Pattern Anal Applic (2012) 15:113–120
DOI 10.1007/s10044-012-0270-1

SURVEY

Confidence voting method ensemble applied to off-line signature verification

Juan Ramón Rico-Juan · José M. Iñesta

Received: 5 August 2009 / Accepted: 13 March 2012 / Published online: 8 April 2012

Abstract In this paper, a new approximation to off-line signature verification is proposed based on two-class classifiers using an expert decisions ensemble. Different methods to extract sets of local and a global features from the target sample are detailed. Also a normalization by confidence voting method is used in order to decrease the final equal error rate (EER). Each set of features is processed by a single expert, and on the other approach proposed, the decisions of the individual classifiers are combined using weighted votes. Experimental results are given using a subcorpus of the large MCYT signature database for random and skilled forgeries. The results show that the weighted combination outperforms the individual classifiers significantly. The best EER obtained were 6.3 % in the case of skilled forgeries and 2.31 % in the case of random forgeries.

Keywords Off-line signature verification · A posteriori probability · Combination of classifiers

1 Introduction

Signature verification is one of the most important research areas in the field of person authentication using biometric techniques. Some examples of applications are personal identity verification for access control, banking applications, electronic commerce, etc. A detailed state of the art on signature verification is published in [4].

There are two well known categories of verification systems: on-line [5, 11] and off-line systems [3, 18]. In the former, the signature signal is captured by an electronic device, such as pen tablets, digitizers, sensitive screens in PDAs or mobile telephones, etc. during the writing process, thus providing dynamic information on pressure, direction, ordering, and timing for global signature writing or local events like individual strokes, speed and acceleration features, pen-ups, etc. By contrast, in off-line systems the signature is captured once the writing process is finished, usually by document scanning, so only a static image is available.

Off-line signature verification systems have less information than on-line systems, because the former can only use static features related to the signature shape as it appears in the scanned image. Hence, the author-sensitive characteristics described above are not available in the off-line case and it is also difficult to segment signature strokes due to highly stylish and unconventional writing styles that exist. Nevertheless, due to the ease of use of these off-line systems—as compared to on-line methods that require special hardware to capture the dynamic features—a number of applications prefer this approach.

In the signature verification task, there are three types of forgeries, related to intra and inter-personal variability: random forgeries, when the impostor has no previous knowledge about the genuine signature or name; simple forgeries, represented by a signature sample made with the genuine writer's name; and skilled forgeries, where the impostor has knowledge about the genuine signature model and imitates it. Obviously, random forgeries are less difficult to reject than skilled forgeries. Methods based on the off-line approach are usually more suitable for identifying

J. R. Rico-Juan (✉) · J. M. Iñesta
Departamento de Lenguajes y Sistemas Informáticos,
Universidad de Alicante, 03071 Alicante, Spain
e-mail: juanra@dlsi.ua.es

J. M. Iñesta
e-mail: inesta@dlsi.ua.es

random and simple forgeries, while a skilled forgery has a similar shape to that of the genuine signature and is therefore more difficult to detect.

In this work, we propose an expert decisions ensemble using a confidence voting method approach for off-line signature verification for all types of forgeries. We use the term "expert" in the context of ensemble methods for designing what a single classifier performs in any pattern recognition task. So, the idea is to apply a combination of classifier techniques (combination of experts) to solve the verification task in a more robust manner. These strategies have been widely applied in classification tasks and obtained better results than individual classifiers used on their own [6].

There are many ways to apply a combination of classifiers [6]. In particular, methods based on decision confidences, like those reported by van Erp et al. [16], allow the individual classifier decisions to be weighted in order to obtain a good classification performance.

Another category of ensemble-based studies are those using confidence measures for each single classifier based on a posteriori probabilities using the Bayes theory, as in [15], where in order to solve a multiple class problem, linear discriminant functions are calculated between each class and the rest. The normalized sigmoids of the distances to the discriminant functions are taken as the estimates for the posterior probabilities. On the other hand, Arlandis et al. [2] proposed an interesting method based on the k-NN, focused only on the distances to the k-NN prototypes, assigning a zero probability to the prototypes outside that neighborhood. Nevertheless, in most cases, the zero probability is not a realistic situation. A revised formula was proposed by Rico-Juan and Iñesta [12], and involved finding the nearest neighbor to every class, computing their inverse distances, and normalizing them to estimate their posterior probabilities. This way, every class has a non zero probability.

