A NEURAL NETWORK APPROACH TO OFF-LINE SIGNATURE VERIFICATION USING DIRECTIONAL PDF

J.-P. DROUHARD, R. SABOURIN and M. GODBOUT
Laboratoire d'Imagerie, de Vision et d'Intelligence Artificielle (LIVIA), Département de génie de la production automatisée, Ecole de technologie supérieure, 4750 Henri-Julien, Montréal (Québec), Canada H2T 2C8

(Received 24 February 1994; revised 11 April 1995; received for publication 3 July 1995)

Abstract—A neural network approach is proposed to build the first stage of an Automatic Handwritten Signature Verification System. The directional Probability Density Function was used as a global shape factor and its discriminating power was enhanced by reducing its cardinality via filtering. Various experimental protocols were used to implement the backpropagation network (BPN) classifier. A comparison, on the same database and with the same decision rule, shows that the BPN classifier is clearly better than the threshold classifier and compares favourably with the k-Nearest-Neighbour classifier.

Pattern recognition Classifiers Neural networks Backpropagation
Automatic signature verification Directional probability density function

1. INTRODUCTION

The design of a complete Automatic Handwritten Signature Verification System (AHSVS) that will be able to take into account all classes of forgeries is a very difficult task. Indeed, a complete AHSVS should be able to discriminate between genuine signatures and the following forgeries: random forgeries, characterized by a different semantic meaning and consequently by a different overall shape when compared to genuine signatures; simple forgeries, with the same semantic meaning as genuine signatures but an overall shape that differs greatly; freehand and simulated forgeries, produced with the a priori knowledge of both the semantic meaning and the graphical model of a target signature by a skilled or an occasional forger respectively; finally, tracing forgeries and photocopies, with almost the same graphical aspect as genuine signatures, but with different pseudo-dynamic properties such as dissimilarities in grey-level-related features like texture, contrast.

In such a system, in order to take into account all classes of forgeries, the decision is made only at the end of the verification process. Consequently, this approach is a very costly solution in terms of computational resources and in terms of related algorithmic complexity. Since random and simple forgeries represent almost 95% of the cases generally encountered in practice, a better solution might be to subdivide the verification process in such a way to rapidly eliminate gross forgeries. Thus, a two-stage AHSVS seems to be a more practical solution, where the first stage would be responsible for this rapid elimination and the second stage used only in complicated cases. The design of this first stage was made with random forgeries based on the fact that a verification system able to cope with random forgeries will be able to successfully discriminate simple forgeries.

The first stage of this complete AHSVS thus has two main objectives: firstly, to consider only random and simple forgeries and, secondly, to make a rapid decision. To meet the first objective, a characteristic dealing with the overall shape of handwritten signatures seems appropriate. Accordingly, we have chosen to use the directional Probability Density Function (PDF) as a global shape factor. Its discriminating power is not optimum because, even though it is invariant in translation and in scale, it is not invariant in rotation. On the other hand, it does not require too much computer time, thereby satisfying the second objective.

To meet the second objective, we have chosen to use a BackPropagation Network (BPN) as a signature classifier. Indeed, once trained, unlike conventional classifiers such as the k Nearest Neighbour (kNN) classifier, it has a very fast response time since it does not have to memorize in full all signature specimens. However, the learning phase of these classifiers is a relatively difficult task in this application. As a matter of fact, with this type of classifier, we must know a priori all the false signatures and have many examples prior to training. In addition, due to the very high variability of handwritten signatures, the separation between true and false signatures is not a sharp one. This results in very long convergence times and the overall performance will never be perfect. Nevertheless, as shown in a feasibility study, these difficulties can be diminished if some precautions are taken and the results obtained correspond to the main
objective of the first stage of the decision, which is the rapid elimination of gross forgeries.

This paper addresses the problems related to the design of the first stage of a complete AHSVS. In Section 2, we give all the definitions that are used throughout this paper. The choice of the pre-treatment performed on the full directional PDF in order to enhance its performances is briefly presented in Section 3. The design of the BPN used as a signature classifier is summarized in Section 4. Finally, in Section 5 the BPN classifier performances are compared, on the same database, to those obtained with the conventional kNN and T classifiers.

2. DEFINITIONS

2.1. Data sets

A standard signature database of 40 signatures written by 20 individuals (800 images) is used in this study. For the choice of pre-treatment (Section 3), the first 20 and the last 20 signatures of each writer were used to build the reference set and the test set, respectively. For the design of the BPN (Section 4), the first 20 signatures were used to build the training set, but the last 20 signatures were divided into two groups to build the test set (first 10) and the validation set (last 10).

