CSM neural network for degraded printed character optical recognition

A. Namane, A. Guessoum, E.H. Soubari, P. Meyrueis

Laboratoire du Traitement du Signal et de l’Image (LATSI), Department of Electronic, Faculty of Technology, University Saad Dahlab of Baida, Algeria
ICUBE, Laboratoire des Sciences de l’Ingénieur, de l’Informatique et de l’Imagerie, University of Strasbourg, E.Phot Group, France

Article info
Article history:
Received 9 February 2013
Accepted 30 March 2014
Available online 13 April 2014

Keywords:
Component
Complementary similarity measure
Degraded printed characters
Hamming net
Competitive neural network
OCR
Character recognition
CSM net
Combination method

Abstract
In pattern recognition applications, the classification power of a system can be improved by combining several classifiers. Obviously performance of the system cannot be improved if the individual classifiers make all the same mistakes, thus it is important to use different features and different structures in the individual classifiers. In this context, we propose a two subnets neural network called CSM net. The first subnet, or similarity layer, is operating as a similarity measure neural network; it is based on the complementary similarity measure method (CSM). The second subnet is a competitive neural network (CNN) based on the winner takes all algorithm (WTA) that is used for the classification. In the proposed neural architecture, the statistical CSM method is analyzed, and implemented in the form of a feed forward neural network, it is named “similarity measure neural network” (SMNN). We show that the resulting SMNN synaptic weights are modified versions of the model patterns used in the training set, and that they can be considered as a memory network. We introduce a relative distance data calculated from the SMNN output, and we use it as a quality measurement tool of the degraded characters, what makes the SMNN classifier very powerful, and very well-suited for features rejections. This relative distance is used by the SMNN and compared to a first rejection threshold to accept, or reject, the incoming characters. In order to guarantee a higher recognition and reliability rates for the cascaded method, the SMNN is combined with a second subnet based on the WTA for classification using a second specific rejection threshold. These two subnets combination (CSM net) boost the performance of the SMNN classifier. This is resulting in a robust multiple classifiers that can be used for setting the entire rejection threshold. The experimental results that we introduce are related to the proposed method, but the tests are introduced with various impulse noise levels, as well as the tests with broken and manually corrupted characters, and characters with various levels of additive Gaussian noise. The experiments show the effective ability of the model to yield relevant and robust recognition on poor quality printed checks, and show that the CSM net outperforms the previous works, both in efficiency and accuracy.

1. Introduction
The processing of huge amounts of printed documents is a big task to be handled economically. In today’s world of information, forms, reports, contracts, letters and bank checks are generated everyday in large quantities. Hence the need to store, retrieve, update, replicate and distribute printed documents, becomes increasingly important [1,2].

The automatic reading of bank checks poorly printed, is one of the most significant applications in the area of pattern recognition, specifically in the third world, where the needs are important. A local bank can have to process daily thousands of poorly printed checks. The treatment of these documents is expensive [3,4]. The main targeted application of our proposed method is thus to improve third world bank checks processing. This is the case selected as an example, for two reasons:

- The printings and papers are of poor quality.
- The important daily use of account check numbers (ACN) for customer balances.

The ACN is used to retrieve the following data from the customer’s references database: the personal information about the customer’s accounts, and the customer's signatures for checks validations by signatures verifications. Bank customers are often asking about their accounts balance at the bank counter before doing a money withdrawal operation. Thus, the account check number is used in many account balance operations. Hundreds of thousands
of this kind of operations are operated daily in such banks. Our contribution consists in proposing a powerful system of characters recognition, able to read degraded printed ACN and customer’s first and last name (see Fig. 1), in order to facilitate and speed up the treatments of bank checks with reliability, even if they are poorly printed. This system can be applied to the recognition of post checks printed by an old typesetting machine, as they can be met in the third world.

The recognition of degraded documents remains an ongoing challenge in the field of optical character recognition (OCR). This is due to the degradation that could be originated from: low quality originals, quantization errors in the digitization process, and non optimal lights and contrasts settings [5].

In spite of significant improvements in the area of OCR [5–8], the recognition of degraded printed characters, in particular, is still lacking satisfactory solutions for commercial uses. Studies on designing recognition systems with high performances for degraded quality documents and characters are in progress along three different directions. The first one is to use a robust classifier, the second one is to enhance the degraded documents images for a better display quality, and an accurate recognition, and the third one is to use several classifiers [9–14].