In this paper, we describe in detail the appropriate features, the individual classifiers, and their suitable combination in order to perform an off-line biometric verification system. Contrary to what happens in other related studies, like those by Abreu and Fairhurst [1], where a weighted sum-based fusion method is proposed in order to ensemble different classifier votes and the weights used in the fusion need to be estimated, or in [13] where SIFT descriptors of a signature image are used as features and a Naive Bayes classifier in used for verification, in our approach, the use of a function distance automatically estimates the confidence of each individual classifier (see Sect. 3).

In Sect. 2, we explain the different local and global features that were chosen to describe the signature image. In Sect. 3, the results obtained when applying different verification systems to a database of signatures are shown. Finally, some conclusions and future lines of work are presented.

2 Signature feature extraction from a binary image

The main idea of this feature extraction stage is to obtain different kinds of features to use with a specific verification system. Four different representations of the same image (Fig. 1a) were designed in order to obtain relevant information. The goal of the first three methods is to summarize the image matrix information in rectangular regions that represent signature shapes (Fig. 1b) as local features. The fourth method extracts global information applying some morphological erosions [14] until the signature disappears.

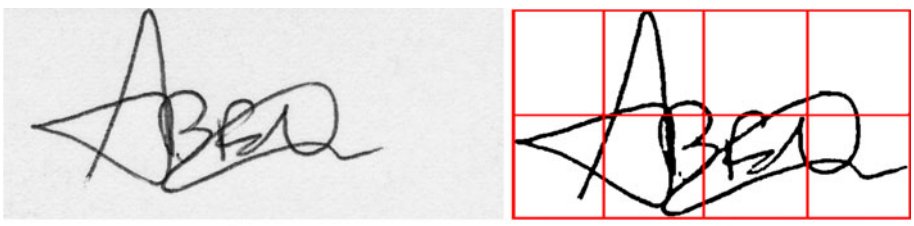
2.1 Preprocessing

The input signature image is preprocessed in two consecutive stages that are described below.

Binarization First, the image is binarized using a global histogram threshold algorithm. A morphological closing filter [14] is then applied in order to correct gaps and spurious points that may have appeared after thresholding.

From this moment on, the classical representation of a bitmap image as a matrix with the black pixels as the smallest portions of the object ($I[x, y] = 1$) and the white pixels as the background ($I[x, y] = 0$) is utilized.

Segmentation and partition Once we have the binary image, the next step is to locate the signature in the image and to extract the region of interest (ROI) as the signature bounding box. Then, this ROI is divided into a sub-structure of smaller regions. The foreground and background features described below are extracted from these areas of the ROI.



(a) (b)

Fig. 1 **a** Original image. **b** After being binarized, morphologically closed and divided into 4×2 regions

2.2 Foreground features

The subregions defined to represent the signature ROI are described by the number of foreground pixels that belong to each of them. The number of sub-regions along the X or Y axis of the image may not be a divisor of the number of pixels for the ROI in those directions, so limiting pixels can be shared proportionally among different regions. If this situation occurs, each region involved accumulates the proportional value of this pixel to its account, as displayed in the simple example in Fig. 2.

The bitmap image ROI is named $I[I_{XM}, I_{YM}]$, with $I_{XM} \times I_{YM}$ dimensions. And $R[R_{XM}, R_{YM}]$ will be the matrix with the proportional summatory of foreground pixels, with $R_{XM} \times R_{YM}$ representing the ROI subdivision size in X and Y. A discussion on the optimal number of ROI region subdivision will be presented in the results section.

A foreground feature matrix is defined, $\mathbf{F} \in \mathbb{Q}^{R_{XM} \times R_{YM}}$: $\mathbf{F} = f_{ij};\ i = 1 \ldots R_{XM},\ j = 1 \ldots R_{YM}$. Each value of this matrix, f_{ij}, represents the number of foreground pixels in the corresponding image subregion R_{ij}, as shown in the example in Fig. 3, based on the signature displayed in Fig. 1b.