2.2. Reference and training sets

The reference set for the kNN classifier and the training set for the BPN classifier were composed of 280 examples: 140 related to the genuine signatures of an individual (class \(\omega_1\)) and another 140 related to random forgeries defined as a subset of signatures from all the other individuals (class \(\omega_2\)). For each writer, the 140 examples in class \(\omega_1\) were obtained by rotating all the full directional PDF issued from the 20 reference signatures in a circular fashion from -6 to +6° in 2° increments. In the case of class \(\omega_2\), the 140 examples were obtained by choosing seven or eight reference signatures at random from 19 other individuals. The cardinality of classes \(\omega_1\) and \(\omega_2\) is equal in order not to favour one class over the other.

2.3. Test set for the kNN classifier

The test set for the kNN classifier used in Section 3 contains 160 examples: 20 genuine test signatures (class \(\omega_1\)) and 140 signatures taken at random from all the other writers (class \(\omega_2\)), in the same way as for the reference and training sets. In this case, we do not introduce any rotation in class \(\omega_2\), since it is necessary to check the classifier performance with statistically independent signatures.

2.4. Test set for the BPN classifier

The test set for the BPN classifier used in Section 4 contains 105 examples: 10 genuine test signatures (class \(\omega_1\)) and 95 signatures taken at random from all the other writers (class \(\omega_2\)). The 10 examples of class \(\omega_1\) are the first 10 signatures of the test set of one individual. The 95 examples of class \(\omega_2\) are five random examples of the last 10 signatures of the test set of the other 19 individuals. The cardinality of class \(\omega_2\) was reduced to offset the diminution in the cardinality of class \(\omega_1\), but not too much in order not to affect the accuracy of the performance measure.

2.5. Validation set for all classifiers

The validation set for all classifiers used in Section 5 contains 105 examples: 10 genuine test signatures (class \(\omega_1\)) and 95 signatures taken at random from all the other writers (class \(\omega_2\)). The 10 examples of class \(\omega_1\) are the last 10 signatures of the test set of one individual. The 95 examples of class \(\omega_2\) are five random examples of the last 10 signatures of the test set of the other 19 individuals.

2.6. Performance measures

The performance of each classifier is evaluated globally for the 20 writers and for 25 experiments for which the reference, training and test sets are changed each time. Thus, in each experiment class \(\omega_1\) will always be the same, but class \(\omega_2\) will be always different. In this way, it is possible to reduce the bias that could have been introduced by particular random forgeries. Classifier performance is measured by means of the total error rate \(e_t\), expressed in terms of \(e_1\) (type I error rate, the false classification of genuine signatures), \(e_2\) (type II error rate, the false classification of random forgeries) and \(P[\omega_1]\), the a priori probability for classes \(\omega_1\), which is set at 0.5 in our case: \(e_t = (e_1 \times P[\omega_1]) + (e_2 \times P[\omega_2])\). When appropriate, we use the total rejection rate \(R_t\), which is obtained with an equation similar to the latter one where error rates \(e_i\) are substituted with rejection rates \(R_i\), (\(R_t\), rejection of genuine signatures and \(R_2\), rejection of random forgeries). Finally, in order to take into account in a single parameter the total error and rejection rates, a reliability factor\(^{46}\) defined as follows: \(RF = (100 - e_t - R_t)/(100 - R_t)\) is used to find the best configuration of the BPN classifier.

3. CHOICE OF PRE-TREATMENT ON THE FULL DIRECTIONAL PDF

The goal of pre-treatment on the full directional PDF is to enhance its discriminating power, in other words, to improve the performances of a classifier that uses the PDF as an input vector. A classifier will be more efficient if the dimension of the input vector is not too big (unnecessary data) and if the variations in the input vector are not too abrupt (noisy data). Unfortunately, the full directional PDF does not have these characteristics (Fig. 1). Thus, pre-treatment, including filtering and compression, must be carried out. The purpose of this section is to determine which is the best...
combination based on the following heuristic: "the best performance having the higher compression rate and the shorter computer time".

