A static signature verification system based on a cooperating neural networks architecture

Hubert Cardot, Marinette Revenu, B. Victorri, M.-J. Revillet

1994

A STATIC SIGNATURE VERIFICATION SYSTEM BASED ON A COOPERATING NEURAL NETWORKS ARCHITECTURE

HUBERT CARDOT and MARINETTE REVENU
LAIAC (Laboratoire d'Algorithmique et d'Intelligence Artificielle de Caen)
ISMRA, 6 bd. du Maréchal Juin, 14050 Caen, Cedex, France
E-mail : hcardot@L2i.ismra.fr

BERNARD VICTORRI
ELSAP, Université de Caen, 14032 Caen, Cedex, France

MARIE-JOSÈPHE REVILLET
SEPT, 42 rue des Coutures, 14000 Caen, France

We are applying neural networks to the problem of handwritten signature verification. Our system is working on checks, so we can only use the static information (the image). This static information is used in three representations: geometrical parameters, outline and image. Our system is composed of several neural networks which cooperate together during the learning and decision phases. The performances in generalization, obtained with a large-scale database of 6000 signatures from real checks on random forgeries, are False Acceptance Rate (FAR) = 2% and False Rejection Rate (FRR) = 4%.

Keywords: Handwritten signature, verification, static information, neural networks.

1. INTRODUCTION

1.1. Presentation of the Problem

Individuals are frequently asked to prove their identity when writing official documents. This is done to stop them from using someone else's signature and also to stop them from disowning a document that they have previously acknowledged. Texts are often typed, so it is not possible to verify these documents from the handwriting text. However, it is customary to append a mark verifying the author of the document, thus showing that he agrees with the text of the document. Nowadays this mark is generally a handwritten signature, so it would be useful to devise an automatic and reliable system for the verification of handwritten signatures appended on the numerous documents which are produced daily.

A signature verification system would have a use in several applications, we will focus on the verification of checks from the French Post Office.

Our goal is to detect rough forgeries, which are signatures written by someone who is not imitating a genuine signature. Those rough forgeries are the most commonly found forgeries. Systems based on dynamic information (duration, speed of the signing, ...) are able to detect good imitations. In our application however, this dynamic information is lost because the image of the check contains only static information.

International Journal of Pattern Recognition and Artificial Intelligence Vol. 8 No. 3 (1994) 679–692

1.2. Use of Neural Networks

Signature verification achieved by human experts is a difficult task to model. To solve this problem, Neural Networks (NNs) seem more appropriate than symbolic methods:

- We think that the learning and the generalisation abilities of NNs would be helpful to cope with the diversity and the variations of signatures.
- Once this learning is achieved (it can be done off-line), the response of a NN to an input is extremely fast which is interesting because it is during the exploitation phase (treating a flow of checks) that rapidity is necessary.
- It is possible to compare two images with NNs, a procedure which is difficult and long with classical methods.
- Although it is not presently included in our system, it is possible to follow the evolution of the signatures in time by regularly repeating the learning of the NNs with more recent signatures.

2. HANDWRITTEN SIGNATURE VERIFICATION

2.1. Handwritten Signatures

Two types of signatures are distinguished:

- An American type, which are cursive signatures. In this case, verification systems can use the presence of characters to help eliminating rough forgeries.
- A European type, which are graphical signatures that must be processed globally.

Two kinds of information can be extracted from signatures:

- Static information, generally acquired with a CCD camera, constitutes the image of the signature. The acquisition can be done off-line, that is to say after the appending of the signature.
- Dynamic information, acquired by a digitalisation device, contains the speed of the signing, its duration, its acceleration, and the lifting of the pen (tip of the pen no longer touching the support). It needs an on-line acquisition device while the writer is signing.

We can also define pseudodynamic information corresponding to dynamic information obtained from static information: fluctuations of the line width give an indication about the pressure exerted on the pen.

Our system works on European type signatures and only the static information is used.

