Authentication of Individuals using Hand Geometry Biometrics: A Neural Network Approach

Article *in* Neural Processing Letters · October 2007

DOI: 10.1007/s11063-007-9052-y · Source: DBLP

Marcos Faundez-Zanuy
Fundació TecnoCampus Mataró-Maresme
290 PUBLICATIONS **3,499** CITATIONS
SEE PROFILE

David A. Elizondo
De Montfort University
62 PUBLICATIONS **779** CITATIONS
SEE PROFILE

Carlos M. Travieso
Universidad de Las Palmas de Gran Canaria
372 PUBLICATIONS **2,751** CITATIONS
SEE PROFILE

Neural Process Lett (2007) 26:201–216
DOI 10.1007/s11063-007-9052-y

Authentication of Individuals using Hand Geometry Biometrics: A Neural Network Approach

Marcos Faundez-Zanuy · David A. Elizondo ·
Miguel-Ángel Ferrer-Ballester ·
Carlos M. Travieso-González

Published online: 7 October 2007

Abstract Biometric based systems for individual authentication are increasingly becoming indispensable for protecting life and property. They provide ways for uniquely and reliably authenticating people, and are difficult to counterfeit. Biometric based authenticity systems are currently used in governmental, commercial and public sectors. However, these systems can be expensive to put in place and often impose physical constraint to the users. This paper introduces an inexpensive, powerful and easy to use hand geometry based biometric person authentication system using neural networks. The proposed approach followed to construct this system consists of an acquisition device, a pre-processing stage, and a neural network based classifier. One of the novelties of this work comprises on the introduction of hand geometry's related, position independent, feature extraction and identification which can be useful in problems related to image processing and pattern recognition. Another novelty of this research comprises on the use of error correction codes to enhance the level of performance of the neural network model. A dataset made of scanned images of the right hand of fifty different people was created for this study. Identification rates and Detection Cost Function (DCF) values obtained with the system were evaluated. Several strategies for coding the outputs of the neural networks were studied. Experimental results show that, when using *Error Correction Output Codes (ECOC)*, up to 100% identification rates and 0% DCF can

M. Faundez-Zanuy
Escola Universitària Politècnica de Mataró,
Barcelona, Spain
e-mail: faundez@eupmt.es

D. A. Elizondo (✉)
Centre for Computational Intelligence, School of Computing,
Faculty of Computing Sciences and Engineering, De Montfort University,
Leicester, UK
e-mail: elizondo@dmu.ac.uk

M.-Á. Ferrer-Ballester · C. M. Travieso-González
Universidad de Las Palmas de Gran Canaria Departamento de Señales y Comunicaciones,
Las Palmas de Gran Canaria, Spain
URL: www.gpds.ulpgc.es

be obtained. For comparison purposes, results are also given for the Support Vector Machine method.

Keywords Biometrics · Hand geometry · Neural network · Biometrical features · Feature extraction · Feature identification · Authentication of individual

1 Introduction

Biometric authentication systems are becoming increasingly indispensable for protecting life and property. They are robust and reliable, difficult to counterfeit, and can authenticate individuals quickly [1]. Some biometric techniques commonly used for user authentication include fingerprint scan identification, facial geometry verification, retina scan identification, and hand geometry recognition [23].

In recent years, hand geometry recognition has become a very popular biometric access control, and it is used in around a quarter of real world physical access control applications [11]. Although fingerprint [4,9] is the most popular access system, the study of other biometric systems is interesting, because the reliability of a biometric system [5] can be improved by using some kind of data fusion [7] between different biometric traits. This is a key point in order to popularise biometric systems [6,8].

Typical systems for authentication of individuals consist of a data acquisition device (a digitalisation device) , a preprocessing stage (software performing data pre-processing operations such as the extraction of features), and the construction of a classifying model based on this features. Previous research on people authentication using biometric indicators from hand images include [15]. The authors propose the use of a Gaussian mixture model (GMM), normally used for speaker identification, to people verification based on hand geometry biometrics. The features used for the development of the GMM algorithm were extracted from colour hand images (both the top and the side view of the hands). The edges of the hands are detected from the images and a morphological analysis is performed, obtaining a feature vector. The authors obtain error rates below 5%.

A feature-based hierarchical framework for hand geometry recognition, based upon matching of geometrical and shape features was proposed by [22]. Geometrical significant landmarks are extracted from the segmented images, and are used for hand alignment and the construction of recognition features. The recognition process uses a Gaussian Mixture Model for the first group of features, followed by a distance metric classification of the second group of features if necessary. The method provides results of 89% hit true acceptance rate (hit) and 2.2% false acceptance rate.

Another system for personal identification using hand images is presented in [12]. The authors attempt to improve the performance of palm print based verification systems by using hand geometry features. Both palm print and hand geometry features are extracted from the same image. Experimental results show improvements, on the level of generalisation, by combining hand geometry features with those from palm prints.

This paper introduces a powerful, inexpensive and easy to use hand geometry based biometric individual authentication system using neural networks. The proposed approach consists of the use of an office scanner for the acquisition of digital images, the proposition of methods for the pre-processing of these images, including contour detection and contour extraction, and the creation of a multilayer neural network based classifier for the authentication of individuals. One of the main original aspects introduced in this work include the introduction of hand geometry's related, position independent, feature extraction and identification system which can be useful in problems related to image processing and pattern recognition. Another original aspect of this research consists on the use of error correction codes and learning issues (over learning and local minima avoidance) that can enhance the level of performance of the multilayer neural network model. The error correction codes prove useful when linked to the creation of the structure of the neural network model (topology in terms of the output layer).