3.1. Description of the pre-treatments

The treatments that we have chosen to evaluate can be classified in three categories. The first, called the integration filter (I), consists of making a summation of the full directional PDF samples inside a window, whose width is equal to the compression step which can take the values 1, 2, 3, 4, 5, 6, 9, 10 (A) or 12 (B). The second, called the undersampling filter (U), consists of making a simple undersampling of the full directional PDF with the value of the compression step. The third, called weighted filter, consists of first smoothing the full directional PDF with three types of filters: rectangular (R), triangular (T) and binomial (B), whose window width can take various values: 3, 5, 7, 9 or 11(A), and then performing a simple undersampling of this smoothed curve. In each case the process is the same and only the coefficient number varies with the window width and its value will depend on the filter used. In addition, to facilitate the comparisons, the sum of the coefficients is always equal to one to maintain the window area constant.

3.2. Experimental protocol

To find the pre-treatment that improves the discriminating power of the directional PDF the most, we used the kNN classifier because it permits the evaluation of a lower limit of the total error rate when the maximum available information is kept in memory. The reference set and the test set needed by a kNN classifier have been described in Section 2.2 and 2.3, respectively. The total error rate $e_k$ as defined in Section 2.6, is evaluated for each writer in the signature database. The mean of these $e_k$ is then determined to obtain the global total error $e_{avg}$ of the verification system. This procedure is repeated 25 times with 25 different reference and test sets. Finally, it is the mean of these 25 $e_{avg}$ that is used to choose the best pre-treatment. In this part of the study, no rejection is permitted and it is the first minimum distance met that is taken into consideration.

3.3. Results analysis

We note a two-fold reduction in the error as soon as a smoothing is performed. In addition, up to a compression step of 6, the width of the window does not affect the result, while above this the performance is slightly lower when the width diminishes. Except for the integration filter, the performance seems to reach
a maximum at this compression value. The simple undersampling, at the other hand, reduces the error less and even after a compression step of 6 the error increases. Consequently, we already know that simple undersampling is not a good pre-treatment to apply to the full directional PDF. Finally, in all cases the standard deviation is low enough (between 0.2 and 0.35) so that we may conclude that the classification results are stable for each pre-treatment studied.

From our statistical analysis we retained the integration filter with a compression step of 10, which corresponds very well to our heuristic. Consequently, by applying this pre-treatment on the full directional PDF, we increase its discriminating power roughly two fold ($\epsilon$ drops from 5.94 to 2.69%) and, overall, we reduce the input vector dimension from 180 to 18. This is very important since it will greatly facilitate the design and training of the BPN classifier.

4. BACKPROPAGATION NETWORK CLASSIFIER

4.1. Backpropagation network (BPN)

Neural networks present a computational paradigm for constructing classifiers that can perform as accurately as conventional techniques. For the first stage of the AHSVS we used a completely connected feed-forward neural network with the classical backpropagation learning algorithm, more simply known as the Backpropagation Network (BPN) which is described in detail in many textbooks. To build a BPN, there are many parameters to choose from dealing with the network size or the learning law. Unfortunately, there is no way to determine them rigorously since they are strongly dependent on the application. The first is the number of hidden layers, which has been settled to one (1) since many authors consider that a single hidden layer is sufficient for most applications. The number of neurons on the input layer ($N_i$) is 18, which corresponds to the dimension of the vector $F(\theta)$ after applying the best pre-treatment found in Section 3.3. The number of neurons on the hidden layer ($N_h$) is two, since we have two classes ($\omega_1$ and $\omega_2$) and we want to use various rejection methods. It is not so easy to find the number of neurons on the hidden layer ($N_h$) whose upper limit is theoretically $2N_i + 1$. Up to now, only rules of thumb have been proposed to determine $N_h$, and we have arbitrarily settled $N_h = 12$, a number included between the maximum and the minimum proposed by the various methods. Later, we have adjusted this number by examining its influence on the global performance of the AHSVS (see Section 4.4). Concerning the learning law, there are two parameters to choose: the learning rate $\eta$ and the smoothing rate or momentum $\alpha$. Again, there is no way to find a rigorous value for these parameters. Moreover, the empirical rules proposed are often contradictory. Consequently, after a few preliminary trials, we arbitrarily decided to settle $\eta = 0.6$ and $\alpha = 0.0$. In order to facilitate the start of the training phase, the weights should be initialized to small random values, and the bias term should be used to avoid a saturation of the output of the neurons. We therefore used a bias term for the hidden and output layers and all weights were initialized randomly in the $[-0.1$ to 0.1] range. One last way to improve the convergence time of the BPN during the training phase is to normalize the input vectors between 0 and 1. This is particularly true when the data values are very similar. Since this is so in our case, we have normalized all training, test and performance sets used by the BPN.