2.3 Background features

The algorithm used to extract the background features is based on that of Vellasques et al. [17]. This algorithm computes four projections (up, down, left, and right) that are plotted for each pixel in the image. When any of these projections touch the foreground object, the counter associated with that pixel increases one unit. Note that these projections are considered until the ROI limits, not just to sub-region limits. In this way, we can distinguish four different categories of background pixels, according to the value of their counter. In addition, a fifth category is added in order to provide more information: there are two situations that are similar in geometry but totally different from a topological point of view. A background pixel can be surrounded by object pixels and then the projections will touch them and the number will be 4, but this pixel could belong either to an isolated region or to an open region. So, our algorithm assigns a value of 5 to the number if the pixel lies in an isolated background area.

Therefore, five matrices are extracted as features, one for each counter value. $\mathbf{B}^k \in \mathbb{Q}^{R_{XM} \times R_{YM}}$: $\mathbf{B}^k = b_{ij}^k;\ k = 1 \ldots 5, i = 1 \ldots R_{XM},\ j = 1 \ldots R_{YM}$ as background features. Each value of these matrices, b_{ij}, represents the number of pixels with a particular counter value in the corresponding image subregion R_{ij}, as shown in the example in Fig. 4.

For example, note how feature 2 gets the highest scores in most of the subregions because for this particular image most of the pixel projections hit foreground for two directions. In the case of the top left-hand region (R_{11}), the down and right projections of 224 pixels hit foreground.

2.4 Contour features

The object contour is encoded by the links between each pair of 8-neighbor pixels using 4-chain codes in the way proposed by Oda et al. [9], where only the orientations of the links are taken into account (see Fig. 5), so there are two directions for the 8-vertical codes (north and south) that are mapped to 1, etc.

Four matrices are extracted, one for each direction, denoted as $\mathbf{C}^k \in \mathbb{Q}^{R_{XM} \times R_{YM}}$: $\mathbf{C}^k = c_{ij}^k;\ k = 1 \ldots 4, i = 1 \ldots R_{XM},\ j = 1 \ldots R_{YM}$, as contour features. In a similar way to that done

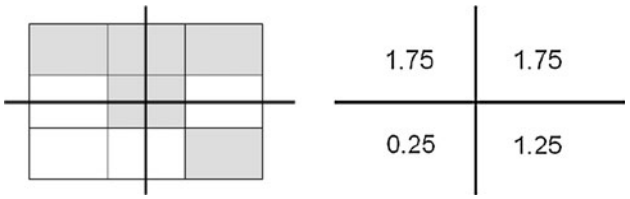


Fig. 2 Toy example to illustrate the proportional accumulative accounts for the foreground pixels from a 3×3 pixel ROI divided into 2×2 subregions. The shared pixel accumulate their proportional parts to the subregions they belong to

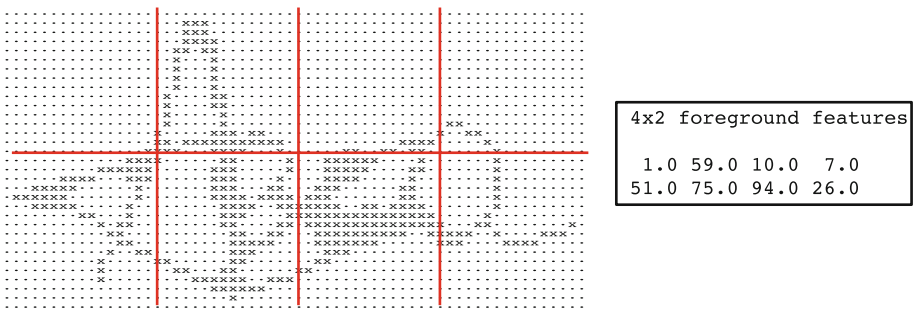
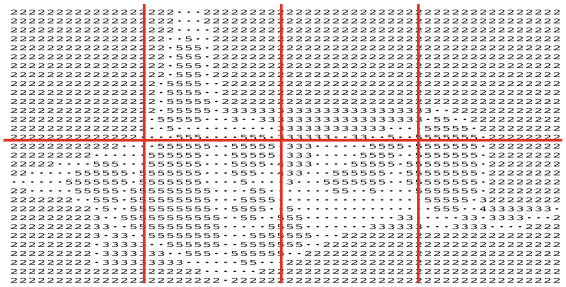