Sawaki and Hagita [9] proposed a robust recognition method based on a complementary similarity measure (CSM) for characters with graphical designs and degraded characters. Their experimental results concern newspaper headlines with graphical designs and characters are in progress along three different directions. The first one is to use a robust classifier, the second one is to enhance the degraded documents images for a better display quality, and an accurate recognition, and the third one is to use several classifiers [9–14].

In this paper we propose a new method for degraded characters recognitions which consists in using only the models similarities in order to accept or reject an incoming character by a first classifier (SMNN), based on a character quality measurement compared to a first rejection threshold. This quality measurement parameter is also compared to a second rejection threshold, which represents a lower bound rejection threshold ($R_{th}$) to reject definitely the highly and severely degraded characters that are not recognizable and avoid more errors with the classifier.

All the recognition methods that are based on the use of features extractions should use appropriate classification methods. Strong characters degradations can severely affect the enhancement results [10–12] and the extracted features [14] which subsequently affect the classifier decisions. This could happen in the cases of stained or broken printed characters, or characters with missing parts.

In this paper we propose a new method for degraded characters recognitions which consists in using only the models similarities in order to accept or reject an incoming character by a first classifier (SMNN), based on a character quality measurement compared to a first rejection threshold. This quality measurement parameter is also compared to a second rejection threshold, which represents a lower bound rejection threshold ($R_{th}$) to reject definitely the highly and severely degraded characters that are not recognizable and avoid more errors with the classifier.

The second classifier (WTA) uses all the similarities produced by the first classifier (cases of rejection) as shown in Fig. 2. In general, these similarities must be higher, in order to make the subsequent
classifier; usually a competitive neural network, responds with a higher output related to the winner neuron. In case of produced low similarities, the CNN-based classifier cannot make a correct decision (equal higher outputs) concerning an incoming pattern and this could lead into a confusion state. Our contribution is summarized in the following three points:

- Design of a similarity measure neural network SMNN.
- Introduction of a relative distance used as a quality measurement.
- Design of a cascaded combination SMNN–WTA, or CSM net.

The CSM method, which has a tendency to keep the errors low for a higher rejection, is well-suited for rejection [16]. This better behavior in terms of rejection can be improved by the introduction of a relative distance, used as a character quality measurement, to make the CSM method very well suited for rejection.

The SMNN, as a first subnet, produces all the models similarities that will be used by a CNN network (in case of rejection), based on a winner takes all (WTA), used as a second subnet (see Fig. 2). The combination method is inspired from the Hamming net which combines Hamming distance for the similarity layer and the CNN for the classification. However, in the proposed combination, classification is either made by the SMNN subnet (case of SMNN acceptance) or by the WTA subnet (case of SMNN rejection), based primarily on the characters quality measurement (see Fig. 2). But in the Hamming net case, classification is always made by the WTA subnet using all the similarities produced by the Hamming subnet or similarity layer. Moreover, the CSM net works in both modes, namely, rejection and acceptance modes, whereas the Hamming net works only in the acceptance mode, in detriment of its reliability. The rejection mode for the CSM net is performed by the introduction of a second rejection threshold ($R_{LB}$).

This paper is then organized in the following way: Section 2 presents the character data analysis and preprocessing. Section 3 describes the Hamming subnet and CNN-based classifier. Section 4 presents the CSM method. Section 5 presents the design of SMNN subnet and character quality measurement. Section 6 describes the development of the proposed combined method, or CSM net. Finally, section 7 presents experimental results.

2. Character data analysis and preprocessing

2.1. Character data analysis

The characters that we consider come from a large number of post checks printed by old typesetting machines. These machines can print millions of these check books per year, even by using papers of poor quality. Our samples selection process is the following: After a check scanning with a random choice of test samples, out of a 500 issues harvested in 20 different banks offices in Algeria, the characters are visually cut manually using our own software realized with C++ language. These characters are Latin, with only black and white colours, and belong to two types of font. They form, after this process, a mixture of 35 printed characters distributed as follows.

Account number characters: 0 (652), 1 (918), 2 (810), 3 (548), 4 (1006), 5 (701), 6 (642), 7 (1108), 8 (612), 9 (561), C (2520), E (840), L (840).