2.2. Signature Verification Systems

2.2.1. General view

Signature verification systems are usually divided into two modules (Fig. 1):

- an acquisition module
- a verification module

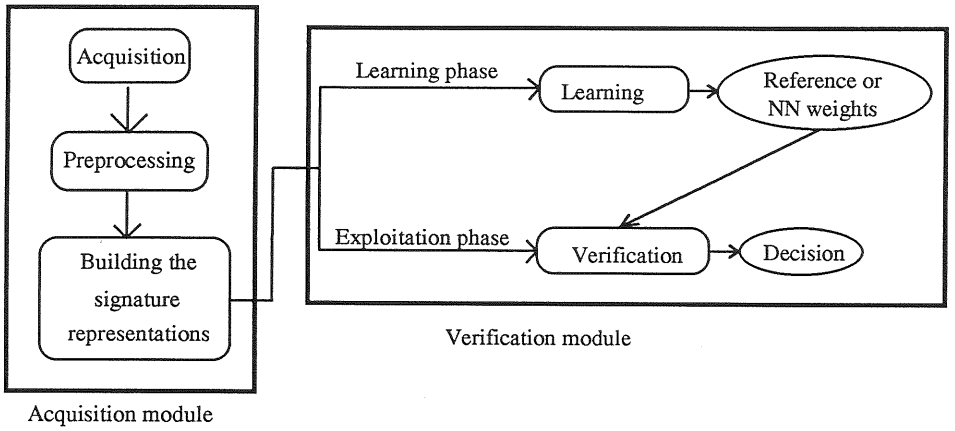


Fig. 1. A signature verification system.

In systems based on statistical methods, the learning phase consists of building the reference associated to each signatory. This reference can be several reference signatures or an average signature calculated from the reference signatures.

In the case of neural methods, the learning phase consists of showing signatures to the NNs and of modifying its weights according to its response. Thus, the reference is "contained" in the weights of the network, which is all the data we have to store. This gives an advantage to our method because the amount of data about each signatory does not increase according to the number of reference signatures used.

During the exploitation phase of systems based on statistical methods, the presented signature is compared with the reference of the supposed signatory. Then, the system has to make a decision either to accept or to reject the signature according to the result of this comparison.

In systems based on NNs, the decision is taken according to the response of the NNs to the presented signature.

2.2.2. Global view of our verification system

Our verification system takes up the different elements of the general verification system (see Sec. 1.2.1). Figure 2 gives an overview of our system.

2.3. Signature Representation

The image of a signature represents an important amount of data which is difficult to process globally. That is the reason why it is common, as in other image interpretation systems, to extract parameters that should be informative, discriminating and stable for each signatory.[1] In our system, we have used three representations of the signatures which will now be described.

We have used the pixel image itself as we felt this would be better than using a more processed image because the processing, which aims at extracting significant information, also removes some of the useful information. However, we cannot use

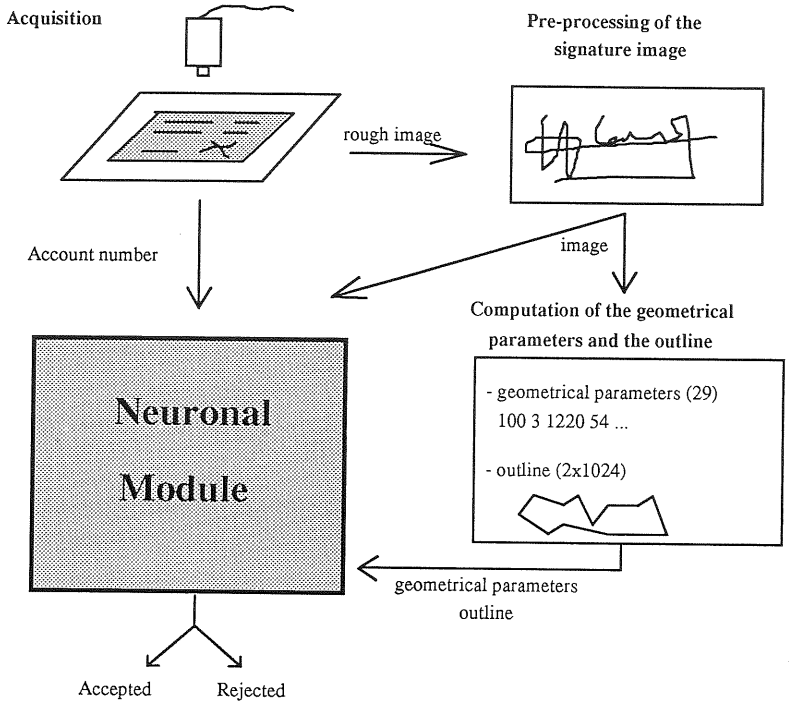


Fig. 2. Our verification system.