A dataset containing ten scanned images of the right hand of fifty different individuals was created for this study. Three dimensional hand profiles are commonly used on commercial biometric systems since they provide more information than two dimensional ones. However, they also require more expensive and sophisticated hardware. Two dimensional profiles of a hand can be obtained using a document scanner or a low cost digital camera. Therefore, a system based on two dimensional profiles, using a conventional and inexpensive office digital document scanner, is presented. Despite the drawback of loss of information encountered by using 2D images as opposed to 3D ones, the data sets seem to contain enough information to produce a robust authentication system.

Using an innovative approach, biometrical features were extracted from the dataset and used to develop a neural network for authentication of individuals. Several schemas for encoding the output layer of the neural network were studied. The proposed authentication model offers identification rates and detection cost function values of up to 100% and 0% respectively and therefore, can be combined with other biometric traits to overcome the vulnerability of biometric systems.

This paper is divided into five sections. The second section describes the different steps taken in order to gather the hand images used to extract biometrical features. In the third section, a description of the pre-processing and biometrical feature extraction from the hand images is presented. The fourth section provides details of the implementation of the neural network based authentication system. Section 5 presents some of the results obtained with the proposed authentication system. For comparison purposes, some results obtained with the Support Vector Machine method [2] are presented. Finally, some conclusions are presented in Sect. 6.

2 Data Collection and Preprocessing

Digital scanned images of individual hands were collected and preprocessed. The preprocessing of the images involved the reduction in resolution of the images, the conversion of the colour 8 bit images into monochrome 1 bit ones, the detection of hand contours, and the location and measurement of hand measure points. The goal of this preprocessing was to extract a set of appropriate features in order to compose the MLP classifier's 10 components input patterns (input vectors).

A conventional desktop document scanner, where the user can place the hand palm freely over the scanning surface, was used. No pegs, templates or any other physical constraints for the users were used to capture the images [20]. This makes the system hand orientation independent and therefore, very easy to use. The images contain eight bits per pixel (256 grey levels), and a resolution of 150 dpi. To facilitate later computation, the resolution of every image was reduced by a factor of 20%. Each image file has a size of 1405 KB and a resolution of 1403 $\times$ 1021 pixels. Figure 1 shows an example of a scanned image of a hand.

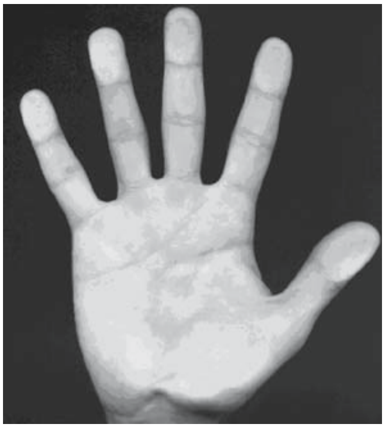


Fig. 1 Example of a scanned image of a hand

Ten images of the right hand, of fifty different people, were collected for a total of 500 images. The ages of the individuals used on this study ranged from 23 to 30 years old. To make the task more rigorous, the users selected for the experiments had approximately equal hand characteristics.

A total of 57 users were needed in order to produce a data set containing images of 50 users that were free of errors. The only images that were removed were those with hands hidden by clothes, images with the tip of the fingers out of the image, etc. The entire set of images from a user was discarded from the data set when one of his/her 10 scanned images presented any of the errors described above. After only a few trials, the users were familiarised with the system, which was easily operated.

The reason why this procedure was followed for collecting the data is because in many pattern recognition applications, usually there are a limited number of classes and a great number of training samples. Such is the case, for instance, with the recognition of handwritten digits (10 classes) from US postal envelopes. This is reversed in biometrics where three to five measurements per person, during enrolment, are taken (three to five samples for training per class), and the people enrolled in the database is large (much more than samples per class). In these situations, there are not enough training samples and the mean vector nor the covariance matrix can be estimated accurately. In fact, a small number of training samples is a desirable property of biometric systems. Otherwise, the enrolment procedure for a new user will be time consuming and the system will lack of commercial value.[1]

3 Biometrical Feature Extraction

Biometrical features, to be used on the training of the neural network model, were extracted from the scanned hand images. Since the subject of hand recognition is relatively new. As a consequence, there is no clear description in the literature about the biometrical features from hand images that best work for individual authentification. Therefore, based on experimentation, the following steps for extracting biometrical features were considered: binarisation, contour extraction, and computation of the geometric measurements. These features were then stored in a new reduced data set (see Fig. 2).

[1] Available at http://www.gpds.ulpgc.es/download/index.htm.

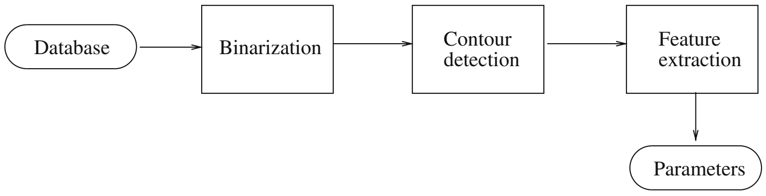
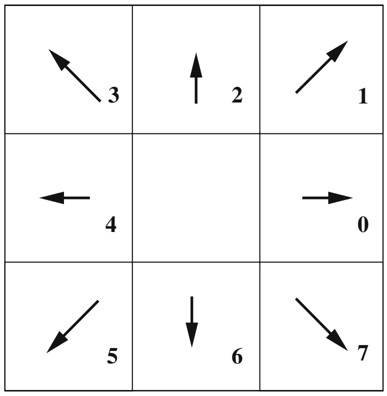


Fig. 2 Steps in the feature extraction process

Fig. 3 Contour coding algorithm using eight directions



3.1 Binarisation Process

The goal of this step was the conversion from a 8 bit per pixel image to a monochrome image, containing 1 bit per pixel, by applying a threshold:

$$\begin{cases} InputImage(i, j) < threshold, \; OutputImage(i, j) = 1 \\ InputImage(i, j) \geq threshold, \; OutputImage(i, j) = 0 \end{cases}$$

As the contrast between the image and the background is quite high, it reduces the complexity of the binarisation process. After performing several experiments changing the threshold and evaluating the results with different images extracted from the data set, conclusions were reached that, with a selected threshold of 0.25, the results were adequate for the purposes of this work. The use of other binarisation algorithms such as the one suggested by Lloyd, Ridler-Calvar and Otsu [13] was discarded because the results obtained are similar and the computational burden higher.