Alphabetic characters: A (986), B (272), C (85), D (333), E (449), F (100), G (61), H (338), I (514), J (65), K (131), L (419), M (535), N (151), O (317), P (90), R (368), S (151), T (105), U (277), Y (58), Z (75).

where the first character represents the character class, and between brackets its number of samples.

The analysis of the printed checks reveals that the characters are severely altered. For instance by an excess of ink (stains) or

![Fig. 2. Proposed CSM net block diagram.](http://iranpaper.ir)
by a lack of ink. We found that all these degradations, present a good challenge for degraded characters automatic recognition because of their very poor quality. The usefulness of our proposed method is adapted to countries of the third world where this low quality is common. The data set represents an excellent database for our application. The results obtained confirmed the rationality of the test set that we proposed concerning the account check recognition, which is not a trivial task.

We have used and updated this data set for several years. Testing conditions (lighting, focus, paper quality and printing) are partially optimized by the preprocessing step. This step concerns in reducing the noise and in increasing the image character quality. We have done a large number of validating test; we have found the same results for several years. The effectiveness of our proposed method is such that it can be used by concerned banks very rapidly if necessary.

2.2. Preprocessing

Preprocessing generally consists in series of image to image transformations to improve the input images quality and to standardize and normalize the input for a given recognition system. Filtering is the first step of any recognition system to improve the image characters quality. We propose a Gaussian low pass filtering process for noise reduction to be applied to the input images. The binarization of an image has several advantages. One of the most significant advantages is certainly the low memory capacity needed, as well as the simplicity of the operators that are associated to it. The binarization procedure that we propose is carried out with Niblack’s method [17], using a local thresholding. Fig. 3 illustrates the results of the global and local thresholding applied to ACNs, and customer first and last name. Following the thresholding, the resultant image character is centered by positioning the image character centroid to the centre of a fixed size frame (40-by-40 pixels).

3. Hamming network

Hamming network was developed by Lippman [18], it uses the Hamming distance to measure the similarity, or dissimilarity, between an input vector, and the prototype vectors, stored in the memory [19]. Hamming network is widely used in the area of pattern recognition [20,21]. The basic Hamming neural network, or Hamming net, generally consists in two layers or subnets, namely, the similarity measure layer (Hamming subnet) and the competitive network (CNN subnet), or classification layer based on the winner takes all (WTA). The general structure of a Hamming neural network is given by Fig. 4. In this figure, the vector \( x \) is the input pattern having a dimension \( n \), which represents the space of observation and characterizes the unknown pattern. Similarity \( a \) is the first network output vector, and also the second network input of dimension \( p \). The vector \( O \) is the second output network, of dimension \( p \), that represents the dimension of space interpretation (number of classes).

3.1. Hamming subnet

The first subnet or similarity layer is \( n \)-dimensional and has \( p \) neurons to store \( p \) pattern models. If the \( p \) models are represented by their patterns: \( y^{(1)}, y^{(2)}, \ldots, y^{(p)} \), they can be stored in the memory network by choosing the synaptic weights as follows:

\[
W_H = \frac{1}{2} \begin{bmatrix} y^{(1)} & y^{(2)} & \cdots & y^{(p)} \end{bmatrix}^T
\]

(1)

with:

\[
y^{(m)} = [y_1^{(m)}, y_2^{(m)}, \ldots, y_n^{(m)}]
\]

where \( y_1 = 1 \) (black) or \(-1\) (white).

The synaptic weights represented by Eq. (1) are then:

\[
W_H = \frac{1}{2} \begin{bmatrix} y^{(1)} & y^{(2)} & \cdots & y^{(p)} \end{bmatrix} = \begin{bmatrix} W_{H1} & W_{H2} & \cdots & W_{HP} \end{bmatrix}
\]

(2)

In the Hamming subnet, the stored prototype or model \( y \) and input vector \( x \) are coded in a bipolar way (1 and -1). For the Hamming subnet, one can choose the similarity between the two vectors as follows:

\[
a = \frac{1}{2} (y \cdot x^T + n)
\]

(3)
where \( b_i = \frac{1}{2} \) represents the bias of the network, \( W_{ii} \) is the \( i \)th model vector, or the model \( y^{(i)} \) stored in the matrix \( W_P \) and \( x \) is the input vector as shown in Fig. 5.