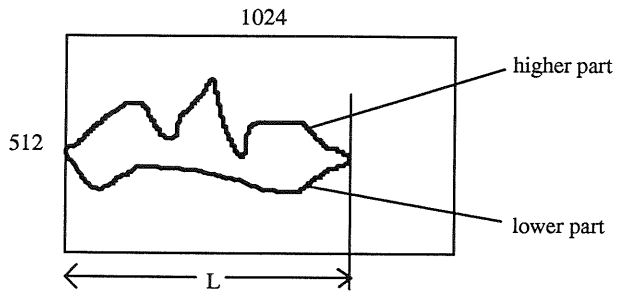


Fig. 3. Outline of a signature.

the image without performing some pre-processing to reduce the amount of data, which is too large to be processed straightaway. For example, our signature images are coded on 1024 × 512 pixels on 256 grey levels.

The outline of the signature (Fig. 3), that corresponds to the extreme vertical points, is interesting because, although it is made of much less data compared to the whole image, it keeps the global appearance of the signature ignoring local details.

It cannot be used on its own because it is not informative for some signatures (Fig. 4). Moreover, the outline is quite sensitive to the orientation of the signature. In our system, we use the outline only for a first rough comparison.

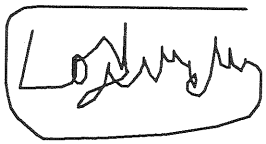


Fig. 4. Signature with a non-informative outline.

Geometrical parameters can be extracted either from the image or from the outline of the signature. These parameters are interesting because they constitute a more concentrated piece of information making them easier to manipulate and compare. Moreover, some geometrical parameters are invariant to rotation and magnification, which is particularly convenient. The main difficulty is to choose them so that they contain enough discriminative information and so that they remain stable in spite of local or global distortions.

A significant study has been done by the SEPT (French Telecom Research Centre)[2] to select twenty nine geometrical parameters, which include among others: size of the signature, orientation of the strokes and inertia moments. These parameters constitute, in our system, the most discriminant features for signature representation.

Getting inspiration from handwritten character recognition, we might have also considered using the number and the position of loops and intersections. But, for the European signature type, these characteristics are not stable. Frequently a stroke underlines or crosses the signature and the position of this stroke completely modifies the number of loops and intersections.

2.4. The Database

Our system is a model of a real-scale system which would be able to work on 300 000 people. In order to develop and test our neuronal architecture, a large representative database was needed and to constitute this database, 6000 checks were digitalized.

The image of the checks is binarized by an automatic and adaptive thresholding method. For our application, the right lower quarter of the image, which contains the signature, is extracted. The signature is often surcharged because banks apply stamps on the checks, the machines that automatically fill in the check amount and the date do it sometimes on the signature, and some signatures are written on the surrounding inscriptions (date, address, numbers).

To reduce these alterations, a processing is applied to the image. For every check, the position of every character line of the surrounding inscriptions is computed by scanning all the lines of the image, from the left. From the position of the first

character of each line, we compute the positions of the others, the size of all these being the same, except 1 or 2 pixels. These positions are then fitted to the real positions of the characters. In the same processing, we detect if the signature is between two characters or on some of them. The right placed characters without signature are suppressed when it is possible to reconstruct the signature. If reconstruction is impossible, the character is mixed to the signature: that modification of the signature is less important than if the rectangle is just suppressed without reconstruction.

A rotation is applied to make the axis of inertia of the signature horizontal, and a window is computed around the signature.

When studying the signature images after these operations, one will realize that this automatic pre-processing is not flawless: there are problems with some signatures which contain parasitic information that disturb the computation of the window around the signature and the rotation operation. This operation entailed a lot of work but still it could be improved.

