. However, at the time of writing no reference to a machine based definition of legibility has been found in the literature, which is probably not surprising considering the novelty of this concept.

Our definition of handwritten legibility has therefore been based on our existing recogniser's performance [34]. This recogniser is a holistic word level recogniser (HVBC) that uses three features namely, Holes, Vertical bars and Cups. This definition of legibility can be extended to any available recogniser. Fig. 1 shows that almost all correct words are located within the top10 positions. Thus legible words could be further defined as those that are likely to be placed the in the top 10 of the correct word list with a score of 75 or greater. Illegible words could be those that would produce a list containing the correct word anywhere in the word list with a score of less than 45. Middle words (those between legible and illegible) are then defined as those that would produce a list containing the correct word with a score of 45 to 75.

These thresholds have been arrived at experimentally and merely provide a starting point. They can be changed depending on the application in which they are to be used. The following

experiments serve to assess the validity of this approach by conducting a binary followed by triple style classification.

3. Feature extraction

During the design process of this classification system, thirty-six features from the contour of a reasonably large number of hand-written word images were extracted. The data set is provided by eighteen different writers (150 words each) [34]. The reasoning behind the choice for data sets and features is provided below and in further detail in [6,10,19,24]. Some sample images are available from http://www.doc.ntu.ac.uk/ns/c_sample.html.

3.1 Contour-based features

As a starting point, based on human perception of style, it was assumed that the word contour, as defined by tracing around the outside of the whole word, could contain information about the relationship of the underlying characters used in constructing the word [6]. We extend this to the hypothesis that the 'synergy' within the word resulting from the way in which the neighbouring characters follow/influence each other is encapsulated in the word shape. A number of features were therefore introduced which are based on the contour of the handwritten word images.

A handwritten word can be described as a sequence of disjointed loop contours

$$WI = \{C_i \mid C_i \cap C_j = \phi, i \neq j, j = 1,2,\ldots,N\}. \quad (1)$$

Each loop contour C_i is a sequence of consecutive points on the x-y plane:

$$C_i = \{p_j \mid j = 1,2,\ldots,M_i, p_1 = p_{M_i}\}, \quad (2)$$

where p_1 and p_{M_i} are the end points of i^{th} loop contour.

The contour-based features used in our system are mainly based on:

(a) The chain coding from the eight primitive directions given by Freeman encoding [12].

Fig. 2 refers to the eight primitive directions and represents the writing direction from a start point to an end point by following the upper contour of the word. Each loop contour C_i can be represented by a chain code sequence

$$D_i = \{d_j \mid j = 1,2,...,M_i - 1\}, \quad (3)$$

and

$$D = \bigcup_{i=1}^{N} D_i \quad (4)$$

(b) Consecutive exterior angles and contour angles formed by pairs of vectors along the word images. Fig. 3 shows the exterior angle a_l at point p_l formed by a pair of vectors d_l and d_{l-1}, and are located on the left-hand side of the vectors. The value of a_l can be obtained easily using lookup Table 1. The sequences of exterior angles in a loop contour, C_i, is calculated as:

$$A_i = \{a_j \mid j = 2,3,...,M_i - 1\} \quad (5)$$

$(d_{l-1} - d_l) \bmod 8$	0	1	2	3	5	6	7
a_l	180	135	90	45	315	270	225

Table 1: a_l as a function of $(d_{l-1} - d_l)$.

(c) Dominant points.

Dominant points refer to points of the following types:

(1) End points of the segmented regions of each individual loop contour.

(2) Points corresponding to local extreme of curvatures of each individual loop contour.

(3) Midpoints between two consecutive points of type (1) or (2).

Using the above concepts, the following subsections define the selected features in detail.

3.2 Global Features

Madhvanath in 2001 [25] shows how word shape contains sufficient information to classify words in certain lexicons. These characteristics of handwriting are different from one writer to another. A number of features based on the overall shape of a given word have been nominated. Assuming N is number of loop contours.

(1) *An estimate of number of sharp angles in the whole word:* Ratio of number of original sharp angles to the total number of angles (**ROSP**):

$$\textbf{ROSP} = \frac{\sum_{i=1}^{N} card(A_i^{90})}{card(P)} \tag{6}$$

Where

$$\mathrm{A}_i^{\theta} = \left\{ a_j \in A_i \mid a_j \le \theta \, , \, j = 2 \ldots M_{i-1} \right\} \tag{7}$$

$$P = \bigcup_{i=1}^{N} C_i \tag{8}$$

$$card(P) = \sum_{i=1}^{N} card(C_i) = \sum_{i=1}^{N} M_i \tag{9}$$

and *card* stands for the number of members in a set and sharp angles are the angle less than or equal to 90 degree.

(2) *Average of the component length (disjoint loop contours) or averaged component length* (**ACOL**):

$$\textbf{ACOL=} \frac{card(P)}{N} \tag{10}$$

(3) *Ratio of Vertical direction (2 and 6 directions given by Freeman code) to the total original chain code* (**RVO**):

$$\textbf{RVO} = \frac{card(N^{ver})}{card(P)} \tag{11}$$

Where

$$N^{ver} = \bigcup_{i=1}^{N} N_i^{ver} \tag{12}$$

$$N_i^{ver} = \left\{ \mathrm{d_j} \in D_i \mid d_j = 2 \vee d_j = 6 \right\} \tag{13}$$

and $card(N^{ver}) = \sum_{i=1}^{N} card(N_i^{ver})$ as $N_i^{ver} \cap N_j^{ver} = \phi \quad for\ i \neq j$.

(14)

(4) *Ratio of Horizontal directions (any 0 and 4 directions given by Freeman code) to the total original chain code* (**RHO**):

$$\textbf{RHO} = \frac{card(N^{hor})}{card(P)} \tag{15}$$

Where

$$N^{hor} = \bigcup_{i=1}^{N} N_i^{hor} \tag{16}$$

$$N_i^{hor} = \left\{ d_j \in D_i \mid d_j = 0 \vee d_j = 4 \right\} \tag{17}$$

$$\text{and } card(N^{hor}) = \sum_{i=1}^{N} card(N_i^{hor}) \text{ as } N_i^{hor} \cap N_j^{hor} = \phi \quad for\ i \neq j\,. \tag{18}$$

(5) *Ratio of diagonal directions (any 1,3,5 and 7 directions given by Freeman code) to the total original chain code* **(RDO):**

$$\textbf{RDO} = \frac{card(N^{dia})}{card(P)} \tag{19}$$

Where

$$N^{dia} = \bigcup_{i=1}^{N} N_i^{dia} \tag{20}$$

$$N_i^{dia} = \left\{ d_j \in D_i \mid d_j = 1 \vee d_j = 3 \vee d_j = 5 \vee d_j = 7 \right\} \tag{21}$$

$$\text{and } card(N^{dia}) = \sum_{i=1}^{N} card(N_i^{dia}) \text{ as } N_i^{dia} \cap N_j^{dia} = \phi \quad for\ i \neq j\,. \tag{22}$$

3.3 Region- based Features

The region-based features were proposed in order to measure the plain, concave and convex regions and this variability of writing could be used for style or legibility of handwriting [23].

The region-based features used are the dominant points in the contours and direction primitives between dominant points. Prior to the process of finding dominant points, a Gaussian Average Filter is used to reduce the influence of digitisation noise. The filtered version of A_i is denoted as:

$$\overline{A_i} = \left\{ \overline{a}_i \mid i = 2,3,...,M_i - 1 \right\}. \tag{23}$$

After performing Gaussian Average Filter on A_i, each contour C_i can be partitioned into a sequence of convex, concave and plain regions.

$$C_i = \bigcup_{j=1}^{T_i} R_{ij}^k \tag{24}$$

Where

T_i is the number of disjointed regions of C_i

$R_{ij}^k, k \in \{ 1,2,3 \}$, are series of consecutive points on contours C_i, in such a way that :

$$R_{ij}^1 = \{ p_l \in C_i \mid p_l \text{ are consecutive points}, \bar{a}_l = 180 \} \quad \text{(Plain region)} \tag{25}$$

$$R_{ij}^2 = \{ p_l \in C_i \mid p_l \text{ are consecutive points}, \bar{a}_l < 180 \} \quad \text{(Concave region)} \tag{26}$$

$$R_{ij}^3 = \{ p_l \in C_i \mid p_l \text{ are consecutive points}, \bar{a}_l > 180 \} \quad \text{(Convex regions)} \tag{27}$$

Figs. 4,5,6 and 7 show an example of a typical word with its concave, convex and plain regions consecutivly.

The contour angle v_l at p_l is defined within a support region and its value estimated by averaging angles a_{lk}, where $k = 1,2,3,....,K$ and a_{lk} is formed by the pair of vectors d_{l-k} and d_{l+k-1}. Denoting the sequence of contour angles in the region as;

V = $v_2 v_3 .... v_{M_i - 1}$, one can easily obtain the maximum within a convex region and the minimum in a concave region. All such maxima and minima constitute the local extremes of the curvature (corner points) along a word. More details of the above technique can be found in [23]. Fig. 8 shows the corner points, which are detected on words after using Average Gaussian Filtering, with 2 iterations while $K = 3$ is considered. It should be noted that the experiments show that as the number of iteration is increased the filtering process will remove some of the dominant points as well as the noise. On the other hand if the number of iterations is not enough the system will detect some of the noise as dominant points.