3.2 Hand Contour Detection

Contour detection is the process of identifying the edges of a hand from an image which results in a numerical sequence which describes the shape of the hand-palm. Contour following is a procedure by which one runs through the hand silhouette by following the edge of the image. The algorithm implemented for the hand contour detection is a modification of the method created by Sonka, Hlavac and Boyle [16]. In the original algorithm, the authors use a 4-value description (0, 1, 2, 3) to code the direction of each point in the boundary contour. In the implementation used in this paper, account is taken of the eight possible changes of direction following the order of the mask displayed in Fig. 3.

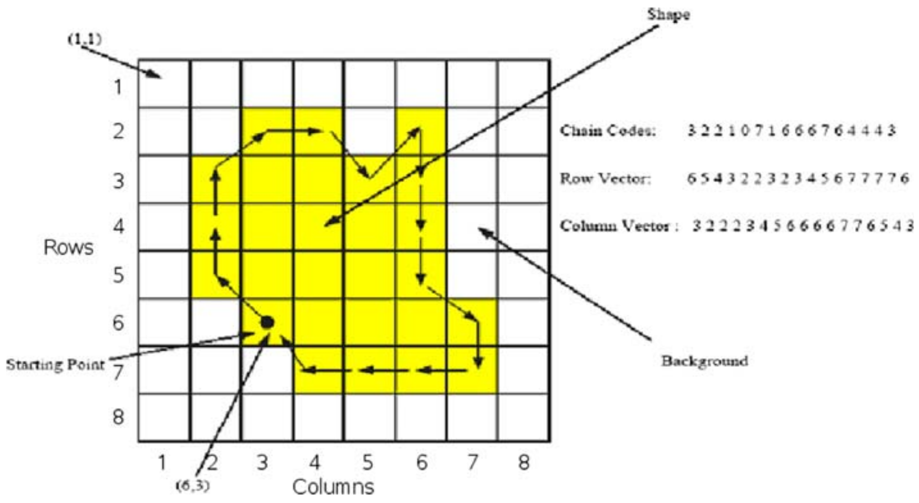

Shape
(1,1)
1
2
3
4
Rows
5
6
Starting Point
7
8
1 2 3 4 5 6 7 8
(6,3)
Columns
Chain Codes: 3 2 2 1 0 7 1 6 6 6 7 6 4 4 4 3
Row Vector: 6 5 4 3 2 2 3 2 3 4 5 6 7 7 7 7 6
Column Vector : 3 2 2 2 3 4 5 6 6 6 6 7 7 6 5 4 3
Background

Fig. 4 Contour following procedure example using the chain code described in Fig. 3. Row and column values correspond to the coordinates of each contour point

The bitmap graphical files are transformed into text files that contain the contour description. The encoding algorithm consists of a chain code. In chain coding the direction vectors between successive boundary pixels are encoded. Our eight directional code can be coded by 3-bit code words. Once the chain code is obtained, the perimeter can be easily computed: for each segment, an even code implies +1 and an odd code $+\sqrt{2}$ units. the start and end points of the fingers and wrist are found by looking for minimum and maximum values of the chain code.

An example of the 8-value description can be seen in Fig. 4, where a shape was drawn and the chain codes and coordinates for every contour point were extracted. The reason to use this method is because chain codes simplify the process for working out lengths, flat, convex and concave areas, which are relevant for a hand-image feature extraction. This asserting has been corroborated by experimental results. The output of the implemented algorithm returns three vectors d, x and y, where d has the direction codes, and $x - y$ the location of every contour point.

There are three stages involved in the detection of a hand contour. The first stage involves looking for the starting vectorisation point. From the valley between the thumb and the first finger, a vertical line is traced down the image until it cuts the contour of the hand. The procedure is shown in Fig. 5. From this point, a horizontal line is traced to find the starting point at the location where the line and the hand silhouette cross each other. The part of the hand located from this line downwards is removed, because it is not useful in the recognition task. The starting coordinates are assigned to the pixel position P_0.

The second stage involves, starting from the current pixel position P_{n-1}, the search, in the clockwise direction starting from (d+3) mod 8, of the first pixel with a value of “0”. This will be the new boundary point. Its coordinate values are then assigned to P_n and the new direction to d.

The third stage checks if the new ordinate is equal to the original ordinate, in which case the detection of contour halts. Otherwise the second step is repeated. Figure 6 shows the result obtained after extracting the contour information from the hand-palm of a sample image.