3. DESCRIPTION OF THE VERIFICATION MODULE

3.1. Architecture of the Verification Module

3.1.1. Structure of the verification module

Our system is made up of three levels (Fig. 5). The first one is formed by two NNs of the non-supervised Kohonen map type. These two NNs use as input, geometrical parameters and outline respectively. Their function is to classify signatures belonging to the signatories of the same set (for instance: a postal check centre) into signature classes. The second level is also formed of two NNs, that are multilayer networks using an error gradient backpropagation learning algorithm. One uses as input geometrical parameters and the other the image of the signature. These two NNs are specific to each signatory. The third level is formed only by one NN of the backpropagation type. It takes as input the outputs of the previous NNs and makes the final decision about whether to accept or reject the signature under examination.

The links represent the cooperation existing between the NNs. They are all used during the learning phase but only the links connected to the decision NN are also used during the exploitation phase.

3.1.2. Building the training database

Some examples of genuine signatures and forgeries must be shown to the two NNs on the second level during the learning phase. As one NN is associated to each signatory, we can use the reference signatures for the genuine signatures. But, for the forgeries we have to find some representative examples among the signatures of other people. Several solutions to this problem could be considered:

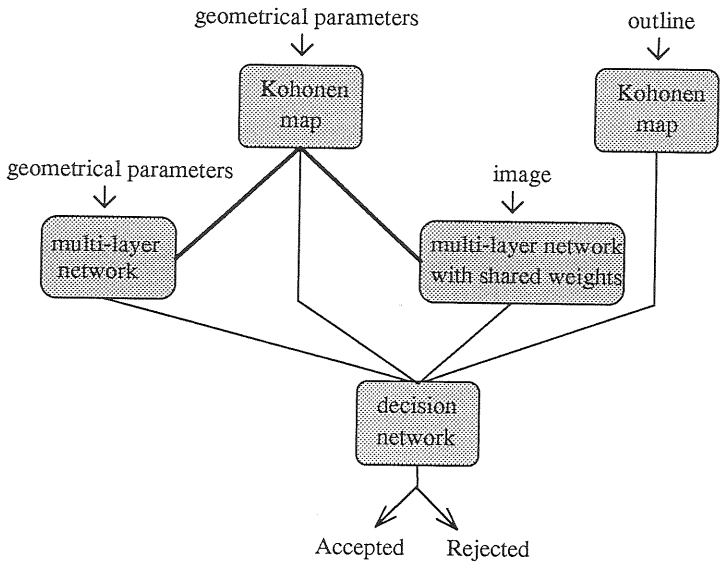


Fig. 5. The three levels of the verification module.

– The first consists in taking a few signatures from all the other people in the database. However, this would be unsatisfactory because the learning phase becomes too long when dealing with a real-scale database. Moreover, it would be necessary to go over again the learning for all the signatories when a new signatory is added to the database.

– The second possible solution involves presenting random values to the NNs instead of the geometrical parameters of false signatures; we have tested this solution because it simplifies the test protocols as the learning can be done independently from the rest of the database. But it gives poor results because NNs have to learn to differentiate the genuine signatures from everything within the representation space, the dimension of which equals the number of inputs to the NN. In reality, the representation space is much larger than the area where the genuine signatures can be found, although they can be very different from one another and it is important to limit the inputs of the NNs to the existing signatures only.

– In the third solution, we take a sample of genuine signatures from other people to represent the false signatures. These false signatures are chosen randomly among the reference signatures of the other signatories. Results are better than with the second solution but we have noticed some differences when the random generator was initialised with different values. Globally, on all signatories, the results were about the same, but when analyzing more precisely the results for a given person, significant variations can be noticed. We concluded that the choice of false signatures had some influence on the learning of the NNs, and that a random choice, although being a good solution, is certainly not the best one.

The question we have to answer is: which false signatures should be learned, for a given person, in order to optimize the learning phase? We noticed an improvement in the results when we reduced the space of signatures from randomly-made signatures to real ones. So we continued in this way and tried to reduce the space of signatures used for learning: the idea was to present to the NNs, false signatures that resembled the genuine ones of a given person. Figure 6 summarizes how the representations space was gradually reduced.