Denoting $C_i^{cr} = \{\ p_j^{cr} \in C_i \mid j = 1,2,..,S_i\ \}$ as the dominant or critical points of the i^{th} contour and $D_i^{cr} = \{\ d_j^{cr} \mid j = 1,2,...,S_i - 1\ \}$ as the direction primitives between dominant points, the region-based features are defined as follows:

(1) *Average Region Length* (**AREL**):

$$\mathbf{AREL} = \frac{card(P)}{\sum_{i=1}^{N} \sum_{\substack{j=1 \\ k\in\{1,2,3\}}}^{T_i} card(R_{ij}^k)} \tag{28}$$

(2) *Average Plain Region Length* (**APRL**):

$$\mathbf{APRL} = \frac{card(P)}{\sum_{i=1}^{N} \sum_{j=1}^{T_i} card(R_{ij}^1)} \tag{29}$$

(3) *Average Concave Region Length* (**ACAL**):

$$\mathbf{ACAL} = \frac{card(P)}{\sum_{i=1}^{N} \sum_{j=1}^{T_i} card(R_{ij}^2)} \tag{30}$$

(4) *Average Convex Region Length* (**ACVL**):

$$\mathbf{ACVL} = \frac{card(P)}{\sum_{i=1}^{N} \sum_{j=1}^{T_i} card(R_{ij}^3)} \tag{31}$$

(5) *Ratio of Sharp Angle of critical points to the total number of critical points* (**RSCR**) is:

$$\mathbf{RSCR} = \frac{\sum_{i=1}^{N} card(V_i^{cr,90})}{\sum_{i=1}^{N} card(C_i^{cr})} \tag{32}$$

Where

$$V_i^{cr,\theta} = \{v_j \in V_i \mid v_j < \theta, P_j \in C_i^{cr},\ j = 2,3,....,M_i - 1\} \tag{33}$$

(6) *Ratio of filtered Sharp Angle to the total number of Points* (**RFSP**):

$$\mathbf{RFSP} = \frac{\sum_{i=1}^{N} card(\overline{A}_i^{90})}{card(P)} \tag{34}$$

Where

$$\overline{\mathrm{A}}_{\mathrm{i}}^{\theta} = \left\{ \overline{a}_j \in \overline{A}_i \mid \overline{a}_j < \theta \, , \, j = 2,3,..., M_{i-1} \right\} . \tag{35}$$

(7) *Ratio of critical vertical code to the total critical chain code* (**RVF**):

$$\mathbf{RVF} = \frac{card(\overline{N}^{ver})}{\sum_{i}^{N} card(C_i^{cr})} \tag{36}$$

Where

$$\overline{N}^{ver} = \bigcup_{i=1}^{N} \overline{N}_i^{ver} \tag{37}$$

$$\overline{N}_i^{ver} = \left\{ \mathrm{d}_{\mathrm{j}}^{\mathrm{cr}} \in D_i^{cr} \mid d_j^{cr} = 2 \vee d_j^{cr} = 6 \right\} \tag{38}$$

$$\text{and } card(\overline{N}^{ver}) = \sum_{i=1}^{N} card(\overline{N}_i^{ver}) \tag{39}$$

$$\text{as } N_i^{ver} \cap N_j^{ver} = \phi \quad for\ i \neq j . \tag{40}$$

(8) *Ratio of critical horizontal code to the total critical chain code* (**RHF**):

$$\mathbf{RHF} = \frac{card(\overline{N}^{hor})}{\sum_{i}^{N} card(C_i^{cr})} \tag{41}$$

Where

$$\overline{N}^{hor} = \bigcup_{i=1}^{N} \overline{N}_i^{hor} \tag{42}$$

$$\overline{N}_i^{hor} = \left\{ \mathrm{d}_{\mathrm{j}}^{\mathrm{cr}} \in D_i^{cr} \mid d_j^{cr} = 0 \vee d_j^{cr} = 4 \right\} \tag{43}$$

$$\text{and } card(\overline{N}^{hor}) = \sum_{i=1}^{N} card(\overline{N}_i^{hor}) \tag{44}$$

$$\text{as } N_i^{hor} \cap N_j^{hor} = \phi \quad for\ i \neq j . \tag{45}$$

(9) *Ratio of critical diagonal to the total critical chain code* (**RDF**):

$$\mathbf{RDF} = \frac{card(\overline{N}^{dia})}{\sum_{i}^{N} card(C_i^{cr})} \tag{46}$$

Where

$$\overline{N}^{dia} = \bigcup_{i=1}^{N} \overline{N}_i^{dia} \tag{47}$$

$$\overline{N}_i^{dia} = \left\{ \mathrm{d}_{\mathrm{j}}^{cr} \in D_i^{cr} \mid d_j^{cr} = 1 \vee d_j^{cr} = 3 \vee d_j^{cr} = 5 \vee d_j^{cr} = 7 \right\} \tag{48}$$

$$\text{and } card(\overline{N}^{dia}) = \sum_{i=1}^{N} card(\overline{N}_i^{dia}) \tag{49}$$

$$\text{as } N_i^{dia} \cap N_j^{dia} = \phi \quad for\ i \neq j\,. \tag{50}$$

3.4 Windows-based Features

Fig. 10 shows how handwriting from one person to another person could be different in each window. As this figure shows the number of pixels and the value of slope in each window should be different. Therefore the following features were introduced to investigate this style characteristic.

Four values of slope corresponding to the angle of a direction with the horizontal are extracted from the 8 directions given by the Freeman code. The 4 values correspond to angles of 0, 45, 90 and 135 degrees respectively to the horizontal (Fig. 11).

For a given window i and a given slope k, the $po\,\mathrm{int}\,szone(i \mid k)$ is computed as follows:

$$po\,\mathrm{int}\,szone(i \mid k) = \frac{\left(\dfrac{card(i \mid k)}{\sum_k card(i \mid k)} \right)}{\max_{i,k} \left(\dfrac{card(i \mid k)}{\sum_k card(i \mid k)} \right)} \tag{51}$$

Where

$card(i \mid k)$ is the number of contour points with a given slope k

The total number of local features extracted for a given window position is a made up of 3 slope features for each of the 3 zones. These are defined as follows:

(1) Ratio of vertical directions in lower window (**RVLZ**):

$$\mathbf{RVLZ} = point szone(0 \mid 2) \tag{52}$$

(2) Ratio of horizontal directions in lower window (**RHLZ**):

$$\mathbf{RHLZ} = point szone(0 \mid 0) \tag{53}$$

(3) Ratio of diagonal directions in lower window (**RDLZ**):

$$\mathbf{RDLZ} = point szone(0 \mid 1) + point szone(0 \mid 3) \tag{54}$$

(4) Ratio of vertical directions in middle window (**RVZM**):

$$\mathbf{RVZM} = point szone(1 \mid 2) \tag{55}$$

(5) Ratio of horizontal directions in middle window (**RHZM**):

$$\mathbf{RHZM} = point szone(1 \mid 0) + point szone(1,4) \tag{56}$$

(6) Ratio of diagonal directions in middle window (**RDZM**):

$$\mathbf{RDZM} = point szone(1 \mid 1) + point szone(1 \mid 3) \tag{57}$$

(7) Ratio of vertical directions in upper window (**RVZU**):

$$\mathbf{RVZU} = point szone(2 \mid 2) \tag{58}$$

(8) Ratio of horizontal directions in upper window (**RHZU**):

$$\mathbf{RHZU} = point szone(2 \mid 0) \tag{59}$$

(9) Ratio of diagonal directions in upper window (**RDZU**):

$$\mathbf{RDZU} = point szone(2 \mid 1) + point szone(2 \mid 3) \tag{60}$$

In addition to the above features the following feature is also defined:

*(10) Ratio of number of points in middle area to total number of points (**RPCE**):*

$$\mathbf{RPCE} = \frac{cardMid(P)}{card(P)} \tag{61}$$

Where

$cardMid(P)$ is the number of points in the middle zone.

3.5 Feature-Based Moments

In addition to the slope features described above, an additional feature, **NOM1**, based on the first moment is also extracted. The moment features capture the global information of word images, which could help for legibility classification of handwriting [24].

$$M_1 = (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2 \tag{62}$$

Where the co-ordinates of a contour pixel is given by the 2D binary image of the cursive word and the central moment is given by:

$$\mu_{pq} = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^p (y_i - \bar{y})^q \tag{63}$$

Where

$p_i = (x_i, y_j) \in P$ and,

$$\bar{x} = \frac{1}{N}\sum x_i; \bar{y} = \frac{1}{N}\sum y_i \tag{64}$$

and N is the total number of points in the contour word image.

3.6 Zero-Crossing Feature

As Fig. 12 shows the number of intersections of a horizontal line passing through the midline of a word are different. The following features were therefore introduced to make use of this characteristic. A horizontal line is drawn through the centre of the word.

$$\text{Centre of the word} = \frac{1}{S}\left(\sum_{i=1}^{S} x_i, \sum_{i=1}^{S} y_i\right) \tag{65}$$

Where S is the total number of points in the contour word images.

The number of intersections of this line with the contoured word gives the number of zero crossing (**NCRS**) (Fig. 12).

In addition to above features group-based features and horizontal-based histogram features are used in this research, for more details of these features refer to [8,9,10].

Features, which are used in this research, are listed in table 2 for subsequent references.

1	**Average Region Length**
2	**Average Plain Region Length**
3	**Average Concave Region Length**
4	**Average Convex Region Length**
5	**Average number of sharp angle in each region**
6	**Average number of filtered sharp angle whole word**
7	**Ratio of critical vertical code to the total critical chain code**
8	**Ratio of critical horizontal code to the total critical chain code**
9	**Ratio of critical diagonal to the total critical chain code**
10	**An estimate of number of sharp angles in the whole**
11	**An estimate of the component length (disjoint contours) or averaged component** (C_i) **length**
12	**Ratio of critical vertical code to the total critical chain code**
13	**Ratio of critical horizontal code to the total critical chain code**
14	**Ratio of critical diagonal to the total critical chain code**
15	**Ratio of vertical directions in lower window**
16	**Ratio of horizontal directions in lower window**
17	**Ratio of diagonal directions in lower window**
18	**Ratio of vertical directions in middle window**
19	**Ratio of horizontal directions in middle window**
20	**Ratio of diagonal directions in middle window**
21	**Ratio of vertical directions in upper window**
22	**Ratio of horizontal directions in upper window**
23	**Ratio of diagonal directions in upper window**
24	**Ratio of number of points in middle area to total number of points**
25	**Zero crossing**
26	**First moment feature**
27	**Ratio of number of points in middle area to total number of points**
28	**Ratio of number of black pixels in the upper zone to number of black pixels in all three zone of a word.**
29	**Spread or first moment of the histograms**
30	**Average number of groups in each word**
31	**Ratios of distance between upper bounding box and upper zone to distance between lower and upper zone for the first three groups of the word**
32	**Ratios of distance between upper bounding box and upper zone to distance between lower and upper zone for the second three groups of the word**
33	**Ratios of distance between upper bounding box and upper zone to distance between lower and upper zone for the third three groups of the word**
34	**Ratios of distance between lower bounding box and lower zone to distance between lower and upper zone for the first groups of the word**
35	**Ratios of distance between lower bounding box and lower zone to distance between lower and upper zone for the second groups of the word**
36	**Ratios of distance between lower bounding box and lower zone to distance between lower and upper zone for the third groups of the word**