3.2. CNN subnet

The second subnet, or competitive layer, in the Hamming net is a competitive learning network generally based on the winner takes all network (WTA), as shown earlier in Fig. 4. The output neurons; \( O_1, O_2, \ldots, O_P \) represent the \( P \) classes that the network must distinguish. For each output neuron \( O_i \) a synaptic weight vector \( W_i \) is associated, that relates it to all the input neurons \( a_i \). The values of different \( W_i \) are defined by a competitive training process (unsupervised learning). In general, the network consists of \( N \) input neurons and \( P \) output neurons.

The competition between neurons in the network is based on calculating the scalar product between input vector, or similarity vector \( a \) and synaptic weight vector \( W_i \):

\[
O_i = W_i \cdot a_i
\]

One thus seeks the output neuron \( O_i \) which was subjected to the strongest activation and designates it as the winner:

\[
O_i = \max_{j=1}^P W_j \cdot a_j \quad \text{and} \quad O_i > O_j \quad \forall j \in P \quad \text{with} \quad (i \neq j)
\]

(7)

The synaptic weight modification could be simply done by adding to the old synaptic weight vector \( W_i \) of the winner neuron, a fraction (difference between \( a \) and \( W_i \)) as follows:

\[
W_i = W_i + \phi \cdot (a - W_i)
\]

where \( \phi \) is a training parameter generally in the range 0.1–0.7.

4. Complementary similarity measure method

The CSM is used as discriminating functions for the recognition phases applied to binary character images [9, 16, 22]. It is based primarily on the similarity measures between two binary images; a model \( y \) stored in the training set, and a sample image \( x \) of the size \( n = \sqrt{axv}/n \). The attribution of a sample image character to a single class among other ones (where each class is represented by one or more model images) is performed by using the highest score of similarity between \( x \) and \( y \). These two image characters are expressed as \( n \)-dimensional binary features vectors as follows:

\[
x = [x_1, x_2, \ldots, x_n, \ldots, x_n] \quad \text{where} \quad x_i = 0 \quad \text{(white)} \quad \text{or} \quad 1 \quad \text{(black)}.
\]

\[
y = [y_1, y_2, \ldots, y_n, \ldots, y_n] \quad \text{where} \quad y_i = 0 \quad \text{(white)} \quad \text{or} \quad 1 \quad \text{(black)}.
\]

The complementary similarity measure \( S_c(x,y) \) of \( x \) and \( y \) is defined as:

\[
S_c(x,y) = \frac{a \cdot e - b \cdot c}{T(n-T)}
\]

(9)

where:

\[
a = \sum_{i=1}^n x_i \cdot y_i = x \cdot y^T = y \cdot x^T
\]

(10)

\[
b = \sum_{i=1}^n (1-x_i) \cdot y_i, \quad c = \sum_{i=1}^n y_i \cdot (1-y_i)
\]

\[
e = \sum_{i=1}^n (1-x_i) \cdot (1-y_i) \quad \text{and} \quad T = \|y\| = \sum_{i=1}^n y_i
\]

The parameter \( T \) represents the number of black pixels in the model \( y \).

According to our experiments, the number \( T \) of black pixels in the model does not affect the value of \( S_c \). For an input image that represents a model from the training set, and according to Eq. (9), the highest value of \( S_c \) is always equal to one \((a = T, b = 0, c = 0 \text{ and } e = n - T)\) whatever the value of its \( T \) is. To show that the similarity is independent of the number \( T \), we first corrupt all the thirteen characters of the ACN with different values of impulse noise (see Fig. 6), and second we calculate the corresponding complementary similarity. From Fig. 7, the complementary similarity for different values of \( T \) is the same. What clearly confirms that the \( S_c \) is independent of \( T \) for the same degradations. The \( S_c \)'s values of the same degradations are very close, showing very small values for \( S_c \) standard deviation (see Table 1).

The similarity \( a \) is the number of black pixels in the model that is identical to the one of the sample character. Hence this similarity is always less or equal to the number of black pixels \( F \) of the image to be recognized. This leads to the determination of the upper bound similarity as follows:

\[
a \leq F \Rightarrow n \cdot a \leq n \cdot F \iff n \cdot a - T \cdot F \leq (n - T) \cdot F
\]

which corresponds to:

\[
S_c(x,y) = \frac{n \cdot a - T \cdot F}{T(n-T)} \leq F
\]

(11)

In case of \( a = F \), \( S_c \) is equal to the upper bound: \( F \).