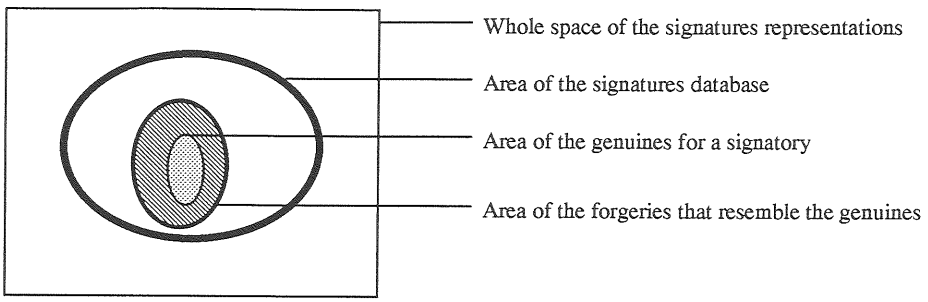


Fig. 6. Reducing the size of the representation space.

To determine whether a false signature is close to a genuine signature of a given person, we use a Kohonen map; it classifies signatures into a predefined number of classes. The false signatures, used in the learning of the NN associated to each signatory, are chosen among the false signatures that have been classified in the same class as most of the genuine signatures of that person.

3.1.3. Cooperation within the verification module

Two types of cooperation[3] are used in our verification module:

The first one, of the "modulation" type, establishes cooperation between NNs of the first level and NNs of the second level. The former performs a rough classification of the signatures of the learning base, in order to improve the learning of the latter.

The second type of cooperation of the so-called "associative" type, is achieved by a third level which takes as its inputs the outputs of the NNs from the two first levels, and makes the decision whether to accept or reject the signature. Details about this NN are given in Sec. 3.4.

3.2. The First Level of Our Verification Module

The goal of this layer is to perform an initial classification, using the geometrical parameters and the outline of the signature. We build one NN for the parameters, and one for the outline. Both NNs work on the whole set of signatories, as we aim to classify each signature in comparison with all the others.

As the number of signatories can be very large, it is impossible to have as many classes as signatories, which would have enabled supervised learning. So, we have to limit the number of classes of each NN. In the case of a real-scale application, this number of classes would be much smaller than the number of signatories and we would notice that several people share the same class and that a signatory can be classified in more than one class.

We cannot make a prior forecast about which signatories are to be found in the same class. That is the reason why it is difficult to use a supervised learning method, in which we would tell the NN, for each reference signature, which class it is to choose. So, we decided on a type of NN enabling non-supervised learning. In this case, during the learning phase, the NN has to group all the signatures it judges quite similar to one another into the same class.

To implement this scheme, we chose the most current type of NN that enables unsupervised learning: the Kohonen self-organising network,[4] which will now be described. In our system, the Kohonen NN has as many input cells as the number of components of the data vector, i.e. 29 for geometrical parameters and 400 for the outline (400 equals twice the width of the outline). The output layer has two dimensions of the same size. The number of output cells is chosen experimentally below 50 (49, 36 or 25). This number of output cells corresponds to the number of classes in which the NN will classify the signatures. The input cells are all connected to the output cells.

The way we use those NNs can be decomposed in three stages:

- First, the learning of the weights with the reference signatures.
- Second, the determination of the associated class for each signatory. It is the most frequent class given by the NN when each reference signature of a signatory is presented to it.
- Third, the verification stage. The signature is presented to the NN, if it is classified in the associated class of the signatory the activation value of the corresponding cell is transmitted to the decision NN otherwise 0 is transmitted to it.

As we have seen before, the classification done in this level helps with the choice of the forgeries used for the training of the NNs on the second level.

3.3. The Second Level of Our Verification Module

3.3.1. Multilayer network working on the geometrical parameters

This network has an input layer of 29 cells corresponding to the 29 geometrical parameters, an output layer with a unique cell and a variable number of hidden layers. The learning phase is done using the error gradient backpropagation algorithm.[5] Cells of one layer are completely connected to the cells of the next layer. The transfer function is a sigmoid.