Table 2. 36 extracted features

4. Classification techniques

4.1 Linear discriminant transformation (MDA)

A Multiple Discriminant Analysis (MDA) is used to transform the feature space of 36 dimensions into an optimal discriminant space for a nearest mean classifier. A brief summary of the technique is given here for clarity, but for more detail see [39]. The aim of MDA is to maximise the ratio of between-class variance and within-class variance:

$$\frac{\left|\tilde{W}_b\right|}{\left|\tilde{W}_w\right|} = \frac{\left|\phi^t W_b \phi\right|}{\left|\phi^t W_w \phi\right|} \tag{66}$$

In this equation, W_b is the between-class scatter matrix, W_w is the within-class scatter matrix and ϕ is the transformation we are searching for in order to form the optimal discrminant space. We can define the following, with $\underline{f}^{i,j} = \left(f_1^{i,j},\ldots, f_p^{i,j}\right)$ being the p extracted features of word image i in j^{th} class and n_j being the number of word images in class j^{th}:

$$\overline{\underline{f}}^{\,j} = \frac{1}{n_j}\sum_{m=1}^{n_j} \underline{f}^{m,j} \text{ (Mean of features in } j^{th} \text{ class)} \tag{67}$$

$$\overline{\underline{f}} = \frac{1}{n}\sum_{j=1}^{n} n_j \overline{\underline{f}}^{\,j} \text{ (Mean of features in all classes)} \tag{68}$$

where n is a number of classes $(j = 1,2,\ldots,n)$.

$$W^j = \sum_{i=1}^{n_j}\left(\underline{f}^{i,j} - \overline{\underline{f}}^{\,j}\right)\left(\underline{f}^{i,j} - \overline{\underline{f}}^{\,j}\right)^t \text{ (covriance in } j^{th} \text{ class)} \tag{69}$$

$$W_w = \sum_{j=1}^{n} W^j \text{ (Within class covariance)} \tag{70}$$

$$W_b = \sum_{j=1}^{n_j} n_j\left(\overline{\underline{f}}^{\,j} - \overline{\underline{f}}\right)\left(\overline{\underline{f}}^{\,j} - \overline{\underline{f}}\right)^t \text{ (between class covariance)} \tag{71}$$

Both the within-class scatters W_w and the between-class scatter W_b are analogous to their respective covariance matrices.

In looking for ϕ we can define

$\underline{y} = \phi^t \underline{F}$ (Transform $\underline{F}$ by ϕ^t) (72)

$\psi^j \equiv \left\{ \underline{y}^j \mid \underline{f}^j \in j^{th}\ class, \underline{y}^j = \phi^t \underline{f}^j \right\}$

$\underline{\bar{y}}^j = \frac{1}{n_j} \sum_{\underline{y} \in \psi^j} \underline{y}$ (Mean of transformed features in j^{th} class) (73)

$\underline{\bar{y}} = \frac{1}{n} \sum_{j=1}^{n} n_j \underline{\bar{y}}^j$ (Mean of transformed features in all classes) (74)

$\tilde{W}_w = \sum_j \sum_{y \in \psi^j} (\underline{y} - \underline{\bar{y}}^j)(\underline{y} - \underline{\bar{y}}^j)^t$ (Within-class covariance of transformed features) (75)

$\tilde{W}_b = \sum_j n_j (\underline{\bar{y}}^j - \underline{\bar{y}})(\underline{\bar{y}}^j - \underline{\bar{y}})^t$ (Between class covariance transformed features) (76)

from these it follows that

$\tilde{W}_w = \phi^t W_w \phi$ (77)

$\tilde{W}_b = \phi^t W_b \phi$ (78)

Taking the determinant of a scatter matrix is equivalent to finding the product of the eigenvalues, which, in turn, corresponds to the product of the variance. As may be seen with reference to Eq. (66) by maximising this ratio, we are looking for a transform ϕ that maximizes the between-class variance with respect to the within-class variance. The solution of Eq. (66) can be shown to correspond to the generalised eigenvectors of the following equation [31,39]:

$W_b \underline{\phi}_j = \lambda_j W_w \underline{\phi}_j$ (79)

where the vectors $\underline{\phi}_j$ then form the columns of the matrix ϕ.

In addition, the individual dimensions of the discriminant space created by each eigenvector $\underline{\phi}_j$ are now ordered. The between-class variance in dimension j is proportional to the eigenvalue λ_j. Assuming a constant within-class variance, the higher the between-class variance of a dimension, the better the discriminant capacity of that dimension.

One additional step that can be taken to scale all of the within-class variances to uniform size in the discriminant space. The variance in dimension j can be computed as $\underline{\phi}_j^t W_w \underline{\phi}_j$ and each dimension can be scaled by replacing $\underline{\phi}_j$ with

$$\hat{\underline{\phi}}_j = \frac{\underline{\phi}_j}{\sqrt{\underline{\phi}_j^t W_w \underline{\phi}_j}} \tag{80}$$

giving each new dimension uniform variance.

The decision as to whether the particular word image is allocated to one class or another is then based on measuring the Euclidean distance between its transform scores (created by the MDA) and the centroids of all the classes in the discriminant space (nearest mean classifier). The nearest mean classifier is very simple and robust. Each pattern class is represented by a single prototype, which is the mean vector of all training samples in that class. Further, this classifier does not require any user specific parameters.

4.2 Non-linear Classification PNN Method

A statistical classification method based on a Bayesian [14] decision can also be used to classify the style of an unseen word. The basic idea behind the Bayesian decision rule is to calculate the probability density functions of the features of the word images in each of the classes ω_i (i = L (leible), I (illegible) and M (middle)). The probability that a particular set of features from word image $\underline{f} = (f_1, \ldots, f_{36})$ comes from class ω_i is denoted as:

$$p(\omega_i \mid \underline{f}) \tag{81}$$

where,

$$p(\omega_i \mid \underline{f}) = \frac{p(\underline{f} \mid \omega_i) p(\omega_i)}{\sum_{j=1}^{C} p(\underline{f} \mid \omega_j) p(\omega_j)} \tag{82}$$

and C is number of classes. This equation requires knowledge of the class-conditional density. This is described in the next section.

4.2.1 Parzen Method

The accuracy of the Baysian decision in Eq. (82) depends on the accuracy with which the underlying class-conditional density is estimated. A Parzen model [28] is a class of smooth and continuous Probability Density Function (PDF) estimators, which become progressively more representative of the true class-conditional density as the number of samples increases. The Parzen model uses weight functions $W(d)$ which has a maximum value at $d = 0$ and which decreases as the absolute value of d increases. A general formulation of the Parzen model is described by:

$$g(\underline{f}) = \frac{1}{n_j \sigma_1 \cdots \sigma_p} \sum_{i=1}^{n_j} W\left(\frac{(f_1 - f_1^{\,i})}{\sigma_1}, \cdots, \frac{(f_p - f_p^{\,i})}{\sigma_p} \right)$$

(83)

where $\underline{f}^{\,i} = (f^{\,i}{}_1, \ldots, f_p^{\,i})$ and p are the sample points (extracted features) and number of features in the training set, σ_k is the variance of k^{th} features $(k = 1,2, \ldots p)$ of points that surround each sample in the training set, n_j is the number of samples in class ω_j, W is the weight function and $f_k^{\,i}$ is the k^{th} feature which is extracted from i^{th} word image belonging to the ω_j class.

In general each Parzen method should have multiple σ_i values. However to simplify the model a special case can be assumed where $\sigma = \sigma_1 = \sigma_2 = \ldots = \sigma_p$ for all of the weights of function W. A more general density estimator, which assumes a Guassian kernel distribution, is used in this study, which is well behaved and easily computed. Thus Eq. (83) becomes:

$$g(\underline{f}) = \frac{1}{n_i \sigma^{\,p} \sqrt{2\pi}} \sum_{i=1}^{n_j} e^{\frac{-\left\| \underline{f} - \underline{f}^{\,i} \right\|}{2\sigma^2}} \qquad (84)$$

As we don't know in advance which features are important and which are not therefore the presence of features whose variation is meaningless has dilutive effect the useful features. We want the

variation of unimportant features to be small so that they exert minimal influence on the distance measure computed between an unknown point (test word) and each member of the training case. The solution to this problem is to use a separate σ weight for each feature. Eq. (84) then changes to:

$$g(\underline{f}) = \frac{1}{\prod_{k=1}^{p} \sqrt{2\pi}\sigma_k} \sum_{i=1}^{n_j} e^{-D(\underline{f}-\underline{f}^i)} \tag{85}$$

where

$$D(\underline{f}, \underline{f}^i) = \sum_{k=1}^{p} \left(\frac{f_k - f_k^i}{\sigma_k} \right)^2 \tag{86}$$

In this experiment both approaches were tested in order to evaluate the effectiveness of each method. In characterising the function represented by Eq. (84) the estimation of σ_i is critical [28]. A good criterion for selecting appropriate values of σ_i is the number of correctly classified cases that each value produces.

4.2.2 Optimising the σ

For each particular σ a set of Parzen density estimators based on the training data set is estimated. The number of correctly classified words produced by each value is then used to judge the efficiency of a particular value of σ . To estimate an unbiased correct classification rate for each σ , a leave–one-out method was used. In this method, all of the training data set belonging to each class except one is used to train the system and the remaining datum is used for testing. This training and testing using the leave-one-out method was repeated until every datum element in the two or three different classes had been independently tested. The leave-one-out method thus gives class bounds of the true performance of the classifier [13].

The numbers of misclassified words for each σ are then counted as an error function. A final value of σ is then chosen that minimises the error function (number of misclassifications). The minimisation technique involves two stages. First a global search over a reasonable range is used to find a rough minimum. The range can be determined iteratively such that the error rate is

minimised. Then a golden section method [31] is used to refine the estimate. Details were extensively reported by [32,36] and therefore are not reported here.