Fig. 3 Foreground features, **F**, for a 4×2 region division of the ROI containing the signature in Fig. 1b. The image has been undersampled for clarity



5x4x2 background features

1:	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0
2:	224.0	110.0	165.0	204.0
	130.0	23.0	78.0	151.0
3:	0.0	13.0	48.0	1.0
	21.0	6.0	20.0	18.0
4:	0.0	0.0	0.0	0.0
	0.0	2.0	0.0	1.0
5:	0.0	43.0	2.0	13.0
	38.0	134.0	48.0	44.0

Fig. 4 Arrays of background features, $\mathbf{B}^k$. Five arrays are computed for the 4×2 regions for the signature from Fig. 1b

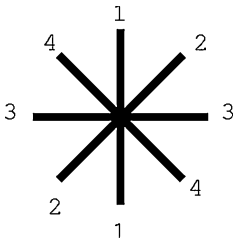


Fig. 5 Four 2D directions for contour chain codes using 8-neighbors, but only coding the four different orientations of the segments

for the previous sets of features, each cell of these matrices represents the summatory of a direction for each region, as shown in an example in Fig. 6.

All the features discussed in the previous sections are normalized using $p'_{ij} = p_{ij} / \sum p_{ij}$ to obtain better performance, where p_{ij} are the matrix values for a particular feature.

2.5 Vector of pixel erosion rates

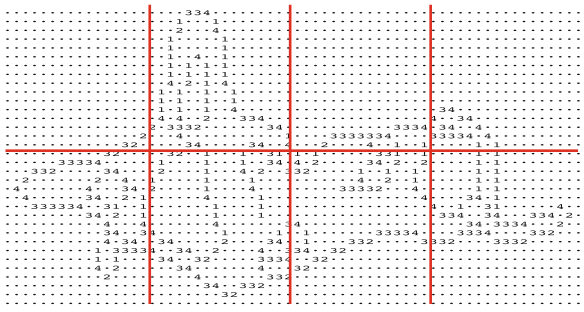
Morphological erosion of n pixels is applied to the binary image. At each step, the number of remaining foreground pixels after $erosion(i)$ is divided by the number of foreground pixels after $erosion(i-1)$. These ratios help to describe the thickness of the strokes. The algorithm is detailed below, where I is the image and N is the length of the final vector of features, $\mathbf{v} \in \mathbb{R}^N$:

```
function computeRatesVectorErosion(I, N)
  float v[N]
  for i := 0 to N − 1 do
      if I.erosion(i).countForegroundPixels()>0 then
        a:=I.erosion(i + 1).countForegroundPixels()
        b:=I.erosion(i).countForegroundPixels()
        v[i]:= a / b
      else
        v[i]:=0
      end if
   return v
 end function
```

When this procedure is applied to the example signature, the values for **v** displayed in the next table are obtained:

i	1	2	3	4	5
$v[i]$	0.2352	0.0340	0.0	0.0	0.0

where $N = 5$ has been fixed heuristically for this particular example. Actually, this maximum vector dimensionality has to be large enough for any foreground pixels to disappear in N or less erosions. That is, all vectors should be zero at the last positions of this feature vector. There is no risk of taking a very large N as the Euclidean distance is used to compare these vectors, so these null extra dimensions would cause just a small computation cost to the



4x4x2 contour features

1:	0.0	30.5	4.0	3.0
	7.5	16.0	8.0	12.0
2:	3.0	5.0	1.0	0.0
	8.0	9.0	10.0	5.0
3:	1.5	11.0	10.5	6.5
	23.5	20.0	18.5	20.5
4:	0.0	14.0	3.5	6.5
	17.0	16.0	8.5	8.5

Fig. 6 Contour features. 4 $\mathbf{C}^k$ 4×2 matrices with the summatories of the number of pixels for each direction from Fig. 1b

distance calculation, but contributing to it with zeroes. On the other hand, choosing a small N would cause an underrepresentation of this feature as very thick strokes need a high number of erosions to be deleted. Therefore, it is preferable in practice to chose a reasonable high value for N.