Figs. 8 and 9, which quantify robustness of CSM method at various noise levels, is brought from our earlier work on CSM-based feature extraction [13]. Fig. 8 shows from left to right; the clean character image, and its corrupted versions with 25% and 50% of impulse noise. The plots of the complementary similarity measures of these three characters, with 95 characters models (numeral, upper case, lower case and special characters) are shown in Fig. 9. The horizontal and vertical axes represent the model class,
and its corresponding similarity value of a sample image and 95 characters models. It is well shown from Fig. 9 that the similarity of the sample image in Fig. 8(a) corresponds to the highest value of the class “3” (thin curve), whereas the similarities with other models is of lower values. The similarity plots of the sample images in Fig. 8(b) and (c), with all models, show also highest values in the recognized class (in this case “3”).

5. SMNN design and character quality measurement

5.1. SMNN design

The design of SMNN is based primarily on the development of Eq. (9), and the parameter $F$ introduced earlier that corresponds to the black pixels in the sample image character $x$. Eq. (9) could be made in the following form:

$$S_c(x, y) = \frac{n - a - T \cdot F}{T(n - T)}$$

$$S_c(x, y) = \alpha \frac{a - F}{n}$$

With $F = ||x|| = \sum_{i=1}^{n} x_i$, and $\alpha = \frac{n}{n - T}$.

By substitution of Eq. (10) in Eq. (13) we obtain the following relation:

$$S_c(x, y) = \alpha (y - \beta) \cdot x^T$$

$$S_c(x, y) = S \cdot x^T$$

Where:

$$S = \alpha \cdot (y - \beta)$$

and $\beta = \frac{T \cdot F}{n}$, $\beta \in [0, 1]$ is a model of density which represents the number of the model black pixels over the image size.

According to Eq. (15), the complementary similarity $S_c(x, y)$ between a sample image $x$ and a model $y$ could be calculated only by using an inner product between $x$ and a new model $S$. From Eq. (16), the model $S$ is a modified version of $y$. The expression of similarity given by Eq. (15) leads to the implementation of the statistical CSM method in a neural network, named similarity measure neural network (SMNN). Thus, in this neural network, the new models $y^{(m)}$, $m = 1, \ldots, P$, are used as memory network instead of the $y^{(m)}$ models themselves, as they are used with the Hamming net.

If $P$ models of $y$, using their corresponding new models $S$, are represented by the patterns: $S^{(1)}, S^{(2)}, \ldots, S^{(P)}$, they can be stored in the memory network (SMNN) by choosing the following synaptic weights $W_{CSM}$:

$$W_{CSM} = \left[ S^{(1)} \ S^{(2)} \ \ldots \ S^{(P)} \right]$$

The memorized pattern $S^{(m)}$ has $n$ dimension representing the image size and could be written as follows:

$$S^{(m)} = \left[ S_1^{(m)} \ S_2^{(m)} \ \ldots \ S_n^{(m)} \right]$$

The synaptic weights represented by Eq. (17) are then:

$$W_{CSM} = \left[ S_1^{(1)} \ S_2^{(1)} \ \ldots \ S_n^{(1)} \ S_1^{(2)} \ S_2^{(2)} \ \ldots \ S_n^{(2)} \ \ldots \ S_1^{(P)} \ S_2^{(P)} \ \ldots \ S_n^{(P)} \right]$$

By substitution of the Eq. (16) in the Eq. (19) we obtain the following CSM synaptic weight matrix:

$$W_{CSM} = \left[ x_1 \cdot (y_1^{(1)} - \beta_1) \ x_1 \cdot (y_2^{(1)} - \beta_1) \ \ldots \ x_1 \cdot (y_n^{(1)} - \beta_1) \ x_2 \cdot (y_1^{(2)} - \beta_2) \ x_2 \cdot (y_2^{(2)} - \beta_2) \ \ldots \ x_2 \cdot (y_n^{(2)} - \beta_2) \ \ldots \ x_p \cdot (y_1^{(P)} - \beta_p) \ x_p \cdot (y_2^{(P)} - \beta_p) \ \ldots \ x_p \cdot (y_n^{(P)} - \beta_p) \right]$$