With the BPN, the training phase is critical, especially when the data to be classified are not clearly distinguishable and when there are not enough examples to conduct training. In this case, the training phase can be very long and it may even be impossible to obtain an acceptable performance. Since this is the case for our application (few signatures with high variability): first, we have defined a criterion for stopping the training phase (Section 4.2); second, we have evaluated several rejection methods to improve the decision taken by this type of classifier (Section 4.3); finally, we have adjusted the number of neurons in the hidden layer of the BPN in order to increase the global performance of the first stage of the AHSVS (Section 4.4).

4.2. Stopping criterion

It is well known that the training phase is crucial in the design of a BPN classifier. The major difficulty is to decide on what basis to stop training. For the BPN we can use the error value $\epsilon$ (the difference between the desired output and the actual output) to stop training when this is lower than a pre-established limit (ideally 0) for all examples included in the training set. However, it is not always possible to reach this stopping criterion, especially when the input data are not clearly separated. A variant of this method, which takes this problem into account, is to stop training when the Root Mean Square (RMS) error on the training set is lower than a fixed threshold (ideally 0). This very popular method was not selected because it is not well suited to our application. Indeed, it is not absolutely necessary for us to have a small RMS error to obtain an acceptable performance of the classifier. In other words, we do not want the BPN to learn to give the exact output level but to make a good decision. This may be done well before the RMS error becomes weak. Consequently, we have based our stopping criterion on the performance measure $\epsilon_m$ defined in Section 2.6. Thus, the network weights are adjusted for each example of the training set and once all examples have been presented to the network (later referred to as a presentation), we freeze the weights and we evaluate $\epsilon_m$ on this training set (later referred to as $\epsilon_m$). In this way, we measure the memorization performance of the BPN classifier. However, the first objective of the BPN classifier is to have a good generalization performance,
that is, on examples not seen during the training phase. It is thus logical to stop training on the generalization performance measure of the BPN classifier. To do this, we evaluate $e_g$ on the test set (later referred to as $e_{tg}$). The drawback of the BPN, which is known as the "overtraining phenomenon", is that we must stop training when $e_{tg}$ goes through a global minimum. We have not retained this criterion since preliminary results showed that for our data $e_g$ was quite noisy and the curve was relatively flat after a few hundred presentations, and that the overtraining phenomenon was often imperceptible even after many thousands of presentations. Under these conditions, it would be best to stop training as soon as $e_{tm}$ and $e_{tg}$ are both almost stable (i.e. when the slope of the curve is very weak). However, this method needs a great deal of computing time and is very unstable when the data are noisy. For these reasons, we decided to stop training when $e_g$ is stable for all 20 writers. This consists of finding the number of presentations ($NBP = T_p$) after which the generalization performance for each writer is not significantly improved with longer training. Nevertheless, in some cases the training phase could be greatly reduced if we stopped it when the memorization and generalization performances are acceptable, that is, when $e_{tm} < T_m$ and $e_{tg} < T_g$. The thresholds $T_m$ and $T_g$ must be settled to 0, since we do not want to impose a limit on the BPN classifier's performance in either memorization or in generalization. $T_m$, the maximum number of presentations used in the training phase, is determined by means of an experimental protocol. Figure 2 illustrates the results found with this experimental protocol. Since the exponential $e_g$ curve is very noisy, we smoothed it before calculating its slope by curve-fitting. We have tried the two following slope values $10^{-3}$ and $10^{-4}$ and the resulting $T_p$ values were 250 and 650, respectively. As we can see in Fig. 2, the first value is quite low and the second is a little too high. Thus, we have set $T_p$ at 500 to obtain a sufficiently stable curve without excessive computation time. In conclusion, the stopping criterion used in this study is stated as follows: "the training phase will be stopped when $(e_{tm} = T_m = 0)$ and $(e_{tg} = T_g = 0)$ or when $(NBP > T_p = 500)$.