3 Verification system schemes

The verification systems utilized are based on classification schemes and were adapted to the verification task. Two kinds of verification algorithms were compared based on the different feature descriptions utilized. The Euclidean distance was used to compare the different vectors of features. After computing it, another distance function is proposed to implement the expert confidence, as explained below. In the first verification algorithm considered, only one expert is used to take the decision. These experts are nearest neighbor classifiers applied to each of the feature categories described above. In the second case, the 11 experts are used and combined to take the final decision. The novelty of this work resides in the use of an ensemble of different measure functions whose results take values in very different ranges. These two systems are explained in more detail as follows:

Individual verification system This system is based on the four sets of features described above. The 11 individual classifiers used are based on the image pixels (1 foreground classifier), the background information (5 background classifiers, one per matrix), the four directions of the chain contour codes (4 classifiers), and the rate vector from erosions (1 classifier based on the n-erosion vector).

All the decisions are taken in these classifiers under a nearest neighbor approach based on the Euclidean distance.

Ensemble verification system This one is based on combining decisions using the confidence voting methods with two rules [16]:

- Sum rule: Each class sums the confidence given by each individual decision system. The winning class (genuine or impostor) is the one that accumulates the highest summation score.
- Product rule: This works in the same way as the *sum rule*, but the product of the individual confidences is used to obtain the final score for each class.

The confidences for each of the individual verification systems are based on the Euclidean distance from the target sample to the training set prototypes. A revision of the expression to compute the a posteriori probability based on [15] and [2] applied to k-NN is used. In this case, the estimation involves computing the nearest neighbor prototype for each class and normalizing the inverse of the distance. So, the estimation is $\hat{P}(\omega_i|\mathbf{x})$ where ω_i is the i-th class (genuine or impostor) and $\mathbf{x}$ is the feature vector of a new example to verify. The new estimation is computed as:

$$\hat{P}(\omega_i|\mathbf{x}) = \frac{\frac{1}{\varepsilon + \min_{\mathbf{y}_i \in \omega_i}\{d(\mathbf{x},\mathbf{y}_i)\}}}{\sum_j \frac{1}{\varepsilon + \min_{\mathbf{y}_j \in \omega_j}\{d(\mathbf{x},\mathbf{y}_j)\}}}$$

This estimation is based on the nearest neighbor sample to genuine and impostor signature prototypes in the database. An ε value is introduced as a positive small value close to 0. This allows the formula to be computed without overflow errors.

4 Results

A subcorpus of the MCYT bimodal database [10] was used for the experiments. In the case of signatures, highly skilled forgeries were also available. Forgers were provided with the signature images of the clients to be forged and, after practicing with them several times, they are asked to imitate the shape with natural dynamics, i.e., without breaks or slowdowns. The resulting subcorpus comprises 2250 signature images (resolution 850×360 pixels), 75 different users with 15 genuine signatures, and 15 forgeries per user (contributed by 3 different user-specific forgers).

The signature subcorpus is divided into training and test sets in order to try out the system verification. It has been used in other studies such as [10] with the similar training/test structure and skilled/random forgeries.

There are two ways to choose training/test sets:

Skilled forgeries: There are 15 genuine/impostor examples per user, so $5 \times 75 \times 2 = 750$ examples were chosen for the training set (5 genuine and 5 impostor signatures) and the rest of examples, $10 \times 75 \times 2 = 1{,}500$, made up the test set. If 10 examples per user were chosen, then the training set had $10 \times 75 \times 2 = 1{,}500$ examples and the test set $5 \times 75 \times 2 = 750$. The signatures were chosen randomly.

Random forgeries: There are 15 genuine examples and 15×74 forgeries (the rest of the genuine signature users). The size of training/test sets was similar to that in the previous case, but the impostor signatures were selected from the rest of the users randomly.

In a verification system, four situations are possible: an impostor is accepted (false acceptance, FA), an impostor is rejected (true rejection, TR), a correct user is rejected

(false rejection, FR) and a correct user is accepted (true acceptance, TA). Performance measures of verification systems are related to the frequency with which the error situations occur. One common performance measure is the so called equal error rate (EER) which is the point attained when FA and FR rates coincide. Also a particular representation of the sensitivity versus specificity called receiver operating characteristic (ROC) curve is often utilized.