4.3 Probabilistic Neural Network

The non-parametric classifier described in the previous section can be implemented as a Probabilistic Neural Network structure. Fig. 13 shows a neural network organization for classification of input pattern $\underline{f} = (f_1, \cdots, f_p)$ (p indicates the number of features) into three classes. The input unit is simultaneously distributed to all neurons in the pattern layer.

The network is trained by setting the W_p weight vector in one of the pattern units equal to each $\underline{f} = (f_1, \cdots, f_p)$ pattern in the training set. The dot product of the input pattern vector $\underline{f}$ with a weight vector W_p is calculated, which performs a non-linear operation on $Y_p = \underline{f}.W_p$ [35]. The summation units simply sum the inputs from the pattern units that correspond to the class from which the training pattern was selected and then apply a Bayes decision rule is used to calculate the probability density functions for each class.

Compared to traditional multi-layer perceptron (MLP) networks, our kernel-based method has a simple architecture consisting of two layers of weights, in which the first layer contains the parameters of the kernel functions and the second layer forms linear combinations of the activations of the kernel functions to generate the outputs. A MLP network often has many layers of weights and a complex pattern of connectivity. All the parameters in a MLP network are usually determined at the same time as part of a single global training strategy involving supervised training. Our kernel-based method, however, is typically trained in two stages, with the kernel functions being determined first using unsupervised techniques on the input data alone and then the second layer weights subsequently being found by fast linear supervised methods.

4.4 Comparison of Appropriate Classification Methods

Most of the standard statistical classification algorithms assume some knowledge of the distribution of the random variables used to classify. Specifically, a multivariate normal

distribution is frequently assumed, and the training set is used only to estimate the mean vectors and covariance matrix of the populations. This means that large deviations from normalities usually cause a classifier to fail. Multimodal distributions cause even the most nonparametric methods to fail. An advantage of neural networks is that they can typically handle even the most complex distributions. Multiple layer feed forward networks (MLFNs) have been shown to be robust classifiers. On the other hand, there are two main problems with MLFN: 1. there is little knowledge about how they operate and 2. what behaviour is theoretically expected of them. Another major problem with MLFN is that their training speed can be very slow. The PNN, however, usually trains orders of magnitude faster than MLFNs, and classifies as well as or better than they do. Its main drawback is that it is slow to classify. However, most important of all for many applications is that the PNN method can provide mathematically sound confidence levels for its decisions. This fact alone has made the PNN a favourite for our investigations.

Another major advantage of using a PNN is the way it handles outliers; points that are very different from the majority. In fact, outliers will have no real impact on decisions regarding the more frequent cases, yet they will be properly handled if the data is valid. Existence of outliers is an important issue for other neural network models or traditional statistical techniques since they can totally devastate the outcome.

As mentioned earlier, it should be emphasised that the outputs of our classifier also have a precise interpretation as the posterior probabilities of class membership. The ability to interpret outputs in this way is of central importance in the effective application of classifiers, as it may be used for rejecting a test pattern in case of doubt. Thus it would have some performance gains over other methods like k-nearest neighbour or support vector machine. Finally, the PNN technique is strongly based on Bayes's method of classification. This means that provided the true probability density function is known, there is a Bayes optimal decision rule that will minimise the expected cost of misclassification.

5. Experimental result and analysis

Previous work [19] had indicated the need for a careful choice of sample words to allow a good representation of a much larger vocabulary without becoming hopelessly unwieldy. Kassel [18] has discussed the design aspects of such data sets and sample words used in this research were chosen based on that work in a free space (no guidelines) and no baseline correction techniques have been applied.

The style classification technique was applied on our existing data set, which consist of scanned images obtained from eighteen writers each containing 150 words at 200×100-dpi resolution. Initially the system is trained on the LegTRn (legible training words), ILegTRn (illegible training words) and MiddleTRn (middle training words) sets containing all 2456 words in training set. The classification system was then tested with: 1) the same data set: LegTRn, ILegTRn and MiddleTRn and; 2) a different data set, LegTEn (legible test words), ILegTEn (illegible test words) and MiddleTEn (middle test words). This latter set containing 518 words, Note that the n in name of the datasets (LegTRn, ILegTRn, MiddleTRn, LegTEn, ILegTEn and MiddleTEn) shows the number of features and TR and TE indicate the training and test sets respectively. Also note that the x, y and z-axes in Figs. 3 to 5 indicate the number of segmented sigma's range, sigma's range and the estimated error in each region respectively. Sigma's range and error function are shown in on the tables under each figure.

5.1 PNN classifier using a common σ for Binary Classification

Tables 3 to 5 show the two class (binary) classification results obtained when using a non-linear classification (PNN) techniques based on the selected values of common σ . The first column in these tables shows the samples that were used as the training data set whist the second column shows the samples that were used as a test set. The third column shows the correct classification results obtained when using a non-linear classification (PNN) technique with common σ using all of the 36 features. The fourth column shows the average of correct classification results when

the system was tested with seen or unseen data and the average classification result for all with common σ is given in the last row. Figs. 14 to 16 indicate the estimated error based on sigma's range then the best value of σ is chosen.

Training set	Test set	% Correct Classification result (common σ)	%Correct Average
LEGTR36, ILLEGTR36	LEGTR36	99.00%	99.50%
LEGTR36, ILLEGTR36	ILLEGTR36	100.00%	
LEGTR36, ILLEGTR36	LEGTE36	69.00%	79.50%
LEGTR36, ILLEGTR36	ILLEGTE36	90.00%	
		Overall	89.50%

Table 3: Classification results using 36 extracted features to discriminant between legible and illegible handwriting using common σ (σ =5.47436)

Training set	Test set	% Correct Classification result (common σ)	%Correct Average
LEGTR36, MiddleTR36	LEGTR36	100.00%	99.50%
LEGTR36, MiddleTR36	MiddleTR36	99.00%	
LEGTR36, MiddleTR36	LEGTE36	81.00%	65.50%
LEGTR36, MiddleTR36	MiddleTE36	50.00%	
		Overall	82.50%

Table 4: Classification results using 36 extracted features to discriminant between legible and middle handwriting using common σ (σ =7.11064).

Training set	Test set	% Correct Classification result (common σ)	%Correct Average
MiddleTR36, ILLEGTR36	MiddleTR36	99.00%	99.50%
MiddleTR36, ILLEGTR36	ILLEGTR36	100.00%	
MiddleTR36, ILLEGTR36	MiddleTE36	52.00%	76.00%
MiddleTR36, ILLEGTR36	ILLEGTE36	100.00%	
		Overall	87.75%

Table 5: Classification result using 36 extracted features to discriminant between middle and illegible handwriting using common σ (σ =0.01386).

Tables 3 to 5 show that the average classification result is 89.50% , 82.50% and 87.75% when classifying between legible/illegible, legible/Middle and illegible/Middle word images respectively using 36 extracted features with common σ. The system can also achiev 99.50%, 99.50% and 99.50% correct classification when the test set is the same as the training set and 79.50%, 65.50% and 76.00% correct classification when the test set is different to the training set.

5.2 PNN classifier using different σ_i for Binary Classification

Tables 6 to 8 show the classification results obtained when using a non-linear classification (PNN) technique with the different values of σ_i $(i = 1,2,\cdots,36)$. The first column in these tables shows the samples that were used as the training data set whilst the second column shows the correct classification results obtained when using a non-linear classification (PNN) technique with different σ_i using all 36 features.

Training set	Test set	% Correct Classification (different σ_i)	% Correct Average
LEGTR36, ILLEGTR36	**LEGTR36**	**99.00%**	**99.50%**
LEGTR36, ILLEGTR36	**ILLEGTR36**	**100.00%**	
LEGTR36, ILLEGTR36	**LEGTE36**	**90.00%**	**86.50%**
LEGTR36, ILLEGTR36	**ILLEGTE36**	**83%**	
		Overall	**93.00%**

Table 6: Classification result using 36 extracted features to discriminant between illegible and legible handwriting using different σ_i.

Training set	Test set	% Correct Classification (different σ_i)	% Correct Average
LEGTR36, MiddleTR36	**LEGTR36**	**100.00%**	**99.50%**
LEGTR36, MiddleTR36	**MiddleTR36**	**99.00%**	
LEGTR36, MiddleTR36	**LEGTE36**	**81.00%**	**65.50%**
LEGTR36, MiddleTR36	**MiddleTE36**	**50.00%**	
		Overall	**82.50%**

Table 7: Classification result using 36 extracted features to discriminant between middle and legible handwriting using different σ_i.

Training set	Test set	% Correct Classification (different σ_i)	% Correct Average
MiddleTR36, ILLEGTR36	**MiddleTR36**	**99.00%**	**99.50%**
MiddleTR36, ILLEGTR36	**ILLEGTR36**	**100.00%**	
MiddleTR36, ILLEGTR36	**MiddleTE36**	**98.00%**	**90.50%**
MiddleTR36, ILLEGTR36	**ILLEGTE36**	**83.00%**	
		Overall	**95.00%**

Table 8: Classification results using 36 extracted features to discriminant between middle and illegible handwriting using different σ_i.

Tables 6 to 8 show that the overall classification results are 93.00%, 82.50% and 95.00% correct classification when classifying legible/middle, illegible/middle and legible/illegible handwriting

word images respectively. These can be broken down into 99.50%, 99.50% and 99.50% correct classification when the test set is the same as the training set and 86.00%, 65.50% and 90.50% correct classification when the test set is different to the training set.

5.3 Multiple Linear Classification (MDA) for binary classification

Tables 9 to 11 show the experimental results obtained using all 36 extracted features to classify between legible/illegible, legible/middle and illegible/middle word images when using the multi-linear discriminant analysis technique. The first column shows the samples that were used as the training data set whilst the second column shows the samples that were used as a test set. The third column shows the correct classification result. The fourth column shows average of correct classification result when the system was tested with seen or unseen data. The last row then shows the average classification results for all data. The training samples and test samples are the same as those used in the non-linear classification experiment.

Training set	Test set	% Correct Classification MDA	% Correct Average
LEGTR36, ILLEGTR36	**LEGTR36**	**78.00%**	**70.50%**
LEGTR36, ILLEGTR36	**ILLEGTR36**	**63.00%**	
LEGTR36, ILLEGTR36	**LEGTE36**	**67.00%**	**60.50%**
LEGTR36, ILLEGTR36	**ILLEGTE36**	**54.00%**	
		Overall	**65.50%**

Table 9: Classification result using 36 features to discriminate between legible and illegible.