4.3. Rejection criterion

For systems not requiring an immediate decision, the addition of a rejection criterion to the decision rule allows significant improvement in classifier performance by refusing to classify doubtful cases. However, even if the cost of a rejection is lower than that of an error, the rejection rate must be as weak as possible for two reasons. The first is that we can also reject good decisions and if the rejection rate of good decisions becomes higher than that of bad decisions, the classifier's performance will be decreased instead of increased. The second concerns classifier utility. Indeed, in an extreme case the classifier can have an error rate of 0% but a rejection rate of 100%. In this case, the
classifier is useless. One way to take this phenomenon into account is to use the reliability factor (RF) defined in Section 2.6. With this measure, we consider the error rate as well as the rejection rate in evaluating the classifier's performance. Consequently, each rejection criterion was evaluated by means of the total error rate ($e_t$), the total rejection rate ($R_t$) and the reliability factor (RF).

Table 1 shows the definition of all the rejection criteria used, where $o_1$ and $o_2$ are the output levels for classes $c_1$ and $c_2$, respectively, and where $T_m$, $T_A$, $T_{R1}$ and $T_{R2}$ are adjustable thresholds. Except for RC4, the rejection area is a linear function of the threshold value which varies between 0 and 1. Since the rejection areas for RC3 and RC4 are equivalent, and given that we are more familiar with a relative variation induced between 0 and 100%, we have used only RC3. From these rejection areas, it is not possible to choose among rejection criteria RC1 to RC3. Consequently, we have to evaluate them by means of an experimental protocol. We have also examined whether or not a combination of RC1 and RC2 (RC5) or RC3 (RC6) can improve classifier quality.

Figure 3 shows the mean value and standard deviation for $e_t$, $R_t$ and RF of the BPN classifier obtained by varying the thresholds $T_m$, $T_A$, $T_{R1}$, for the rejection criteria RC1, RC2, RC3 and one example of RC5, respectively. (For the reasons mentioned before, RC4 has not been evaluated.) These results show a better

<table>
<thead>
<tr>
<th>Rejection criterion</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>RC1</td>
<td>$\max{o_1, o_2} &lt; T_m$</td>
</tr>
<tr>
<td>RC2</td>
<td>$\max{o_1, o_2} - \min{o_1, o_2} &lt; T_R1$</td>
</tr>
<tr>
<td>RC3</td>
<td>$\max{o_1, o_2} - \min{o_1, o_2} &lt; T_R1$</td>
</tr>
<tr>
<td>RC4</td>
<td>$\max{o_1, o_2} - \min{o_1, o_2} &lt; T_R2$</td>
</tr>
<tr>
<td>RC5</td>
<td>$\max{o_1, o_2} &lt; T_m$ and $\max{o_1, o_2} - \min{o_1, o_2} &lt; T_R1$</td>
</tr>
<tr>
<td>RC6</td>
<td>$\max{o_1, o_2} &lt; T_m$ and $\max{o_1, o_2} - \min{o_1, o_2} &lt; T_R1$</td>
</tr>
</tbody>
</table>

Fig. 3. Evaluation of the rejection criteria RC1, RC2, RC3 and RC5 by means of the total error ($e_t$) and rejection ($R_t$) rates and of the reliability factor (RF) for various values of their corresponding thresholds $T_m$, $T_A$ and $T_{R1}$. 
performance of the BPN classifier when the threshold values increase. This better performance, corresponding to a higher reliability (RF) and a lower total error rate ($e_t$), is obtained at the expense of a total rejection rate ($R_t$) that is clearly higher. As seen in Fig. 3, unlike RC2 and RC3, RC1 becomes effective only when the threshold $T_M$ is higher than 0.5. From Fig. 3 we can also conclude that RC1, RC2 and RC3 are equivalent when applied to our data. The results dealing with RC5 and RC6 simply show that the addition of RC2 or RC3 to RC1 is only effective when the threshold $T_A$ or $T_R$ is higher than a value for which it allows a better performance than that already obtained with the threshold $T_M$ (see, for example, RC5 in Fig. 3). This means that below this particular threshold $T_A$ or $T_R$, the rejection criteria RC1 and RC2 or RC3 reject exactly the same examples. All these results can easily be explained by the fact that our output levels are complementary.