In this paper, the overall system performance when a posteriori user independent decision thresholds are used is reported by means of detection error tradeoff (DET) plots [8], which are graphical representations of FA versus FR rates with a particular axis scaling, and ROC curves. Average EER tables when using a posteriori user-dependent thresholds are also given following the operational procedure proposed in [7] for computing the individual EER for each user.

In the preliminary trials, different numbers of regions were tested. Since, the image aspect size was 2×1, the image region divisions tested were proportional to that ratio: 2×1, 4×2, 6×3, 8×4, 10×5, and 12×6, in order to establish the number of divisions for the rest of the experiments. A division of the ROI into 8×4 subregions yielded a lower EER than the early experiments on random forgeries with both the individual and ensemble decisions systems. Thus, the R_{XM} and R_{YM} parameters were fixed to 8 and 4, respectively, to perform the rest of the experiments.

Signature off-line verification performances for the proposed systems using individual and combined decisions are shown for a posterior user-dependent decision thresholding. Five cross-validation tests were performed for all the experiments, so averages and dispersions are presented in the tables. Also the sensitivity of the verification ensemble architecture was tested removing one of the eleven individual experts from the final combination, and the results were always worse than those obtained with a combination of all the experts.

Table 1 Performance on skilled forgeries for a posterior user-dependent decision thresholding

	5 Training examples	10 Training examples
foreground	21.1 ± 0.5	15.8 ± 0.8
background:1	43.6 ± 0.5	43.1 ± 1.2
background:2	24.2 ± 0.6	19.8 ± 0.5
background:3	23.2 ± 0.4	19.1 ± 1.5
background:4	30.0 ± 0.6	26.8 ± 1.0
background:5	33.2 ± 0.3	28.1 ± 0.7
directions:1	21.6 ± 0.2	17.6 ± 0.4
directions:2	19.2 ± 0.6	14.8 ± 0.9
directions:3	22.2 ± 0.4	17.7 ± 0.5
directions:4	21.0 ± 0.6	15.8 ± 0.4
15-erosion	26.9 ± 0.6	24.0 ± 0.5
combined sum rule F+B[1-5]+D[1-4]+E	10.9 ± 0.6	6.3 ± 0.4
combined pro rule F+B[1-5]+D[1-4]+E	11.0 ± 0.6	6.4 ± 0.3

Average EERs (%) ± dispersion (($\max_E - \min_E$)/4)

Table 1 presents the EER results on skilled forgeries and Fig. 7 displays the ROC curves for some of those experiments. From these results, we can see that the best performances for skilled forgeries were obtained by the combination of experts. The summatory rule outperformed the product rule, although there were no significant differences between them. This significance exists when the combined results are compared to the individual classifiers, for any of the features utilized.

Also, when assessed using the ROC, the combined rules (*fbde* in the graphs) clearly outperformed the rest. In these curves the quality is measured by the area under the curve, and the curves corresponding to the combined rules are clearly over the rest. No significant differences are observed between the curves for both combination rules.

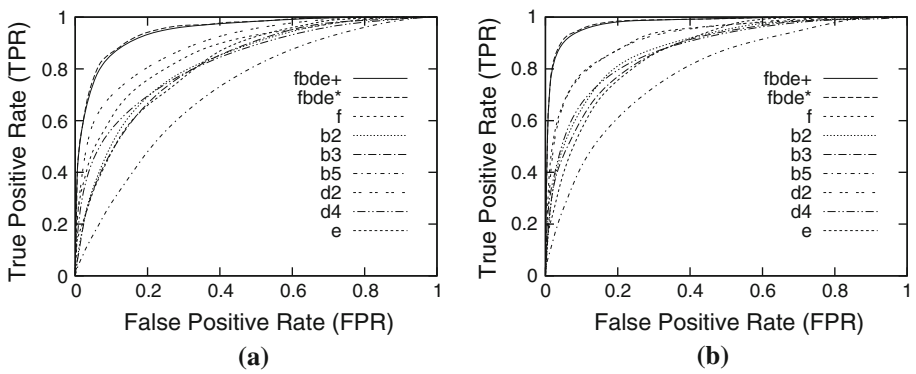


Fig. 7 ROC representation of skilled forgeries using a different number of training set examples. **a** 5 genuine and 5 false. **b** 10 genuine and 10 false