Table 1

<table>
<thead>
<tr>
<th>Class</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>C</th>
<th>L</th>
<th>E</th>
<th>Mean</th>
<th>Standard deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noise level (12.5%)</td>
<td>0.762</td>
<td>0.766</td>
<td>0.765</td>
<td>0.764</td>
<td>0.767</td>
<td>0.771</td>
<td>0.769</td>
<td>0.760</td>
<td>0.765</td>
<td>0.766</td>
<td>0.775</td>
<td>0.764</td>
<td>0.766</td>
<td>0.0038</td>
<td></td>
</tr>
<tr>
<td>Noise level (25.0%)</td>
<td>0.571</td>
<td>0.567</td>
<td>0.576</td>
<td>0.570</td>
<td>0.563</td>
<td>0.571</td>
<td>0.577</td>
<td>0.571</td>
<td>0.570</td>
<td>0.567</td>
<td>0.584</td>
<td>0.589</td>
<td>0.571</td>
<td>0.573</td>
<td>0.0071</td>
</tr>
<tr>
<td>Noise level (37.5%)</td>
<td>0.374</td>
<td>0.371</td>
<td>0.381</td>
<td>0.381</td>
<td>0.375</td>
<td>0.375</td>
<td>0.384</td>
<td>0.394</td>
<td>0.384</td>
<td>0.365</td>
<td>0.384</td>
<td>0.386</td>
<td>0.370</td>
<td>0.379</td>
<td>0.0078</td>
</tr>
<tr>
<td>Noise level (50.0%)</td>
<td>0.206</td>
<td>0.209</td>
<td>0.208</td>
<td>0.208</td>
<td>0.211</td>
<td>0.201</td>
<td>0.212</td>
<td>0.230</td>
<td>0.207</td>
<td>0.196</td>
<td>0.218</td>
<td>0.202</td>
<td>0.199</td>
<td>0.208</td>
<td>0.0088</td>
</tr>
<tr>
<td>$T$</td>
<td>522</td>
<td>343</td>
<td>456</td>
<td>490</td>
<td>462</td>
<td>503</td>
<td>479</td>
<td>346</td>
<td>467</td>
<td>508</td>
<td>544</td>
<td>367</td>
<td>560</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
between a sample image respectively. After calculation of the values (see Eq. (9)). The superiority of the SMNN subnet over the Hamming subnet in terms of math theory can be noticed according to Eqs. (3) and (9). Eq. (9) calculates the complementary similarity measure in the output node.

The complementary similarity measure in the output node of the network is given by:

\[
S_c(x, y) = W_{CSM} \cdot x^T
\]

where: \(y = \begin{bmatrix} y^1 & y^2 & \ldots & y^p \end{bmatrix}^T\).

The complementary similarity measure in the output node \(j\) of the network is given by:

\[
S_c^{(j)}(x, y^{(j)}) = W_{CSM}^{(j)} \cdot x^T
\]

where:

\[
W_{CSM}^{(j)} = \begin{bmatrix}
x_j \cdot (y_1^{(j)} - \beta_i) & x_j \cdot (y_2^{(j)} - \beta_i) & \ldots & x_j \cdot (y_p^{(j)} - \beta_i)
\end{bmatrix}
\]

represents the line \(j\) of the synaptic weight matrix \(W_{CSM}\).

In the Hamming subnet we have shown how the similarity values \(a\), and the synaptic weights \(W_a\) are calculated and produced respectively. The development of the SMNN subnet is based on four similarity values contributing to one global similarity value which is the \(S_c\) value (see Eq. (9)). The superiority of the SMNN subnet over the Hamming subnet in terms of math theory can be noticed according to Eqs. (3) and (9). Eq. (9) calculates the complementary similarity value using four distinct similarity values, namely, the similarity values \(a\) (similar to the one of Hamming subnet) and three other similarities \((b, c, e)\). Whereas from Eq. (3), the Hamming subnet uses only one similarity value which produces, at its output, a weak blob when comparing to that of CSM (see Fig. 12) that is not always sufficient to stimulate correctly the subsequent network. In the SMNN subnet, the produced similarities values present a well and pronounced blob that is sufficient to stimulate the subsequent network, or WTA network, even for highly degraded characters. Fig. 12 illustrates a similarity comparison for the two subnets which confirms the superiority of the SMNN subnet over the Hamming subnet for a given clean character, corrupted characters with 25% and 50% impulse noise (see Fig. 11). In this experiment, the SMNN produces the highest similarity value for label character “1” and is still in this position even for a highly corrupted character with 25% and 50% of impulse noise.