The final structure of this NN was decided, following the results obtained when using it independently from the other NNs of the verification module. We achieved the best results when the multilayer network possessed no hidden layer. This may

seem surprising when you know that hidden layers enable NNs to delimit classes more precisely. In reality, hidden layers improve rote learning of the signatures but diminish generalization capabilities. Those generalization capabilities are particularly interesting as they enable the NNs to give correct answers, even when they have never learnt the input signature.

3.3.2. Multilayer shared weights NN (MSWNN) working on the image

In traditional multilayer NNs, all the neurons of a layer are completely connected to the neurons of the next layer and associated with each connection is a weight. In the case of multilayer shared weights NNs,[6] we only keep local connections to a neuron (Fig. 7) and we use shared weights, i.e. weights common to several connections.

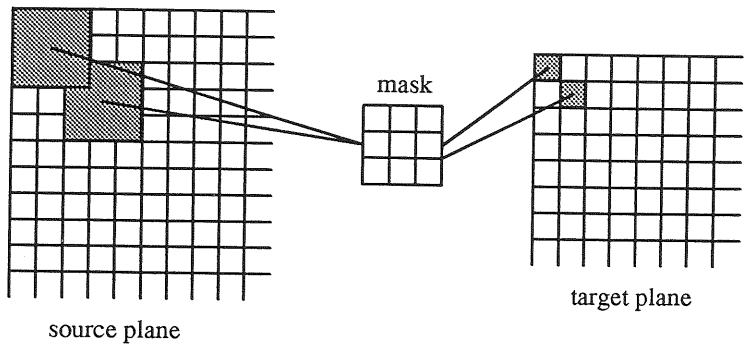


Fig. 7. Principle of shared weights.

To increase the NN efficiency, we use several planes for each layer (Fig. 8). Each plane of a layer "sees" the same data and thus does the same work. Each plane has a mask, whose initial weight values are chosen randomly at the beginning of the learning phase. As the initial mask values are different, each plane converges towards a different "view" of the data, thus extracting different features.

Figure 8 shows the final structure of our MSWNN which contains 2 planes for the first hidden layer, i.e. 2 masks 3×3, and 5 planes for the second layer, i.e. 5 masks 5×5.

3.4. The Third Level of Our Verification Module

In our first experimentation,[7] the decision phase only consisted of merging the boolean (true – false) outputs of each NN. Thanks to the size of our database (6000 signatures), we could then improve our decision strategy by introducing a multilayer NN, whose output was any number between 0 and 1, proportional to the similarity between the tested signature and reference signatures. This achieves the same result as asking an expert to give his advice. His answer may take the form, "I think these signature do not resemble each other". For the final decision,

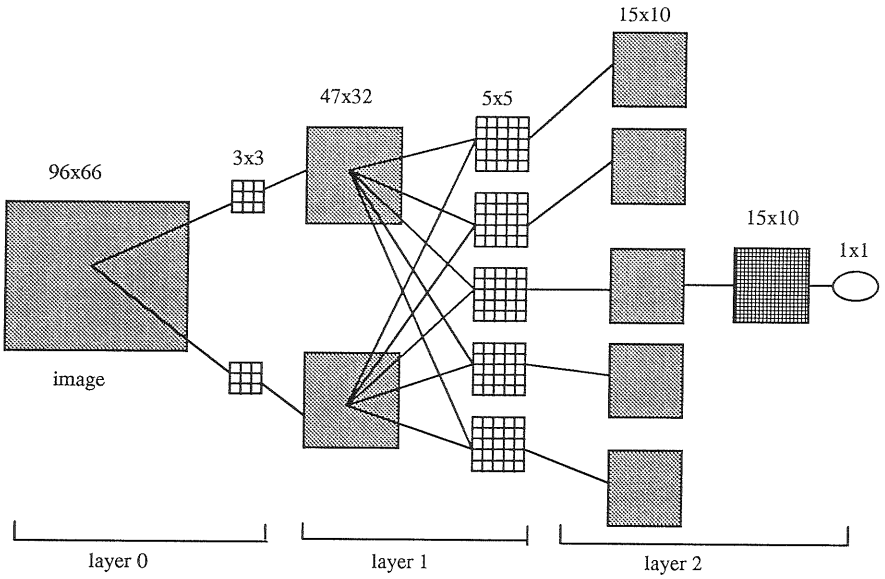


Fig. 8. Architecture of our MSWNN.