Training set	Test set	% Correct Classification MDA	% Correct Average
LEGTR36, MiddleTR36	**LEGTR36**	**70.00%**	**64.00%**
LEGTR36, MiddleTR36	**MiddleTR36**	**58.00%**	
LEGTR36, MiddleTR36	**LEGTE36**	**57.00%**	**63.50%**
LEGTR36, MiddleTR36	**MiddleTE36**	**70.00%**	
		Overall	**63.75%**

Table 10: Classification result using 36 features to discriminate between legible and middle.

Training set	Test set	% Correct Classification MDA	% Correct Average
MiddleTR36,ILLEGTR36	**MiddleTR36**	**66.00%**	**64.50%**
MiddleTR36,ILLEGTR36	**ILLEGTR36**	**63.00%**	
MiddleTR36,ILLEGTR36	**MiddleTR36**	**56.00%**	**57.50%**
MiddleTR36,ILLEGTR36	**ILLEGTR36**	**59.00%**	
		Overall	**61.00%**

Table 11: Classification result using 36 features to discriminate between middle and illegible.

The overall binary classification using 36 features in the MDA technique is 65.50%, 63.75%, and 61.00% for classification between legible/illegible, legible/middle and illegible/middle words. This can be broken down into 70.50%, 64.00% and 64.50% correct classification when the test set is the same as training set and 60.5%, 63.50% and 57.50% correct classification when training set is different to the test set.

5.4 Comparison Between Using the Linear and Non-linear Method for Binary Classification

Tables 12 and 13 summarise the experimental result obtained when using all 36 extracted features using the PNN technique with common σ, different σ_i and MDA technique.

Training se is the same as test set	Legible/Illegible		Illegible/Middle		Middle/Legible		Overall	
	Dif σ_i		Dif σ_i		Dif σ_i		Dif σ_i	
	Com σ	**MDA**	***Com*** σ	**MDA**	***Com*** σ	**MDA**	***Com*** σ	**MDA**
36 extracted features	**99.50%**		**99.50%**		**99.50%**		**99.5%**	
	99.50%	70.50%	*99.50%*	64.50%	*99.50%*	64.00%	*99.50%*	66.33%

Table 12: Comparison between the classification results when (i) PNN using different σ_i, (ii) PNN using common σ and (iii) MDA techniques when the training set is the same as the test.

Training se is different with the test set	Legible/Illegible Dif σ_i *Com* σ **MDA**	Illegible/Middle Dif σ_i ***Com*** σ **MDA**	Middle/Legible Dif σ_i ***Com*** σ **MDA**	Overall Dif σ_i ***Com*** σ **MDA**
36 extracted features	**86.50%** *79.50%* 60.50%	**90.5%** *76.00%* 57.50%	**65.50%** *65.50%* 63.50%	**80.83%** *73.67%* 60.50%

Table 13: Comparison between the classification results when (i) PNN using different σ_i, (ii) PNN using common σ and (iii) MDA techniques when the training set is different with the test set

The experimental results given in tables 12 and 13 show that the PNN technique achieved an improvement of 26.00%, 2.00% and 33% using different σ_i and an improvement of 19.00%, 2.00% and 18.50% using common σ when compared to the MDA technique for classification between legible/illegible, legible/middle and illegible/middle words respectively where the test set is different to the training set. In the case where training set is the same as the test set the PNN technique achieved an improvement of 29.00%, 35.50% and 35.00% using different σ_i and an improvement of 29.00%, 35.50% and 35.00% using common σ compared to the MDA technique for classification between legible/illegible, legible/middle and illegible/middle words respectively.

The results given in table 12 show that when the training set is the same as the test set there is no difference in classification rate between using different σ_i values and common σ value. However, table 13 shows that whilst using different σ_i rather than common σ has no effect on the classification between legible/middle, it does give an improvement of 7.00% and 14.50% for classification between legible/illegible, illegible/middle when the test set is different to the training set.

5.5 Triple Classification using common σ

Table 14 gives the results for the 3 class data sets. The first column shows the samples that were used as the training data set whist the second column shows the samples that were used as the test set. The third column shows the correct classification results obtained when using the non-

linear classification technique with common σ using all 36 features. The fourth, fifth, sixth and seventh columns show the misclassification results in each category and the average classification results for seen and unseen data. The last row shows the overall classification results for all with common σ . For the three-class style classification the best common σ value is 0.001. The details are shown in fig. 17.

			%Misclassification words			
Training files	**Test files**	**%Correct non-linear (PNN)**	**As legible**	**As illegible**	**As Middle**	**%Correct Average**
LEGTR36, ILLEGTR36, MiddleTR36	**LEGTR36**	**100.00%**	**-**	**0**	**0**	**99.67%**
LEGTR36, ILLEGTR36, MiddleTR36	**ILLEGTR36**	**100.00%**	**0**	**-**	**-**	
LEGTR36, ILLEGTR20, MiddleTR36	**MiddleTR36**	**99.00%**	**1.00%**	**0**	**-**	
LEGTR36, ILLEGTR36, MiddleTR36	**LEGTE36**	**72.00%**	**-**	**10.00%**	**18.00%**	**67.33%**
LEGTR36, ILLEGTR36, MiddleTR36	**ILLEGTE36**	**83.00%**	**17.00%**	**-**	**0**	
LEGTR36, ILLEGTR36, MiddleTR36	**MiddleTE36**	**47.00%**	**51.00%**	**2.00%**	**-**	
					Overall	**83.50%**

Table 14: Classification results using 36 features to discriminate between legible, illegible and middle handwriting word images using common σ (σ =0.001).

The experimental results given in table 14 show that a classifier based on the PNN using a common σ value of 0.001 can achieve an overall correct style classification of 67.33% when the test set is different to the training set. The system can also be seen to achieve a 99.67% correct classification when the test set is the same as the training set. This gives an overall correct classification of 83.50% for the three classes.

5.6 Triple Classification using different σ_i

The best values of different σ_i obtained for each legible, illegible and middle classification with an error rate of 0.21840 calculates as 0,000889, 0.000931 and 0.001260 in legible, illegible and middle class respectively. Experimental results using these different σ_i are given in following table.

			%Misclassification words			
Training files	**Test files**	**%Correct non-linear (PNN)**	**AS legible**	**As illegible**	**As Middle**	**%Correct Average**
LEGTR36, ILLEGTR36, MiddleTR36	**LEGTR36**	**100.00%**	**-**	**0**	**0**	**99.33%**
LEGTR36, ILLEGTR36, MiddleTR36	**ILLEGTR36**	**99.00%**	**2.00%**	**-**	**0**	
LEGTR36, ILLEGTR20, MiddleTR36	**MiddleTR36**	**99.00%**	**0.60%**	**0.40%**	**-**	
LEGTR36, ILLEGTR36, MiddleTR36	**LEGTE36**	**72.00%**	**-**	**10.00%**	**18.00%**	**67.33%**
LEGTR36, ILLEGTR36, MiddleTR36	**ILLEGTE36**	**83.00%**	**17.00%**	**-**	**0**	
LEGTR36, ILLEGTR36, MiddleTR36	**MiddleTE36**	**47.00%**	**51.00%**	**2.00%**	**-**	
					Overall	**83.33%**

Table 15: Classification result using 36 extracted features to discriminate between legible, illegible and middle handwriting word images using different σ_i .

Table 15 shows that the PNN classifier using different σ_i values achieves 67.33% correct classification when the test set is different to the training set and 99.33% correct classification when the test set is the same as the training set. This gives an overall 83.33% correct classification.

6. Conclusion and future work

This paper has introduced a novel handwriting legibility classification system that can be used to predict the recognition performance of a recogniser for a given handwriting style in order to

choose the best recogniser. Thirty-six features are extracted and two methods for style classification of the word images are described (MDA and PNN) and a comparison between these two methods are presented.

Experimental results show that some of the features have a more significant influence on classification results than the others. However experiments also show all the features used in this research play some role and are deemed necessary for successful classification. Indeed a significant reduction of feature vectors leads to a much less effective classification [11] .

As the size and quality of writing is important in these experiments, some of the features are not extracted correctly resulting in misscalssification. It is therefore suggested that further examination of the selected fetures should be considered. One possible candidate is fractals. Fractal features may provide useful information for discriminate between legible/illegible/middle handwriting word images. These features have been useful for classifying the regularity in handwriting as well as size of writing [2].

Experimental results using MDA and PNN techniques (using different σ_i and common σ) show that in the case of legible/illegible and illegible/middle the PNN technique using different σ_i gives the superior result as compared to using the PNN with common σ and the MDA technique. However, in the case of middle/legible classification the PNN techniques using common σ and different σ_i values give the same classification result. Therefore the PNN technique using different σ_i is the best classifier. As the PNN in classification between two classes gives superior results in comparison to the MDA, for the time being we use PNN for triple classification and no experiments were carried out for the triple classification with the MDA technique. Experimental results show that those words, which were correctly classified using the MDA technique, were equally correctly classified by using PNN. However, those words, which were misclassified or closely classified by PNN, were correctly classified using MDA.

The Parzen model, used for density estimation in the PNN system, has the same number of kernels as the number of data points. This leads to models that can be slow to evaluate for new input vectors especially when the number of training data points is very large. One way to tackle this problem is to use a clustering technique such as fuzzy clustering to reduce the number of data points prior to PNN. The centre of each cluster can be used as a centre for each kernel thus greatly increasing the classification speed.

Faced with significant style variation of handwriting it is more likely that style-specific classifiers yield higher classification accuracy than the generalised classifiers. Therefore, the next stage of our work would be to use the pre-classifier to route a given data sample to a recogniser which is deemed more suitable to the style of the sample. The work so far has concentrated on a small subset of style classification. The result of our initial experiments in applying the described techniques to determine a writer style has been encouraging.

Further investigation to determine how effectively we can identify a writer will be needed. It is a fact that intra-writer style variation is also a problem [19], which leads to significant user frustration affecting success of today's on-line applications such as PDAs. It would be interesting to see whether there is any scope in treating intra-writer style variation in a similar way.