In conclusion, we can say that, due to the output complementarity, the rejection criteria RC1, RC2 and RC3 are strictly equivalent and the rejection criteria RC5 and RC6 are only useful when we want to impose a lower limit on the BPN classifier performance via RC1. Consequently, the rejection criterion RC1 has been chosen only because of its practical aspect (less computation time). We now have to determine the threshold $T_M$, but this determination must be made with caution. A careful examination of our results proves that the application of a rejection criterion may degrade BPN classifier performance and also indicates that the threshold value must be high enough, but not too high. As already mentioned, BPN classifier performance increases at the expense of an increase in the rejection rate. Consequently, we chose the threshold $T_M$ according to the following heuristic: “obtain the best performance with an acceptable rejection rate”. For our AHSVS application a rejection rate of 5% could be considered as acceptable. In this case from Fig. 3 we find a value of 0.94 for the threshold $T_M$. In addition, our results show that the introduction of a rejection criterion improves the reliability (RF) of the AHSVS slightly, but sharply decreases its total error rate ($e_t$) via the type I error rate ($e_1$).

4.4. Backpropagation network (BPN) optimization

Essentially, there are two ways to optimize a BPN: (1) we can adjust the learning algorithm$^{18-21}$ and (2) we can modify its structure before$^{22-25}$ or during$^{26-28}$ the training phase. We only performed the BPN optimization on the structural aspect of the network. At the time the BPN was defined, we mentioned the importance on BPN performance of the hidden neurons (see Section 4.2). Unfortunately, there are no theoretical means for finding the optimum number of hidden neurons ($N_{hm}$). However, we know its upper limit, which is $N_{hm} = 2N_i + 1^{(12)} = 37$, in our case. Consequently, $N_{hm}$ must be found by experimentation in which $N_h$ varies gradually between 0 and $N_{hm}$.

The results of this experimentation are shown graphically in Fig. 4. From these results, we can see that the BPN performances clearly improve as soon as we have a hidden layer and that, for our database, the effect of the number of hidden neurons is not very pronounced. A statistical analysis shows that the configuration with eight hidden neurons is the best because it has a lower error rate ($e_t = 1.24\%$) and rejection rate ($R_t = 4.506\%$) and a higher reliability factor (RF = 0.987). Thus, the eight-hidden-neuron configuration was used for the comparative study described in the discussion.

5. DISCUSSION

In order to evaluate the behaviour of the BPN classifier more accurately, we carried out a comparative study of the kNN classifier, the T classifier and the BPN classifier on the same data sets. The kNN and BPN classifiers use a new definition of the data sets, as defined in Sections 2.2, 2.4 and 2.5. Therefore, there is no comparison between performances obtained now in this comparative study and those obtained previously in Section 3 for the kNN classifier and in Section 4 for the BPN classifier. The T classifier on the other hand, except for the validation set which is rigorously the same for all classifiers, uses the following reference and test sets. To build the T reference set, we select at random a few examples from the first 20 genuine signatures and apply a rotation as in Section 2.2. For this study, the number of examples before rotation for the T reference set was varied between 1 and 10 with an increment of 1, thus forming 10 T classifiers. As for the T test set, which is used only to find the threshold value, class $\omega_1$ comprises the remaining examples of the first 20 genuine signatures and class $\omega_2$ comprises five random examples of the first 20 signatures of the 19 other individuals. Consequently, class $\omega_2$ always has the same dimension (95), but class $\omega_1$ has a dimension that varies according to the number of examples already picked up for the T reference set.

Before doing the comparative study, we have to find the best T classifier among the 10 we have simulated. Between three and seven examples, the difference is not very pronounced and we have chosen that which gives the best performance, that is, five examples to build the T reference set. Now, it is this best T classifier that will be used for the comparison. As expected, the BPN classifier gives a result ($e_t = 3.22\%$) that lies between the best given by the kNN classifier ($e_t = 1.68\%$) and the worst given by the T classifier ($e_t = 5.61\%$). This is also true for the stability of the decision which is reflected in the standard deviation. Our results clearly show that all classifiers are more unstable between writers than between experiments. This high standard deviation value indicates the difficulty that the classifiers have in identifying some writers. In the two cases, the BPN classifier is closer to the kNN classifier than to the T classifier.
6. CONCLUSIONS

The main objective of this work was to determine whether or not a BPN classifier could be used in the design of the first stage of a complete Automatic Handwritten Signature Verification System (AHSVS). To do this, we have chosen the directional Probability Density Function (PDF) as a global shape factor and the completely connected feed-forward neural network with the classical backpropagation learning algorithm referred to as the Backpropagation Network (BPN).

However, the dimension (180) of the PDF is too large to be properly manipulated by a BPN classifier. Consequently, in a first attempt and by means of the k Nearest Neighbour (kNN) classifier, we have determined the pre-treatment that improves the classification while decreasing the dimension of the input vector. The results in Section 3 show that the pre-treatment with an integration filter with a step of ten gave the best result.