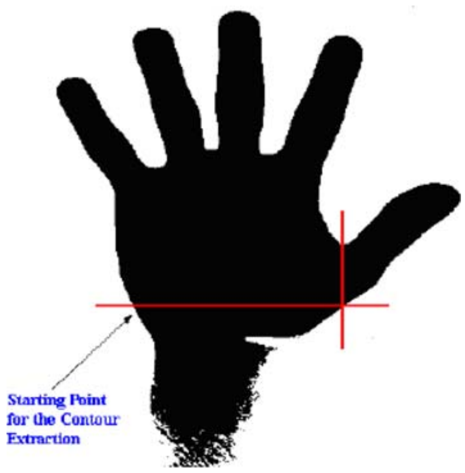


Fig. 5 Looking for the start point for the contour extraction

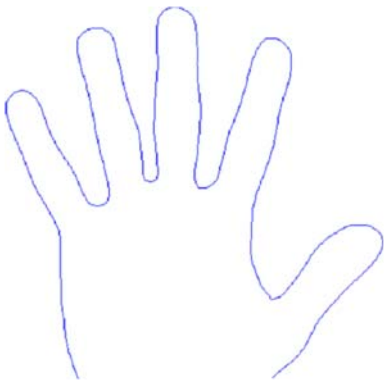


Fig. 6 Contour extracted

3.3 Feature Extraction

Human beings are not used to recognise each other based on hand-geometry. Thus, it is quite difficult to use common sense in order to identify the most appropriate features. This is different from other biometric traits such as signatures, voice, etc., where you can contact a forensic expert and ask him or her for the most discriminative features. This is why in this study, the extraction of features from the hand images is based on the location of main measure points (finger tips, valleys between fingers, etc.) and geometric measurements of the hands (length of the five fingers, distances between main points) from the image data set.

3.3.1 Location of Measure Points

Several intermediate steps were taken in order to detect the main points of the hands from the image data set. The method for the geometric hand-palm features extraction is quite straightforward. From the hand image, the following main points are located: finger tips, valleys between the fingers and three more points that are necessary to define the hand geometry precisely.

Before any further processing can take place, the hand in the binarised image has to be detected. The first two points to be detected are the ones representing the thumb and the little

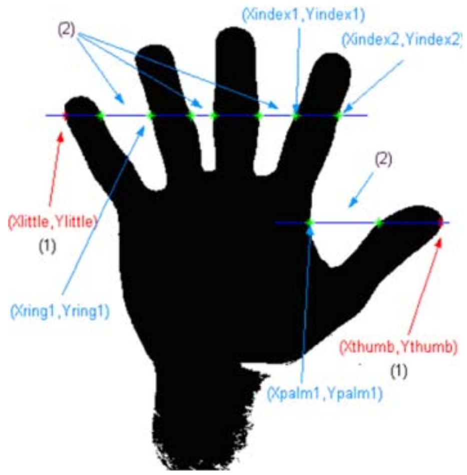


Fig. 7 Geometric process description

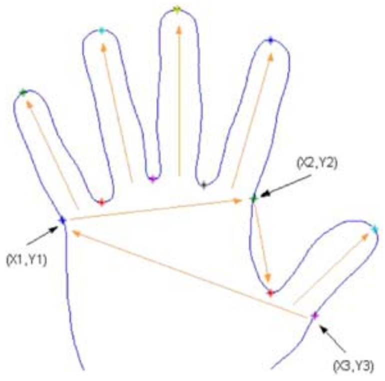


Fig. 8 Geometric measurements

finger. These points (see Fig. 7) enable the extraction of most of the geometrical information and are detected based on the location of the (X_{thumb}, Y_{thumb}), and the (X_{litle}, Y_{litle}) points respectively.

Starting at these two points, namely (X_{little}, Y_{little}), (X_{thumb},Y_{thumb}) for the little and thumb finger respectively, two horizontal lines are traced which cut the hand at several points (see Fig. 7). For example, by tracing a horizontal line from the little finger towards the hand points $(X_{little2}, Y_{little2})$, (X_{ring1}, Y_{ring1}), (X_{ring2}, Y_{ring2}), (X_{midle1}, Y_{midle1}), (X_{midle2}, Y_{midle2}), (X_{first1}, Y_{first1}), and (X_{first2}, Y_{first2}) were found. These points correspond to the crossing points with the little, ring, middle, and first fingers respectively. In the same way, but this time for the thumb, points (X_{thumb1}, Y_{thumb1}) and (X_{palm1}, Y_{palm1}) were extracted, which correspond to the crossing points with the thumb finger and the hand-palm.

Once these points are found, the location of the finger tips and the valleys between them are extracted. The procedures for detecting tips and valleys are very similar. For instance, in order to search for the index finger tip, the points with minimum ordinate found from the point (X_{first1}, Y_{first1}) have to be used. The minimum ordinate is searched in a clockwise direction. The process for locating a valley is the same. For example, to locate the valley between the heart and middle finger, the maximum ordinate points have to be found starting from the point (X_{midle2}, Y_{midle2}).

To complete the hand-palm geometry precisely there are still three important points missing. These points are the symmetry to the points located in the little-ring finger valley, the first-ring finger valley, and the first-thumb valley. The position of these points are presented in Fig. 8. They are labelled as (X_1, Y_1), (X_2, Y_2) and (X_3, Y_3), respectively. To locate these points, the outer points in the contour, with minimum distance to these valleys, need to be found.

3.3.2 Hand Measurements

Finally, using all the main points previously computed, the geometric measurements are obtained. The following eight distances are taken: Length of the 5 fingers, distances between points (X_1, Y_1) and (X_2, Y_2), points (X_2, Y_2) and the valley between the thumb and first finger and the points (X_3, Y_3) and (X_1, Y_1). Figure 8 shows the final results along with the geometric measurements taken into account.

The feature vector contains ten geometric measurements extracted from the hand. These geometric measurements include the eight measurements mentioned above together with the area and the length of the contour.

After examining the data set, it becomes apparent that the major source of variability found among the different parameters is due to the pressing force used to place the hand over the scanning glass surface, which changes for each person and for every image.

4 Model Development

Biometric systems can be approached in two ways. The first way refers to user identification with no identity claim made by the individual. The automatic system must determine who is trying to gain access. The second way deals with verification. In this approach the goal of the system is to determine whether the person is who he or she claims to be. This implies that the user must provide an identity and the system just accepts or rejects the user according to a successful or unsuccessful verification. Sometimes this operation mode is named authentication or detection.