These classification methods can also be applied for identifying the symbol types such as digit, punctuation and lower, upper letters for further work [16]. For example separation of digits and uppercase, lowercase characters or words is an important task in document layout.

This method could be very useful in the field of writer and signature identification. Using the methods presented here it may be possible to determine the characteristics of each writer using the most efficient features in each writer's handwriting.

Reference:

1. Allison L. Coates, Henry S. Baird and Richard J. Fateman, "pessimal print: Areverse turing test", ICDAR'01, pp. 1154-1158, 2001.

2. V. Bouletreau, N. Vincent, R. Sabourin and H.Emptoz,"Synthetic Parameters for Handwriting Classification", ICDAR'97, pp. 102-106, 1997.

3. R.M. Bozinovic and S. N. Srihari, “Off-line cursive script word recognition”, IEEE Trans. PAMI, vol. 11, no.1, pp. 68-83, 1989.

4. F.Camastra, A. Vinciarelli, “ Cursive character recognition by learning vector quantization”, pattern recognition letters, vol. 22, no.6-7, pp. 625-629, 2001.

5. Sung-Hyuk Cha, Sargur N.Srihari, “Apriopi algorithm for sub-category classification Analysis of Handwriting”, ICDAR’01, pp. 1022-1025, 2001.

6. C H Chien and J K Aggarwal.Model, “Construction and Shape Recognition From Occluding Contours”, IEEE Trans. PAMI, vol.11, no. 4, pp. 372-389, 1998.

7. Y. Ding, F. Kimura, Y. Miyake, M. Shridhar, “ Evaluation and improvement of slant estimation for handwriting words”, ICDAR’99, pp. 753-756, 1999.

8. M. Ebadian Dehkordi, N. Sherkat and R. J. Whitrow. “ A principal Component Approach to Classification of handwritten Words”, ICDAR’99, pp. 781-784, 1999.

9. M. Ebadian Dehkordi, N. Sherkat and R. J. Whitrow. “Classification of off-line Hand-written words into upper and lower cases”, IEE Colloquium Document Image Processing and Multimedia”, pp. 8/1-8/4, London, March 1999.

10. M. Ebadian Dehkordi, N. Sherkat and T. Allen, “Case Classification of Off-line Handwritten Words Prior To Recognition”, 4th IAPR international workshop on document analysis systems DAS’2000, pp.325-334, 2000.

11. M. Ebadian Dehkordi, N. Sherkat and T. Allen, “Prediction of handwriting legibility”, ICDAR’01, pp. 997-1000, 2001.

12. H. Freeman, ”On the encoding of arbitrary geometric configuration”, IRE transactions on electronic computers, EC-10 (2): pp. 260-268, 1961.

13. Fukunaga K, Hayes R. “Estimation of classifier performance”, IEEE Trans. PAMI; 11: pp.1087-1097, 1989.

14. R.C. Gonzalez and R. E. Wood. “Digital image processing”, Addison Wesley, 1993.

15. M. Hamanaka and Yamada, “On-line character recognition adaptively controlled by handwriting quality”, proceeding of 7th international workshop on frontiers in handwriting recognition (IWFE’2000), pp. 33-42, 2000.

16. Tin Kam Ho, George Nagy, “ Exploration of contextual constraints for character pre-classification”,ICDAR’01 , pp. 450-454, 2001.

17. J. Hu, D. Yu, H. Yan, “Construction of partitioning paths for touching handwritten characters”, Pattern recognition letters, vol. 20, no. 3, pp. 293-303, 1999.

18. S. Impedow, L. Ottaviano and S. Occhinero, “Optical character recognition: A survey”, int’ I J. pattern recognition and artificial intelligence, vol. 5, no. 2, pp. 1-24, 1991.

19. Marcin S Jedrzejewski, “Automatic characterisation of handwriting style”, Mphil thesis, Department of computing Nottingham Trent University, 1997.

20. M. Jung, Y. Shin and S. N. Srihari, “Mltifont classification using typographical attributes”, ICDAR’99, pp. 353-356, 1999.

21. Kassel R. H., “A Comparison of approaches to on-line handwritten character recognition”, Doctoral Dissertation, Department of electronical engineering and computer science, Massachusetts institute of technology, June 1995.

22. G. Leedham, “Historical perspective of handwriting recognition system”, IEE colloquium on handwriting and pen-based input, (Digest No. 1994/065). london, uk, pp. 1/1-3, march 1994.

23. Li and N. S. Hall, “ Corner detection and shape classification of on-line handprinted Kanji strokes”, Patter recognition, vol. 26, No. 9, pp. 1315-1334, 1993.

24. S. Loncaric, “ A survey of shape analysis techniques”, pattern recognition, vol. 31, no. 8, pp. 983-1001, 1998.

25. S. Madhvanath,V. Govindaraju, “The role of holistic paradigms in handwritten word recogniton”, IEEE Trans. PAMI, vol. 23, no. 2, pp. 149-165, 2001.

26. S. Mori, C. Y. Suen and K. Yamamoto, “Historical review of OCR research and development”, Proc. IEEE, vol. 80, no. 7, pp. 1029-1058, 1991.

27. G. Nicchiotti and Scagliola, “Generalised projections: a tool for cursive handwriting normalisation”, ICDAR’97, pp. 729-732, 1997.

28. Parzen, E. “ On Estimation of a Probability Density Function and Mode”, Annals of Mathematical statistics, 33: pp. 1065-1076, 1962.

29. Powalka R. K., Sherkat N., Whitrow R.J. “Recognizer characterisation for combining handwriting recognition”, ICDAR’95, vol. 1, pp 68-73, 1995.

30. Powalka R. K., Sherkat N., Whitrow R.J. “Multiple recognition combination topologies”, Handwriting and drawing research:... and Applied Issues, pp. 329-342, IOS press, 1996.

31. B. D. Ripley, “Pattern Recognition and Neural networks”, Cambridge, 1997.

32. Schioler, H., and Hartmann, U. “Mapping Neural Network Derived from the Parzen window Estimator”, Neural Networks, 5(6): pp. 903-909, 1992.

33. L. Schomaker, G. Aabbink and S. Selen, “ Writer and writing-style classification in recognition of on-line handwriting”, Proceeding of the European workshop on handwriting analysis and recognition, the institute of electrical engineers, (ISSN 0963-3308), LONDON, 1994.

34. N. Sherkat, T. J. Allen, “Whole word recognition in facsimile images”, ICDAR’99, pp. 547-550, 1999.

35. Donald F. Specht, “Probabilistic Neural Networks and the polynomial adaline as comlementary techniques for classification”, IEEE transaction on networks, vol. 1, no.1, 1990.

36. Specht, Donald F., and Shapiro, Philip D. “Generalization Accuracy of probabilistic Neural Networks comared with back-propagation networks.” Lockheed Missiles & space co., Inc. Independent research project RDD 360, I-887-I-892, 1991.

37. Sargur N. Srihari, Sung_Hyuk Cha, Hina Arora, Sangjik Lee,“ Individuality of handwriting: A validation study”, ICDAR’01, pp. 1195-1204, 2001.

38. J. Wang, H. Yan, “Mending broken handwriting with a macrostructure analysis method to improve recognition”, pattern recognition letters, vol. 20, no. 8, pp. 855- 864, 1999.

39. Andrew Webb, “Statistical pattern recognition”, Arnold, 1999.

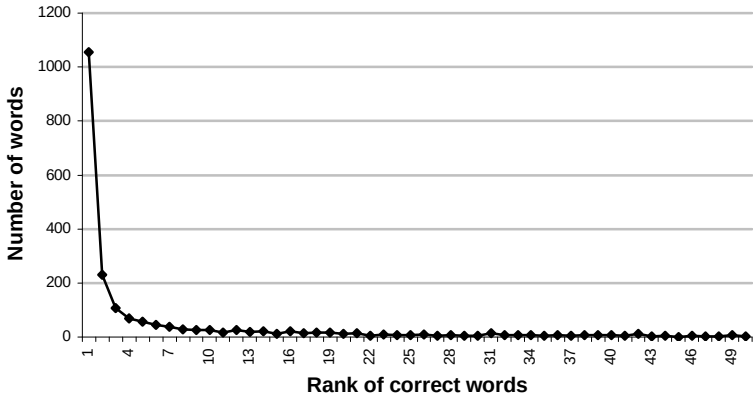


Fig 1: All correct words regardless of rank using HVBC recogniser.

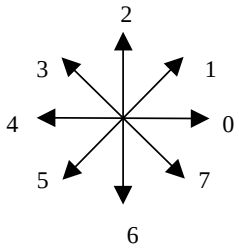


Fig 2: Eight primitive directions.

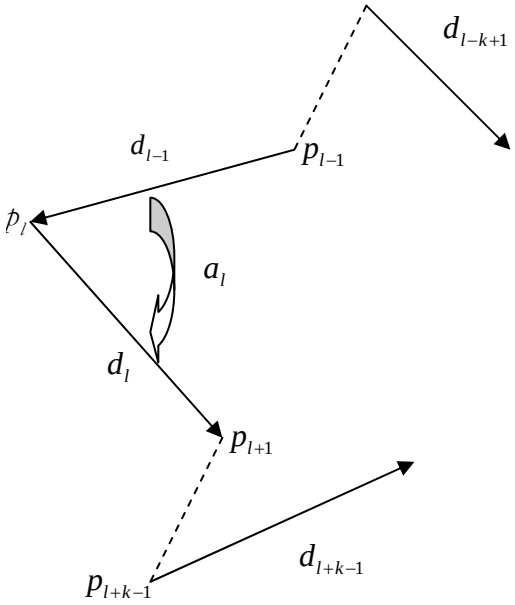


Fig 3: Angle a_l at point p_l.

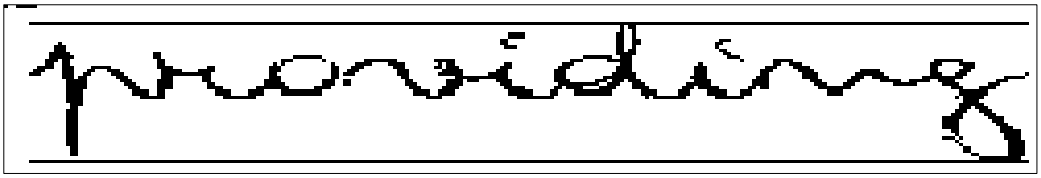


Fig 4: A typical word.