In the case of the BPN classifier, the training phase is crucial and difficult to control. The major difficulty is to decide when to stop training. In Section 4.2, we defined a stopping criterion based on a measure of the performance of the BPN classifier both in memorization ($e_m$) and in generalization ($e_g$), as well as on a maximum number of presentations (NBP) of the training set. Using an experimental protocol, we decided that the training phase would be stopped when
When (NBP > 500), statistical analysis carried out later, during the evaluation of the performance of the BPN, showed that the majority (81%) of the networks stopped their training on the second condition and 14.2% of them stopped their training on the first condition with fewer than 100 presentations.

In order to improve the reliability of the BPN classifier, we have introduced a rejection criterion in the decision rule. In Section 4.3, we showed that in all cases the addition of a rejection criterion sharply decreased the total error rate and slightly increased the reliability at the expense of a rejection rate that is sometimes high. Due to the fact that our data are complementary, all the rejection criteria proposed are similar. Thus, we have chosen the easiest one (RC1) with a threshold (T_M = 0.94) giving an acceptable rejection rate (5%) for our application.

It is well known that the number of neurons on the hidden layer plays an important role on BPN performance. In Section 4.4, we evaluated this role by measuring BPN performance with and without the rejection criterion previously defined when the number of hidden neurons is varied. In our case, the addition or not of the rejection criterion does not significantly affect the number of hidden neurons that gives the best performance. A statistical analysis on our data has shown that the best configuration was with eight hidden neurons.

Finally, in order to better assess the potential of the BPN classifier, we made a comparative study of this one, the kNN classifier, which should give an upper limit for the performance, and the T classifier, which is very popular in spite of its failings. Since these latter do not use a rejection criterion in making their decision, we have used, in Section 5, a BPN classifier without the rejection criterion previously defined. In effect, the BPN classifier behaves much better than the T classifier and a little less well than the kNN classifier.

In conclusion, we can say that, once trained, the BPN classifier compares favourably with the kNN classifier since it has almost the same performance but with a shorter response time at generalization, especially if we use a hardware implementation of the BPN classifier. However, for our application the drawback of the BPN classifier is the way in which it deals with the training phase and this could undermine this conclusion somewhat.

Acknowledgements—This work was supported in part by a PSIR grant from the École de technologie supérieure to Jean-Pierre Drouhard and Robert Sabourin, and by grant OGP0106456 to Robert Sabourin from the NSERC of Canada. Master's student Mario Godbout, who took part in this research project, also received a scholarship from the École de technologie supérieure.

REFERENCES

1. R. Sabourin, Une Approche de Type Compréhension de Scène Appliquée au Problème de la Vérification Automatique de l'Identité par l'Image de la Signature Manus-

About the Author—JEAN-PIERRE DROUHARD received the M.Sc. degree in electronic, electrical and control engineering from the University of Caen, France, in 1972 and the M.Sc.A. and Ph.D. degrees in biomedical engineering from the École Polytechnique de Montréal in 1975 and 1979, respectively. From 1980 to 1989 he was a research associate for the Biomedical Engineering Institute of the École Polytechnique de Montréal. In 1989, he joined the staff of the École de Technologie Supérieure, Université du Québec, Montréal, P.Q., Canada, where he is currently Professor in the Département de Génie de la Production Automatisée. His current research interests are in the application of the artificial intelligence techniques such as expert systems, artificial neural networks and fuzzy systems in the fields of pattern recognition and computer vision.

About the Author—ROBERT SABOURIN received B.ing., M.Sc.A. and Ph.D. degrees in electrical engineering from the École Polytechnique de Montréal in 1977, 1980 and 1991 respectively. In 1977 he joined the physics department of the Université de Montréal where he was responsible for the design and development of scientific instrumentation for the Observatoire du Mont Mégantic. In 1983, he joined the staff of the École de Technologie Supérieure, Université Québec, Montréal, P.Q., Canada, where he is currently Professor in the Département de Génie de la Production Automatisée. His research interests are in the areas of computer vision, scene understanding, segmentation, structural pattern recognition, neural networks and fuzzy systems, character recognition and signature verification.

About the Author—MARIO GODBOUT received the B.ing. degree in Production Automatisée from the École de Technologie Supérieure de Montréal in 1992. He is currently making a master degree at that same school. His research interests are in the field of artificial intelligence.