For identification purposes, in a set containing N different users, and a labelled test set, the goal is to count the number of identities correctly assigned. Verification systems can be evaluated using the false acceptance rate (FAR, those situations where an impostor is accepted) and the false rejection rate (FRR), those situations where a user is incorrectly rejected), also known in detection theory as False Alarm and Miss, respectively.

In order to summarise the performance of a given system with a single number, the minimum value of the detection cost function (DCF) was used. This parameter is defined as [3]:

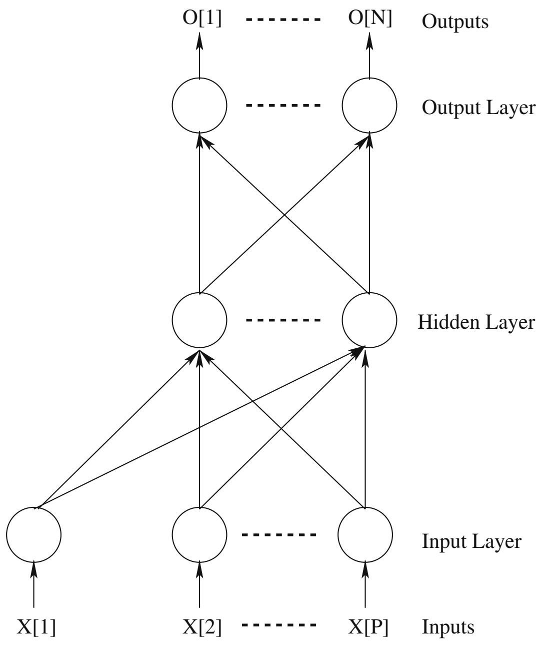
$$DCF = C_{miss} \times P_{miss} \times P_{true} + C_{fa} \times P_{fa} \times P_{false}$$

where C_{miss} is the cost of a miss (rejection), P_{miss} is the probability of a miss (rejection), C_{fa} is the cost of a false alarm (acceptance), P_{fa} is the probability of a false alarm (acceptance), P_{true} is the a priori probability of the target, and $P_{false} = 1 - P_{true}$. In this work $C_{miss} = C_{fa} = 1$ was used.

4.1 Neural Network Model

A multi-layer perceptron (MLP) [14] was developed as a discriminative two class classifier. The output or target values were coded according to the type of person that the input pattern

Fig. 9 Multi-layer perceptron architecture



belonged to. When the input pattern belonged to a genuine person, the output was given a value of 1. When the input pattern represented an impostor person, the output was fixed to -1. Figure 9 shows the architecture of the neural network. This architecture contains 10 input units, a single hidden layer with a total of 40 hidden neurons.

The neural network was trained using the gradient descent algorithm with momentum and weight/bias learning functions. Contrary to other neural network algorithms such as the RDP ([18,19]), the MLP suffers from over learning and local minima. Therefore, the neural network was trained for 2,500 and 10,000 epochs using regularisation which involves a modification on the error function to avoid over learning. Also, a multi-start algorithm was also applied to avoid local minima. The mean, standard deviation, and the best results obtained after 50 random different initialisations are presented.

The input signals were normalised within a $[-1, 1]$ range for each input component.

The technique of regularisation was used to help with the problem of over-fitting. This technique involves modifying the error function. This function is normally chosen to be the sum of squares of the network errors on the training set which leads to good networks that are not over-trained and generalise well.

The classical Mean square error (MSE) implies the computation of:

$$MSE = \frac{1}{N}\sum_{i=1}^{P}(t_i - a_i)^2$$

where t and a are the P dimensional vectors of the observed and neural network predicted output values, respectively. The regularisation uses the following measure:

$$MSEREG = \gamma MSE + (1-\gamma)\frac{1}{N}\sum_{j=1}^{n} W_j^2$$

Thus, it includes one term proportional to the modulus of the weights of the neural net.

Another factor that can influence the level of generalisation of a neural network is the initialisation of the connection weights. This is why for every MLP model developed in this work, the connection weights were randomly initialised and the model trained. This process was repeated fifty times for each MLP model.

4.2 Multi-class Learning Problems via Error-correction Output Codes

Multi-class learning problems involve finding a definition for an unknown function $f(\vec{x})$ whose range is a discrete set containing $k > 2$ values (i.e., k classes), and $\vec{x}$ is the set of measurements that we want to classify. The definition is acquired by studying large collections of training examples of the form $\{\vec{x}, f(\vec{x_i})\}$.

The problem of learning a k-ary classification function $f : I\!R^n \rightarrow \{1, \ldots, k\}$ from examples of the form $\{\vec{x}, f(\vec{x_i})\}$, has to be solved. The standard neural network approach to handle this problem is to construct a 3-layer feed-forward network with k output units, where each output unit designates one of the k classes. During training, the output units are set to a value of 0.0, except for the unit corresponding to the desired class, which is set to a value of 1.0. During classification, a new $\vec{x}$ value is assigned to the class whose output unit has the highest activation. This approach is called the *one-per-class* approach [10,17], since one binary output function is learnt for each class.

Error-control coding techniques [21] detect and possibly correct errors that occur when messages are transmitted in a digital communication system. To accomplish this, the encoder transmits not only the information symbols, but also one or more redundant symbols. The decoder uses the redundant symbols to detect, and possibly correct, whatever errors occurred during transmission.

Block coding is a special case of error-control coding. Block coding techniques map a fixed number of message symbols to a fixed number of code symbols. A block coder treats each block of data independently and it is a memory less device. The information to be encoded consists of a sequence of message symbols. The code that is produced consists of a sequence of code words. Each block of k message symbols is encoded into a code word that consists of n symbols. In this context, k is called the message length, n is called the code word length, and the code is called an $[n, k]$ code.