Table 2 System performance on random forgeries for a posterior user-dependent decision thresholding

	5 Training examples	10 Training examples
foreground	12.4 ± 0.7	8.3 ± 0.4
background:1	40.5 ± 1.0	39.2 ± 0.4
background:2	15.8 ± 0.6	11.6 ± 0.6
background:3	16.1 ± 0.7	11.1 ± 0.3
background:4	22.3 ± 1.3	17.7 ± 1.2
background:5	22.2 ± 0.2	16.8 ± 0.5
directions:1	12.8 ± 0.8	7.5 ± 0.5
directions:2	13.1 ± 0.8	7.5 ± 0.8
directions:3	12.9 ± 0.7	8.4 ± 0.7
directions:4	14.9 ± 0.7	8.8 ± 0.7
15-erosion	27.6 ± 0.8	21.1 ± 0.9
combined sum rule F+B[1-5]+D[1-4]+E	6.5 ± 0.5	2.31 ± 0.14
combined pro rule F+B[1-5]+D[1-4]+E	6.8 ± 0.5	2.35 ± 0.14

Average EERs (%) ± dispersion $((\max_E - \min_E)/4)$

Table 2 presents the EER results on random forgeries and Fig. 8 displays the ROC curves for some of those experiments. Note that the same comments on the skilled forgeries experiments can be applied to these results, with significant better performances for the combination of experts and no significant differences between both combination rules.

When 10 examples were used in the training set, all the systems improved their performances and lower EERs were obtained in all cases. Note that for random forgeries, the EERs obtained by the combined rules were roughly half of those yielded with 5 examples, and for skilled and forgeries the EERs were lowered to a third. They obtained an EER of 6.3 and 2.31 % as the best results in skilled and random forgeries, respectively. The dispersions obtained were much lower in this case for the combined rules, so these systems are also performing more consistently for the different partitions of the database.

These performances are comparable to those published in similar works using the same database [3], although in that work, the authors did not provide any deviation measure, so we can not evaluate the significance of the differences. In any case, the main objective of our work was to show how the combination of partial decisions using a function based on a posteriori probability measure improves both the performance and the robustness of the system.

5 Conclusions and future work

A number of methods for off-line signature shape feature extraction from a binary image have been described. Each set of features was used for an individual signature verification systems based on Euclidean metrics. In order to achieve a better performance, weighted voting-based ensembles of the verification system were constructed. An a posteriori probability estimation has been proposed in order to normalize the confidences provided for each verification system in the voting stage. This approach has proved to reduce the final EER significantly. Also consistent improvements have been found when assessing the system using ROC curves.

This expert combination approach is planned to be applied to different databases in the future to explore its robustness in other situations and how it performs when weighted decisions are used. In addition, we will explore its capabilities when dealing with other kind of data in the biometric recognition field.

Acknowledgments This work was partially supported by the Spanish CICYT under Spanish MICINN projects TIN2009-14205-CO4-01 and TIN2009-14247-C02-02, and by the Spanish research programme Consolider Ingenio 2010: MIPRCV (CSD2007-00018).

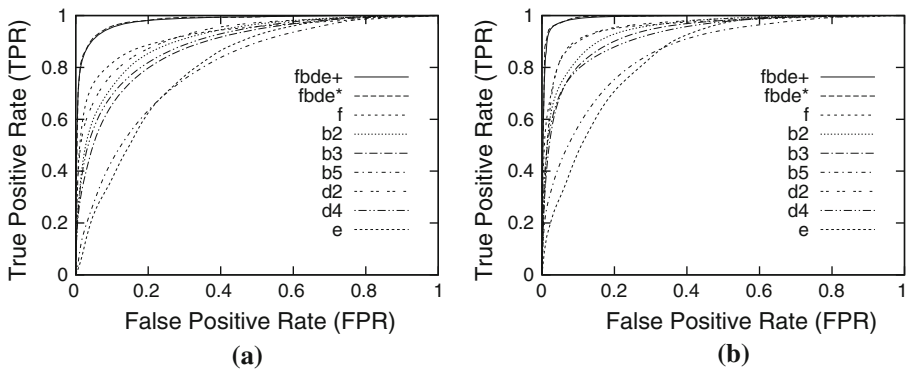


Fig. 8 ROC representation of random forgeries using a different number of training set examples. **a** 5 genuine and 5 false. **b** 10 genuine and 10 false