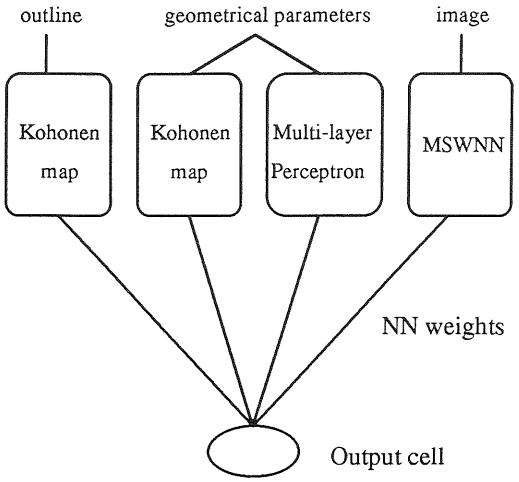


Fig. 9. The decision NN.

the answers of experts are weighted by a "confidence coefficient", corresponding to the fact that the answers of some experts play a more important part in the final decision than the answers of others.

Figure 9 shows the fifth NN, which takes the final decision about whether to accept or reject the signature. The weights of this NN are the same for all signatories. The NN takes as its inputs the outputs of the four former NNs and has only one output cell, the value of which is compared to a threshold, in order to make the decision.

4. RESULTS

To evaluate the performances of a verification system, two rates are generally computed: the false rejection rate (FRR) and the false acceptance rate (FAR). Those verification systems use a parameter, called the decision threshold in order to modify the hypersurface. Practically, with most of the representations used, classes cannot be separated. Then the choice of the decision threshold follows one of the following criteria:

- minimizing the average of FAR and FRR.
- keeping one of the two rates below a desired rate (for instance FAR lower than 1%).

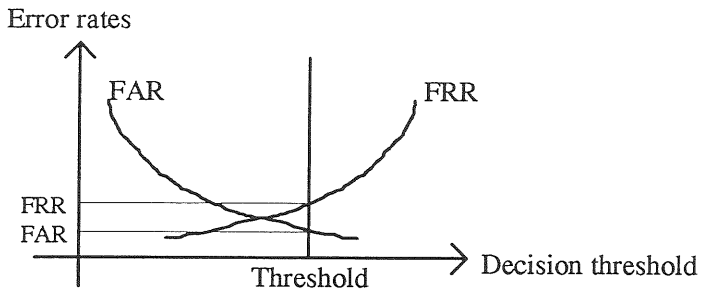


Fig. 10. Choice of a decision threshold based on a criterion.

In order to test our system, we have used a database of 6000 signatures extracted from real checks done by over 300 signatories. This database has been divided into two parts: the training database and the testing database. In real-scale functioning, it is not the learned signatures that will be verified but new ones, so it is important to strictly separate the two parts of the database. In our system, the training database is composed of 6 genuine signatures and 30 random forgeries for each signatory, the rest of the database constitutes the testing database. The database does not include imitations.

The performance in generalization are summed up in the following table (Fig. 11). The first row gives the FRR corresponding to a FAR of 2% and the second row the FRR corresponding to a FAR below 1%.

FAR	FRR
2.0 %	4.0 %
0.9 %	7.4 %

Fig. 11. Results of the complete system.

5. CONCLUSION

We have devised an architecture that enables the cooperation between various NNs using different representations of handwritten signatures namely geometrical parameters, outline and image of the signature. Part of this architecture is specific to each signatory and another part global to the set of signatories. Two types of cooperation are brought into play, fusion of results obtained when working on the different representations of signatures, and improvement of the learning phase of multi-layer NNs. This improvement in the learning phase is thanks to the selection of sample forgeries that more closely resemble genuine signatures than randomly selected ones and this is in turn thanks to a first classification performed by non-supervised Kohonen NNs.