A message for an $[n, k]$ BCH (Bose–Chaudhuri–Hocquenghem) code must be a k-column binary Galois array. The code that corresponds to that message is an n-column binary Galois array. Each row of these Galois arrays represents one word.

BCH codes use special values for n and k. The code word length, n, is an integer of the form $2^m - 1$ for some integer $m > 2$. The message length, k, is a positive integer less than n. However, only some positive integers less than n are valid choices for k. The code can correct all combinations of t or fewer errors, and the minimum distance between codes is:

$$d_{min} \geq 2t + 1$$

Table 1 shows some examples of suitable values for BCH codes.

The *error-correcting output coding* (ECOC) ([10,17]) is an alternative method for encoding the output layer of a neural network that can improve the level of generalisation. In this approach, each class i is assigned an m-bit binary string, c_i, called a code word. The strings are chosen (by BCH coding methods) so that the Hamming distance between each pair of strings is guaranteed to be at least d_{min}. During training on example $\vec{x}$, the values of the m output units of a 3-layer network are set to the appropriate binary string $C_{f(\vec{x})}$. During classification, the new example $\vec{x}$ is assigned to the class i whose codeword c_i is closest

Table 1 Examples of suitable values for BCH codes

n	k	t
7	4	1
15	11	1
	7	2
	5	3
31	26	1
	21	2
	16	3
	11	5
	6	7

(in Hamming distance) to the m-element vector of output activations. A big advantage of this approach is that it can recover from any $t = \left[\frac{d_{min}-1}{2}\right]$ errors in learning the individual output units. Error-correcting codes act as ideal distributed representations.

In [10, 17] some improvements, in terms of the level of generalisation, using this strategy were presented. The strategy was applied to several benchmark classification data sets including, the vowel, letter, soybean data sets.

4.3 Neural Network Design

By looking at the output codes (targets) learnt by the neural network when the input pattern $\vec{x} \in k$ user, it can be observed that just the output number k is activated, and that the number of output units is equal to the number of users. This approach will be named *one-per-class*, and it will have a $10 \times 40 \times 50$ MLP architecture.

An alternative coding method is the use of *natural binary coding*, which provides a reduced number of outputs, because forty different alternatives can be represented using just six bits. Thus, this approach will be need a $10 \times 40 \times 6$ MLP architecture.

Another approach for coding the output layer is to assign to each user a different output code. These codes can be selected from the first fifty BCH (15, 7) codes, BCH (31, 6), etc. For instance, the BCH (15,7) code yields up to $2^7 = 128$ output codes. However, only fifty are needed because this is the number of users. On the other hand, it can be observed that the first output of BCH (15, 7) is always the same. Thus, this output can be removed, lowering the number of output units on the neural network to fourteen.

The BCH (15, 7) code provides a minimum distance of five bits between different codes, while the *one-per-class approach* provides a minimum distance of two bits, and the natural binary coding a minimum distance of one bit. Also, the BCH (15, 7) provides a more balanced amount of ones and zeros, while in the one-per-class approach almost all the outputs will be zeros.

A good error-correcting output code for a k-class problem should satisfy two properties [10]. First, each codeword should be well-separated in terms of the Hamming distance from each of the other codewords (*Row separation*). Second, each bit-position function f_i should be uncorrelated from the functions to be learnt for the other bit positions f_j, $j \neq i$ (*Column separation*).

Error-correcting codes only succeed if the errors made in the individual bit positions are relatively uncorrelated, so that the number of simultaneous errors in many bit positions is small. For this purpose, the algorithm proposed in [15] for random ECOC generation was used.

All the MLP neural networks had an input layer containing ten neurons. After several trials, the number of hidden units in the hidden layer was fixed to forty units. The number of neu-

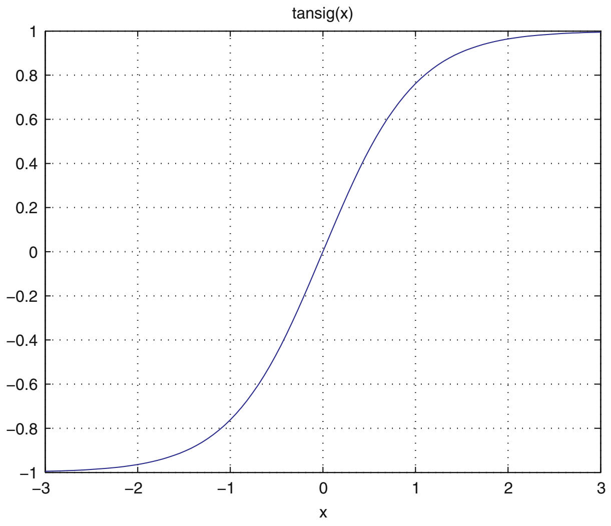


Fig. 10 Tansig non-linear transfer function

Table 2 MLP 10 × 40 × 50 (one-per-class)

Epoch	Train=5 hands, test=5 hands						Train=3 hands, test=7 hands					
	Identif. rate (%)			Min(DCF)			Identif. rate (%)			Min(DCF)		
	mean	σ	max	mean	σ	min	mean	σ	max	mean	σ	min
2500	98.30	0.39	98.80	0.69	0.2	0.34	97.71	0.70	99.14	0.86	0.2	0.50
10000	98.23	0.47	99.20	0.67	0.16	0.37	97.79	0.64	98.57	0.86	0.18	0.53

rons in the output layer varied according to the coding strategy used. The number of neurons in each layer of the MLP will be summarised using the following nomenclature: $inputs \times hidden \times output$. A similar neural network architecture has been successfully used by the authors in previous studies involving face recognition, speech coding and speaker recognition using neural networks.

The $tansig$ non-linear transfer function (see Fig. 10) was used in all the layers. This transfer function is symmetrical around the origin. Thus, the output codes were modified replacing each "0" by a "-1". In addition, the input vectors $\vec{x}$ were normalised for zero mean and maximum modulus equal to 1.