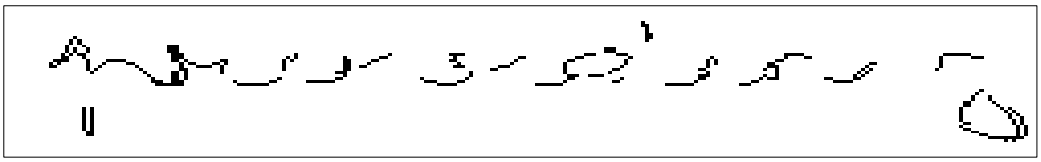


Fig 5: Concave regions.

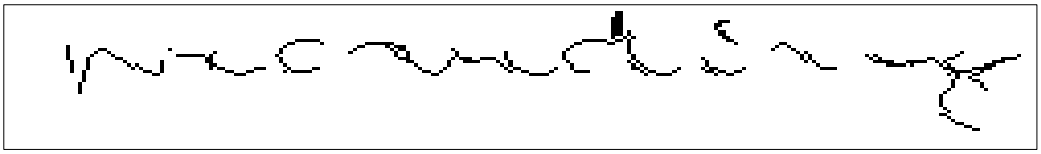


Fig 6: Convex regions.

Fig 7: Plain regions.

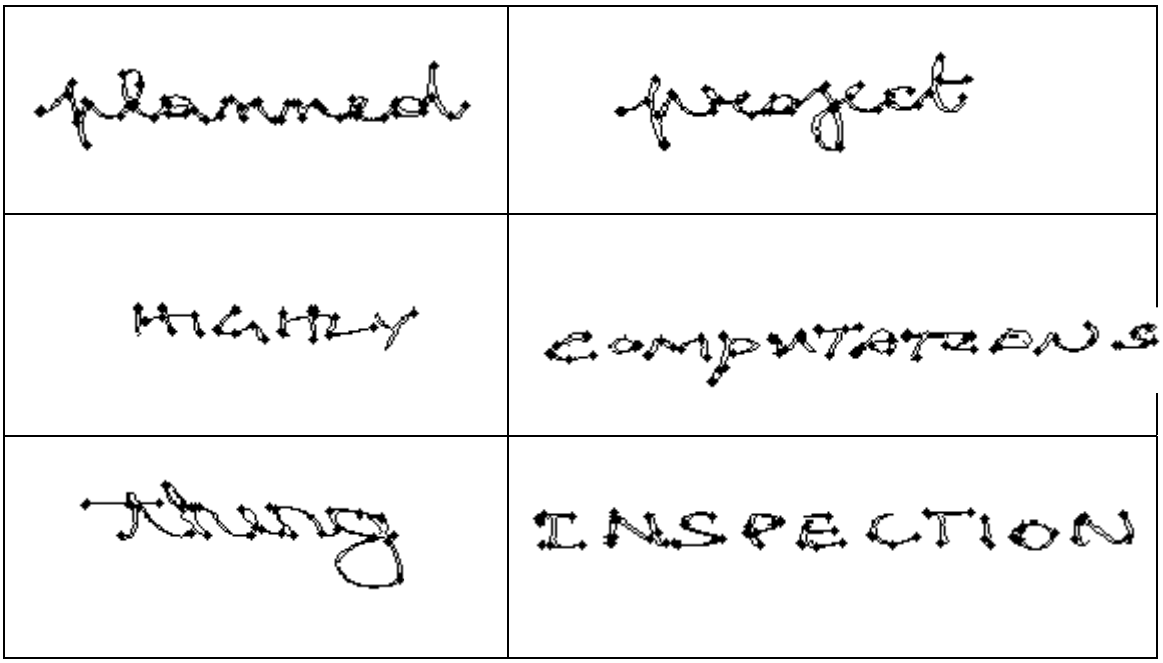


Fig 8: The detected dominant points on words.

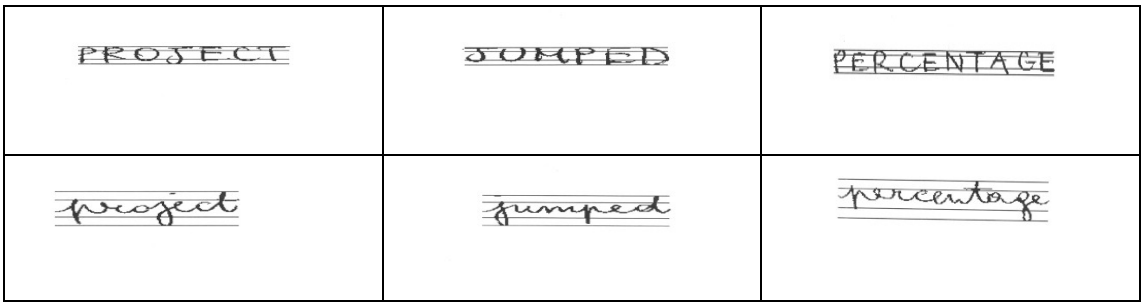


Fig 9: Three regions of interest within a window for some different word case samples.

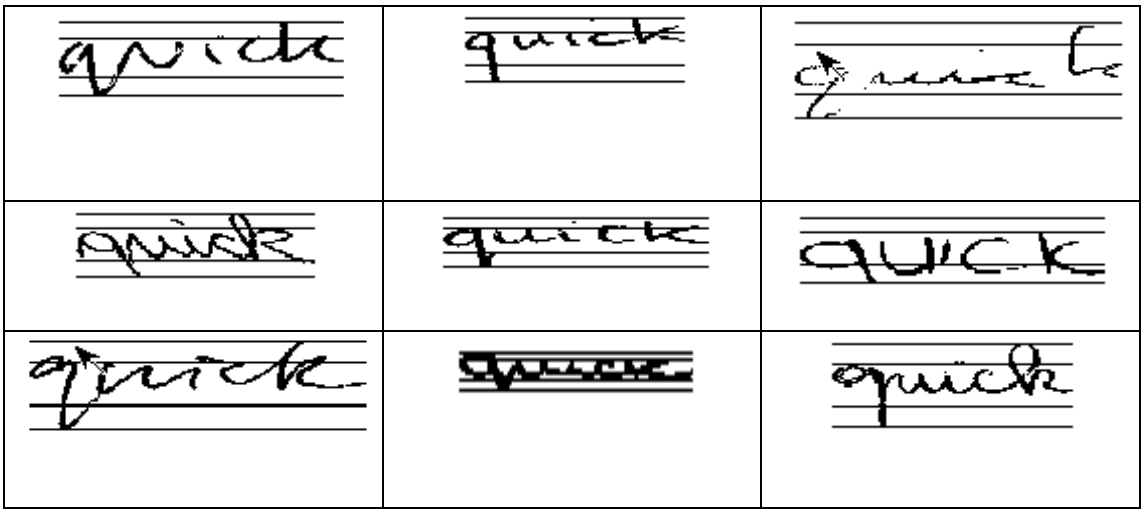


Fig 10: three region of interest within a window for some different style of handwriting (one specific word).

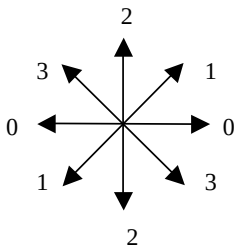


Fig 11: Representation of the four directions (slopes).

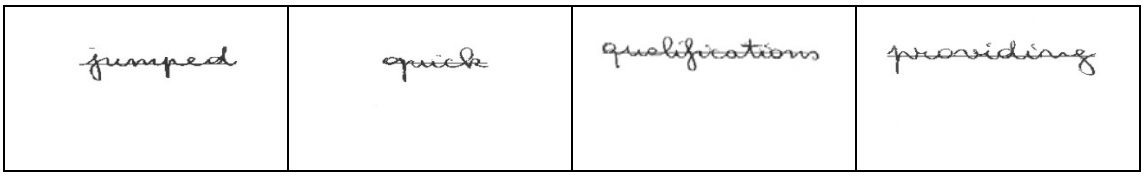


Fig 12: Horizontal lines are drawn from the centre of each word.