Our research is directed towards an industrial application that can deal with millions of signatures from hundreds of thousands of signatories. This would imply strong constraints upon the size of the data to be stored for each signatory and as our present system only needs about 300 values for each signatory, this would be acceptable for a large-scale application.

Our results can be compared with others,[8] taking into account that our application has been developed and tested on a large-scale database. This database was built by the digitalization of 6000 postal checks from more than 300 signatories. The use of real data and a large database validates our system under almost normal conditions.

Finally, it will be possible to combine our verification module with the module developed by the SEPT which is based on statistical methods, because the decision level of our system can easily take into account results from other sources. As the methods used by the SEPT are very different from our approach, we should improve the global performance of the system, although we use the same representation of signatures.

ACKNOWLEDGEMENTS

The study presented in this article was achieved in the ISMRA research team of the LAIAC (Laboratory of Algorithmics and Artificial Intelligence of Caen). This work was performed thanks to a research contract between the ISMRA and the SEPT (France Telecom Research centre). It continues the thesis work done by F. Nouboud[9] in 1988. Special thanks to Christine Porquet and Rachel Cowen who helped with the English translation.

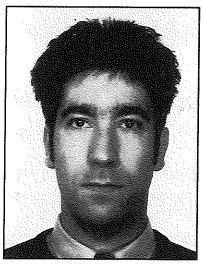
REFERENCES

1. R. Sabourin, "An approach oriented scene comprehension applied to the automatic verification problem of the identity using handwritten signature image", *Thèse de l'Université de Montréal*, Sept. 1990.
2. M.-J. Revillet, "Verification of postal cheques", ICDAR 91, Saint-Malo, Sept.–Oct. 1991.

3. L. Boutkhil, H. Cardot, F. Joublin, V. Lorquet, J.-D. Muller, O. Sarzeaud, and S. Wacquant, "Cooperation of neural structures and algorithms for the resolution of complex problems", *Int. Conf. Neuro-Nîmes*, 1992.
4. T. Kohonen, "Self-Organization and Associative Memory", Springer Series in Information Sciences, vol. 8, Springer Verlag, 1984.
5. D. Rumelhart, G. Hinton, and R. Williams, "Learning representations by backpropagating errors", *Nature* **323**, 9 (1986) 533–536.
6. Y. Le Cun, "Generalization and network design strategies", *Connectionism in Perspective*, eds. R. Pfeifer *et al.*, Elsevier Science Publ., 1989.
7. H. Cardot, M. Revenu, B. Victorri, and M.-J. Revillet, "Coopération de réseaux neuronaux pour l'authentification de signatures manuscrites", *Int. Conf. Neuro-Nîmes*, 1991.
8. R. Plamondon and G. Lorette, "Automatic signature verification and writer identification — The state of the art", *Pattern Recogn.* **22** (1989) 107–131.
9. F. Nouboud, "Contribution to the study and adjustment of a handwritten signature verification system", *Thèse de l'Université de Caen*, 1988.

Received 17 May 1993; revised 26 October 1993.

M. Revenu received the Diploma of electronic engineering in 1969, the DEA (Diplôme d'Etudes Appronfondies) and the Doctoral Thesis Degree (Ph.D.) in computer sciences from the University of Paris, VI, France, in 1982 and 1985, respectively. Since 1978, she has been teaching computer science and artificial intelligence at the Engineering School of Caen, as Assistant Professor. She is interested in applying artificial intelligence techniques, such as knowledge representation and neural networks, to computer vision and more precisely to image segmentation. She is in charge of a research group of eight persons and she supervises two students working in industry for their doctoral thesis degree.



H. Cardot received the Diploma of electronic engineering in 1988, the DEA (Diplôme d'Etudes Appronfondies) and the Doctoral Thesis Degree (Ph.D.) in computer sciences from the University of Caen, France, in 1988 and 1993, respectively. He is currently teaching computer science at the university of Caen as Assistant Professor. His research interests include pattern recognition and neural networks.