The computation of the mean square error (MSE), and the mean absolute difference (MAD) between the neural network predicted output value and each of the codewords provides a distance measure. This measure was converted into a similarity measure computing (1 - $distance$).

The following strategies were evaluated: *One-per-class*: 1 MLP 10 × 40 × 50 (Table 2), *Natural binary code*: 1 MLP 10 × 40 × 6 (Table 3), *Error correction output code* (ECOC) using BCH (15, 7) (Table 4) and BCH (31, 6) (Table 5)), and *error correction output code* (ECOC) using random generation (Table 6).

Neural networks for testing each coding strategy were developed using the entire set of 500 images (ten images × fifty individuals). For each of the fifty individuals, the ten hand images were divided into two subsets. The first subset contained five hands for training and five hands for testing. The second data set contained three hands for training and seven hands for testing.

Table 3 MLP 10 × 40 × 6 (Natural binary code)

	Epoch	Train=5 hands, test=5 hands						Train=3 hands, test=7 hands					
		Identif. rate (%)			Min(DCF)			Identif. rate (%)			Min(DCF)		
		mean	σ	max	mean	σ	min	mean	σ	max	mean	σ	min
MAD	2500	95.97	1.25	98.4	3.94	0.52	2.74	92.43	1.49	96.57	5.70	0.45	4.79
	10000	96.42	1	98.4	3.80	0.49	2.77	92.66	1.17	95.14	5.58	0.39	4.81
MSE	2500	95.97	1.25	98.4	0.88	0.3	0.38	92.43	1.49	96.57	2.60	0.41	1.87
	10000	96.42	1	98.4	0.83	0.3	0.29	92.66	1.17	95.14	2.53	0.42	1.60

Table 4 MLP 10 × 40 × 50 (ECOC BCH (31, 6))

	Epoch	Train=5 hands, test=5 hands						Train=3 hands, test=7 hands					
		Identif. rate (%)			Min(DCF)			Identif. rate (%)			Min(DCF)		
		mean	σ	max	mean	σ	min	mean	σ	max	mean	σ	min
MAD	2500	99.58	0.15	100	0.04	0.05	0	98.54	0.60	99.71	0.49	0.21	0.12
	10000	99.62	0.21	100	0.03	0.04	0	98.50	0.60	99.43	0.45	0.18	0.11
MSE	2500	99.58	0.15	100	0.03	0.04	0	98.59	0.57	99.71	0.46	0.20	0.0
	10000	99.62	0.22	100	0.02	0.03	0	98.53	0.59	99.43	0.43	0.17	0.11

Table 5 MLP 10 × 40 × 14 (ECOC BCH (15, 7))

	Epoch	Train=5 hands, test=5 hands						Train=3 hands, test=7 hands					
		Identif. rate (%)			Min(DCF)			Identif. rate (%)			Min(DCF)		
		mean	σ	max	mean	σ	min	mean	σ	max	mean	σ	min
MAD	2500	99.58	0.15	100	0.04	0.05	0	98.06	0.58	99.43	0.94	0.26	0.49
	10000	99.62	0.21	100	0.03	0.04	0	98.30	0.58	99.43	0.85	0.26	0.44
MSE	2500	99.58	0.15	100	0.03	0.04	0	98.07	0.58	99.14	0.47	0.18	0.18
	10000	99.61	0.22	100	0.02	0.03	0	98.35	0.61	99.43	0.39	0.19	0.06

Table 6 MLP 10 × 40 × 50 (random ECOC BCH generation)

	Epoch	Train=5 hands, test=5 hands						Train=3 hands, test=7 hands					
		Identif. rate (%)			Min(DCF)			Identif. rate (%)			Min(DCF)		
		mean	σ	max	mean	σ	min	mean	σ	max	mean	σ	min
MAD	2500	99.50	0.23	100	0.26	0.12	0.01	98.09	0.78	99.71	1.22	0.38	0.41
	10000	99.58	0.23	100	0.23	0.12	0	98.30	0.70	99.71	1.10	0.31	0.60
MSE	2500	99.50	0.21	100	0.13	0.01	0.004	98.14	0.78	99.71	0.85	0.33	0.22
	10000	99.58	0.23	100	0.09	0.09	0	98.32	0.67	99.71	0.71	0.32	0.09

5 Results

Tables 2 and 3 provide results obtained with the *one-per-class* and the *natural binary code* output strategies. One hand per person (the right hand of each person) was always used. Three different snapshots of the same hand per person were used for training purposes. The hands were removed and placed again on the desktop document scanner surface to avoid consecutive identical acquisitions. Thus, they are different because translation, rotation, finger position etc. varies but they belong to the same physical hand.

These experimental results show a better performance, in terms of the percentage of identification rate and DCF, when using the *one-per-class* output approach. Mean gains in the level of identification rate and detection cost function of around 3% and 5% and below 1% respectively, can be observed when using the *one-per-class* strategy. The gain obtained with the *one-per-class* strategy could be due to the larger number of weights when using this strategy which leads towards a better classifier. Additionally, it can be interpreted that the larger hamming distance of the *one-per-class* approach helps to improve the results.

The error correction output coding scheme leads to more flexibility with respect to the MLP architecture. This is because a wide range of possibilities for the number of outputs, given a determined set of users, was available. In addition, experimental results outperformed the *one-per-class* approach with an average performance of 99% and reaching up to 100% in maximum identification rates.

Very similar identification rates of up to 100% were obtained with both the *ECOC BCH* (15, 7) and the *random ECOC BCH generation* (Tables 4 and 5). Slight better values for the detection cost function, with up to 0%, were obtained with the *ECOC BCH* (15,7) approach. Thus, the *BCH*(15, 7) approach should be preferred because it is simpler in terms of the architecture of the network.