References

1. Abreu M, Fairhurst M (2010) Improving forgery detection in offline forensic signature processing. In: 3rd International conference on crime detection and prevention (ICDP 2009). IET, pp 1–6
2. Arlandis J, Perez-Cortes J, Cano J (2002) Rejection strategies and confidence measures for a k-nn classifier in an ocr task. In: 16th International conference on pattern recognition ICPR-2002, vol 1. IEEE Computer Society, Québec, pp 576–579
3. Fierrez-Aguilar J, Alonso-Hermira N, Moreno-Marquez G, Ortega-Garcia J (2004) An off-line signature verification system based on fusion of local and global information. In: Proceeding of European conference on computer vision. Workshop on biometric authentication, BIOAW. LNCS, vol 3087. Springer, Berlin, pp 295–306
4. Impedovo D, Pirlo G (2008) Automatic signature verification: the state of the art. IEEE Trans Syst Man Cybern Part C: Appl Rev 38(5):609–635
5. Jain AK, Griess FD, Connell SD (2002) On-line signature verification. Pattern Recognit 35:2963–2972
6. Kuncheva LI (2004) Combining pattern classifiers: methods and algorithms. Wiley, New York
7. Maio D, Maltoni D, Cappelli R, Wayman J, Jain A (2002) Fvc 2000: fingerprint verification competition. IEEE Trans Pattern Anal Mach Intell 24(3):402–412. doi:10.1109/34.990140
8. Martin A, Doddington G, Kamm T, Ordowski M, Przybocki M (1997) The det curve in assessment of detection task performance. In: ESCA European conference on speech communication and technology, EuroSpeech, pp 1895–1898
9. Oda H, Zhu B, Tokuno J, Onuma M, Kitadai A, Nakagawa M (2006) A compact on-line and off-line combined recognizer. In: Tenth international workshop on frontiers in handwriting recogntion, vol 1, pp 133–138
10. Ortega-Garcia J, Fierrez-Aguilar J, Simon D, Gonzalez J, MF, Espinosa V, Satue A, Hernaez I, Igarza JJ, Vivaracho C, Escudero D, Moro QI (2003) Mcyt baseline corpus: a bimodal biometric database. IEE Proc Vis Image Signal Process (Special Issue on Biometrics on the Internet) 150(6):395–401
11. Plamondon R (1994) The design of an on-line signature verification system: from theory to practice. IJPRAI 8(3):795–811
12. Rico-Juan JR, Iñesta JM (2007) Normalisation of confidence voting methods applied to a fast and written OCR classification. In: Kurzynski M, Puchala E, Wozniak M, Zolnierek A (eds) Computer recognition systems 2. Advances in soft computing, vol 45. Springer, Wroclaw, pp 405–412
13. Ruiz-Del-Solar J, Devia C, Loncomilla P, Concha F (2008) Offline signature verification using local interest points and descriptors. In: Progress in pattern recognition, image analysis and applications, pp 22–29
14. Serra J (1982) Image analysis and mathematical morphology. Academic Press, New York
15. van Breukelen M, Duin RPW, DT, den Hartog J (1998) Handwritten digit recognition by combined classifiers. Comput Linguist 34(4):381–386
16. van Erp M, Vuurpijl L, Schomaker L (2002) An overview and comparison of voting methods for pattern recognition. In: IWFHR '02: proceedings of the eighth international workshop on frontiers in handwriting recognition (IWFHR'02). IEEE Computer Society, Washington, p 195
17. Vellasques E, Oliveira L, Jr AB, Koerich A, Sabourin R (2006) Modeling segmentation cuts using support vector machines. In: Tenth international workshop on frontiers in handwriting recogntion, vol 1, pp 41–46
18. Yoshiki M, Mitsu Y, Hidetoshi M, Isao Y (2002) An off-line signature verification system using an extracted displacement function. Pattern Recognit Lett 23(13):1569–1577. doi:10.1016/S0167-8655(02)00121–6