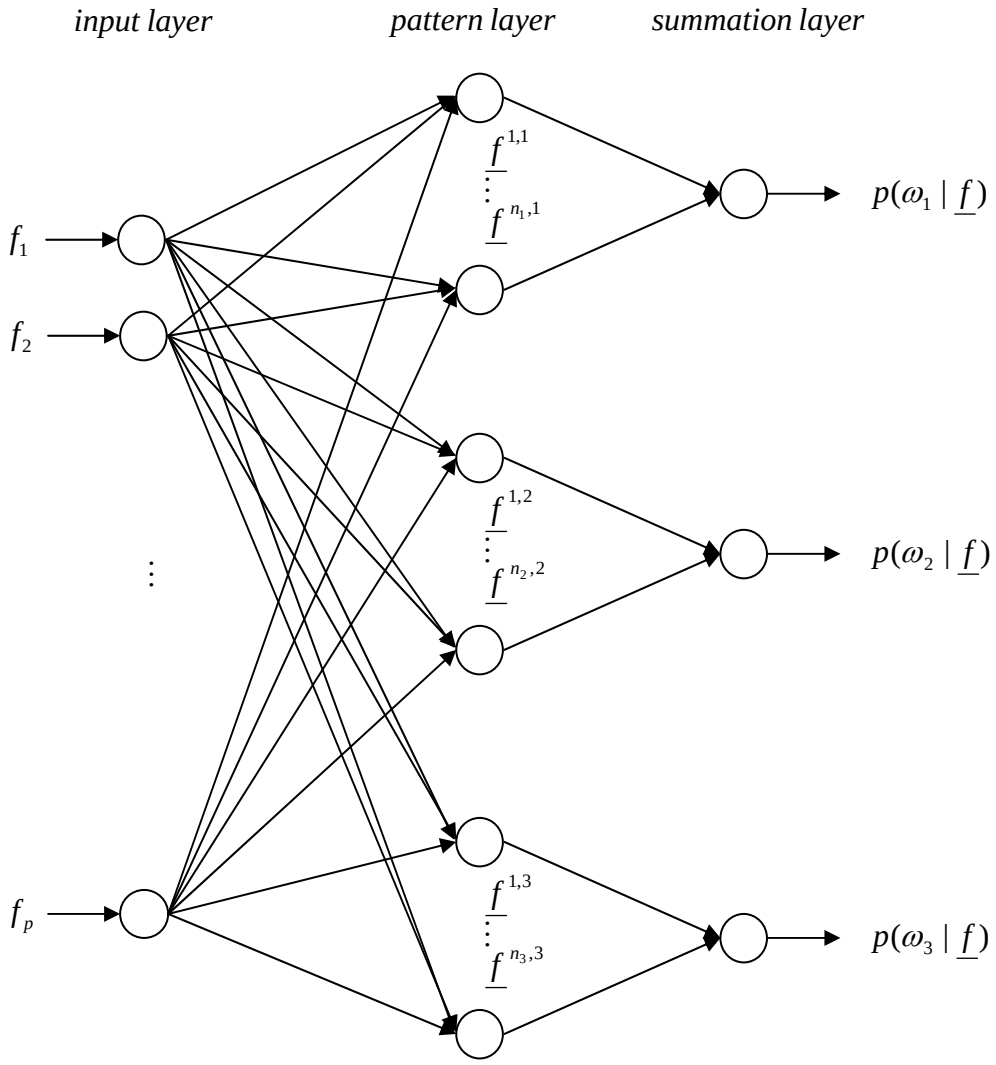
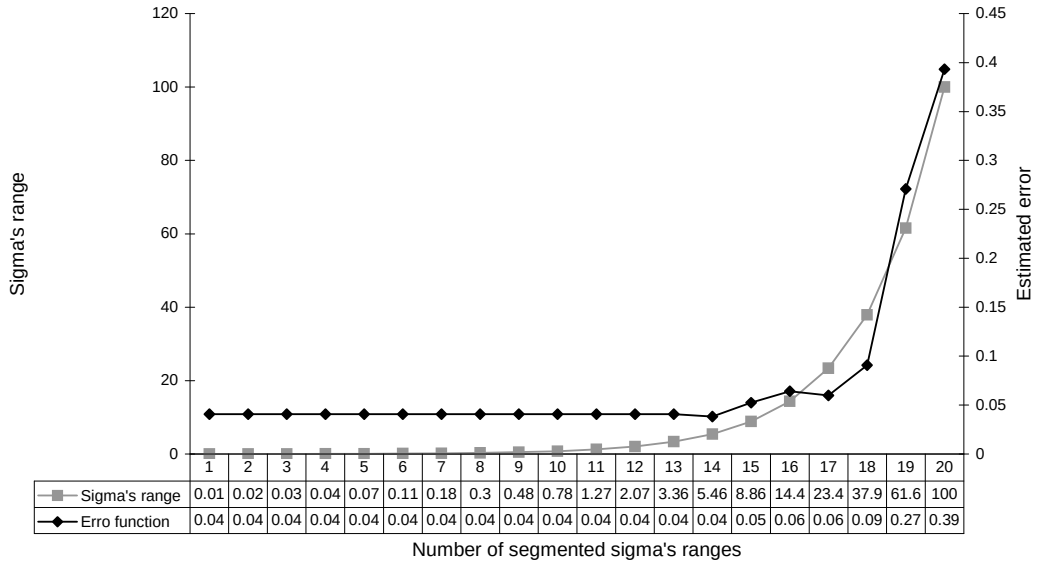


Fig 13. Organization of a probabilistic neural network for classification of patterns into categories.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Sigma's range	0.01	0.02	0.03	0.04	0.07	0.11	0.18	0.3	0.48	0.78	1.27	2.07	3.36	5.46	8.86	14.4	23.4	37.9	61.6	100
Erro function	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.06	0.06	0.09	0.27	0.39

Fig 14: Error estimation of common σ for a classification between legible and illegible handwriting using 36 extracted features (σ =5.47436).

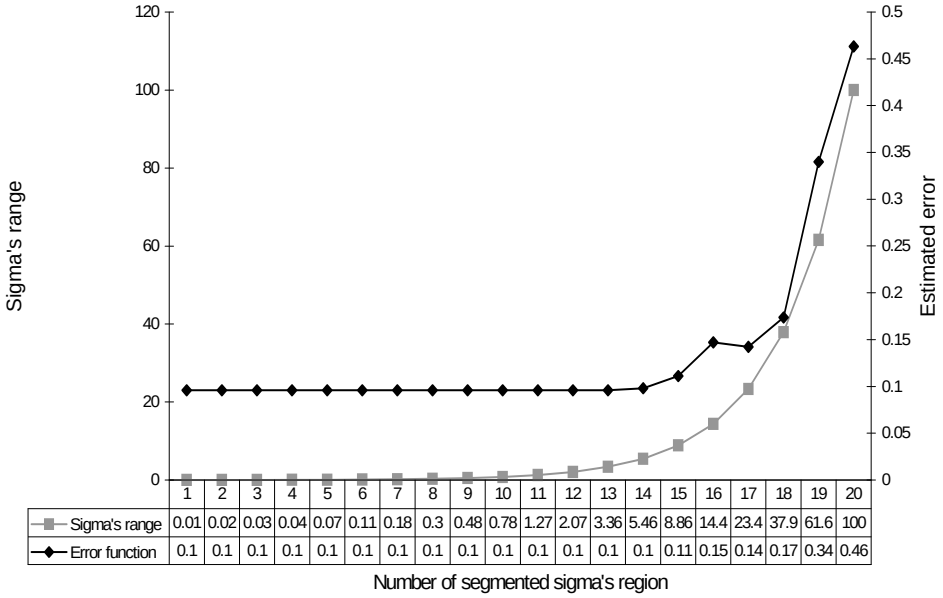
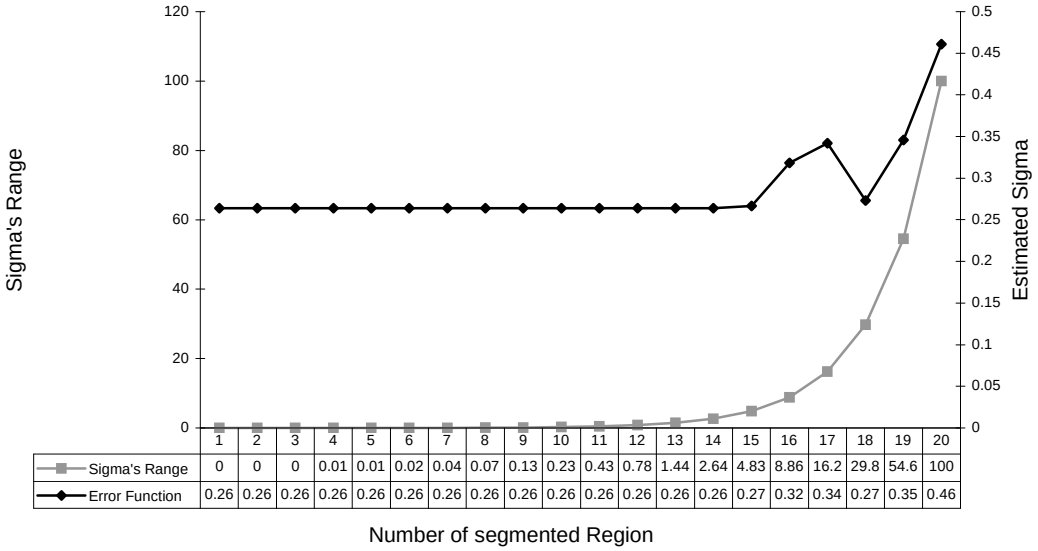


Fig 15: Error estimation of common σ for a classification between middle and illegible handwriting using 36 extracted features (σ =0.01386).



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Sigma's Range	0	0	0	0.01	0.01	0.02	0.04	0.07	0.13	0.23	0.43	0.78	1.44	2.64	4.83	8.86	16.2	29.8	54.6	100
Error Function	0.26	0.26	0.26	0.26	0.26	0.26	0.26	0.26	0.26	0.26	0.26	0.26	0.26	0.26	0.27	0.32	0.34	0.27	0.35	0.46

Fig 16: Error estimation of common σ for a classification between middle and legible handwriting using 36 extracted features (σ =7.11064).

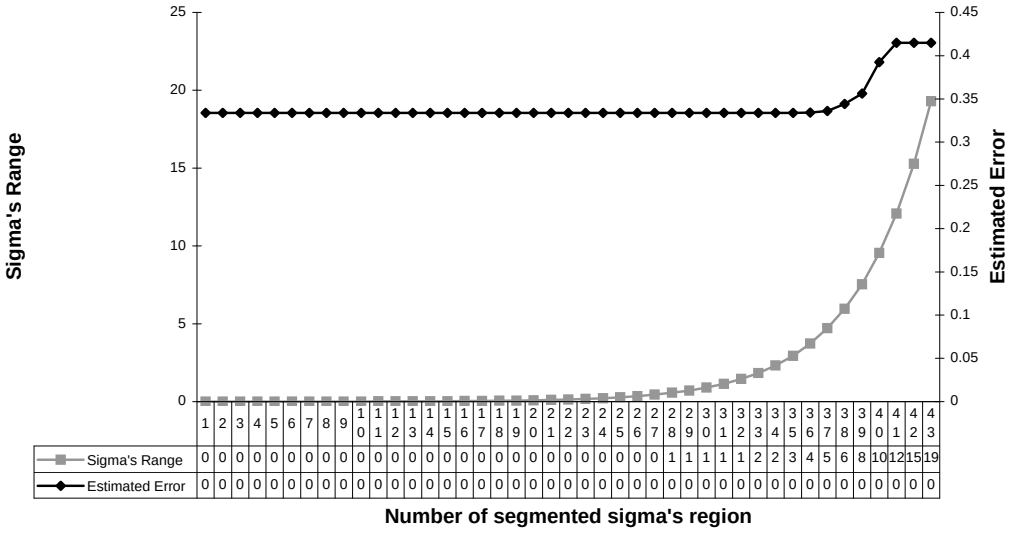


Fig 17: Error estimation of common σ for a classification between legible, illegible and middle handwriting using 36 extracted features (σ =0.001).