It was expected that the *random generation for ECOC* approach should out perform the *BCH codes* one. However, the experimental results reveal a better performance when using the latter one. According to these results, the relevance of *ECOC* strategies is more important in verification applications than in identification ones.

For comparison purposes, Support Vector Machine models ([2]) were developed. The best result obtained with these models, in terms of the minimum value of the DCF, gives a value of 0.86 which is double the mean value obtained with the neural network with a mean value of 0.43 and a minimum value of 0 (see Table 4).

6 Conclusions

A novel method for individual authentication using neural networks and hand geometry biometrics was presented. Contrary to some of the current biometric methods, this method proves to be easy to implement and use as well as inexpensive to deploy.

Several experiments with different topologies and output coding schemes were conducted. The output coding strategies studied included *one-per-class*, *natural binary code*, *error correction output* using the Bose–Chaudhuri–Hocquengbem code, and *error correction output* using random generation. All the neural networks used 10 and 40 neurons for the input and hidden layers respectively. Only the output layer was different for each coding schema.

Over all, the best levels of performance in the experiments were achieved by using five hands to train and five to test as opposed to three hands for training and seven for testing (for each set of images from the 50 individuals). Through out the experiments, no significant performance improvement was obtained when augmenting the number of epochs from 2500 to 10000.

The results obtained in this work show that the proposed neural network system for individual authentication, based on hand geometry biometrics and using the *BCH*(15, 7) output code method, is a reliable, easy to use and inexpensive alternative to other biometric based systems. This model offers identification rates and detection cost function values of up to 100% and 0% respectively and therefore, can be combined with other biometric traits to overcome the vulnerability of biometric systems.

At present, the entire data set contains hands corresponding to exclusively European people. Future research will include the creation of a larger data set containing more individuals of different origins. After performing several experiments, the threshold value, used for the conversion of the colour 8 bits images into black and white 1 bit ones, was set to 0.25. The new data set will probably require the adjustment of this threshold.

The reduction on the resolution of the image files should also be considered since the hand geometrical approach might not require a high resolution. This will reduce the amount of computer power needed to run the system.

References

1. Bleumer G (1999) Biometric authentication and multilateral security. In: Mller G, Rannenberg K (eds) Multilaterial security in communications, Addison-Wesley, pp 157–172
2. Cristianini N, Shawe-Taylor J (2003) An introduction to support vector machines, vol 1. Cambridge University Press
3. Martin A et al (1997) The det curve in assessment of detection performance. In: European speech Processing Conference Eurospeech, vol 4. pp 1895–1898
4. Faundez-Zanuy M (2004a) Door-opening system using a low-cost fingerprint scanner and a pc. IEEE Aerosp Electron Syst Mag 19(8):23–26
5. Faundez-Zanuy M (2004b) On the vulnerability of biometric security systems. IEEE Aerosp Electron Syst Mag 19(6):3–8
6. Faundez-Zanuy M (2005) Biometric recognition: why not massively adopted yet? IEEE Aerosp Electron Syst Mag 20:25–28
7. Faundez-Zanuy M (2005a) Data fusion in biometrics. IEEE Aerosp Electron Syst Mag 20(1):34–38
8. Faundez-Zanuy M (2005b) Privacy issues on biometric systems. IEEE Aerosp Electron Syst Mag 20(2):13–15
9. Faundez-Zanuy M, Fierrez-Aguilan J (2005c) Testing report of a fingerprint-based door-opening system. IEEE Aerosp Electron Syst Mag 20(6):18–20
10. Dietterich TG, Bakiri G (1991) Error-correcting output codes: a general method for improving multiclass inductive learning programs. In: Proceedings of the 9th national conference on artificial intelligence (AAAI-91), AAAI Press, Anaheim, CA
11. Jain AK, Bolte R, Pankari S (1999) Introduction to biometrics in Biometrics Personal identification in networked society. Kluwer Academic Publishers
12. Kumar A, Wong D, Shen H, Jain A (2003) Personal verification using palmprint and hand geometry biometric. In: AVBPA03. pp 668–678
13. OGorman L, Kasturi R (1995) Document image analysis. IEEE Computer Society Press
14. Haykin S (1999) Neural nets. A comprehensive foundation, 2nd edn. Prentice Hall
15. Sanchez-Reillo R (2000) Hand geometry pattern recognition through gaussian mixture modeling. In: 5th international conference on pattern recognition (ICPR’00), vol 2
16. Sonka M, Hlavac V, Boyle R (1998) Image processing analysis and machine vision, 2nd edn. Thomson-Engineering, September 1998
17. Dietterich T (1991) Do hidden units implement error-correcting codes? Technical report, Oregon State University, 1991
18. Tajine M, Elizondo D (1997) The recursive deterministic perceptron neural network. Neural Networks 11:1571–1588
19. Tajine M, Elizondo D (1998) Growing methods for constructing recursive deterministic perceptron neural networks and knowledge extraction. Artif Intell 102:295–322
20. Travieso-González CM, Alonso JB, David S, Ferrer-Ballester MA (2004) Optimization of a biometric system identification by hand geometry. Complex systems intelligence and modern technological applications, pp 581–586, September 2004
21. Wicker SB (1995) Error Control Systems for Digital Communication and Storage. Prentice Hall, Upper Saddle River, NJ
22. Wong ALN, Shi P (2002) Peg-free hand geometry recognition using hierarchical geometry and shape matching. In: IAPR workshop on machine vision applications, Nara, pp 281–284, Japan, December 2002
23. Yoruk E, Konukoglu E, Sankur B, Darbon J (2006) Shape-based hand recognition. IEEE Trans Image Process 15(7):1803–1815