5.2. SMNN for character quality measurement

For a given pattern, the SMNN subnet output is a similarity measure vector (SMV) having a length equal to the number of model patterns with a higher value that corresponds to patterns class. The relative distance data is calculated from the SMV and used as a character quality measurement, which makes the SMNN classifier very powerful, and very well-suited for rejections. This relative distance is a measurement between two members related to two classes. The first and second classes correspond to the first and second highest scores of \(S_c\) respectively. After calculation of the

Fig. 9. CSM method robustness for similarity measure at different noise levels.

Fig. 10. Proposed similarity measure neural network (SMNN) structure.

Fig. 11. Input images used for complementary similarity measure (a) clean data (b) corrupted data with 25% impulsive noise (c) corrupted data with 50% impulsive noise.
similarities $S_e (0 \leq S_e \leq 1)$ between a sample image $x$ and the $P$ existing models $y$ using Eq. (21), only the two highest scores of $S_e$, namely, $S_{e1}$ and $S_{e2}$ $(S_{e1} > S_{e2})$ are used to calculate the relative distance. This could be transformed into two lowest values as: $\delta_1 = 1 - S_{e1}$, $\delta_2 = 1 - S_{e2}$. A typical classification criterion, which is used, consists in rejecting a pattern if the relative distance is:

$$\xi = \frac{\delta_2 - \delta_1}{\delta_2} < R_c$$

(23)

with $\delta_1 = \min_{m=1,2,\ldots,P}\{\delta_m\}$ and $\delta_2 = \min_{m=1,2,\ldots,P}\{\delta_m\}$.

This rejection is based on the comparison of the two lowest values $\delta_1$ and $\delta_2$ for an incoming pattern $x$. The rejection threshold $R_c$ can be chosen for depending on the user requirements. The requirement is that the user takes into account the reliability of the system. The check processing must be achieved with a high reliability, which means with a low error rate, or without errors. This could be done by the choice of the rejection threshold. Higher values of $R_c$ ensures low errors rates (more rejection), or even no errors, but the lower values of $R_c$ ensures more acceptance, what means more errors.

The relative distance $\xi$ can be used also as a quality measurement of the printed character as shown in Fig. 13. Medium and high quality printed characters have their: $\xi \to 1$ ($\delta_1 \to 0$); whereas, low quality characters have their: $\xi \to 0$ ($\delta_2 \to \delta_1$).

6. CSM net design

The CSM net is a new neural architecture network based on the combination of two neural networks, namely, SMNN and the WTA subnets. The use of one of these two subnets is based on the comparison of a relative distance ($\xi$) to a rejection threshold ($R_c$) to achieve higher recognition rates. The main difference between the CSM net and the Hamming net is the use of SMNN instead of the Hamming subnet, with suitable rejection threshold for the SMNN subnet as depicted in Fig. 2. However, the synaptic weights $W_{CSM}$ are real valued, and correspond to the stored patterns $S$, which are modified versions of the binary model patterns $y$, whereas the $W_{H}$ are bipolar, and correspond to the stored model patterns $y$. Furthermore, the CSM net is based on the use of the similarity measure vector $\xi$ (see Eq. (23)), calculated from the similarity measure vector $S_c$ produced by the inner product: $W_{CSM} \times x^t$. Thus, the similarity layer can accept a sample pattern $x$ in the case of $\xi$ being greater or equal to a predefined rejection threshold $R_c$. In the case of acceptance, the class membership is simply established according to the highest score by:

$$C = \arg\max_{m=1,2,...,P} \left\{ S_c^m \right\}$$

(24)

But in the case of rejection, which means that the relative distance is less than $R_c$, each similarity vector $S_e$ is normalized to a unit power:

$$S_c^{\text{norm}} = \frac{S_c}{||S_c||}$$

(25)

where $||S_c||$ is the norm of the $S_c$. Thus, the given pattern $x$ is rejected by the SMNN layer, and all the related similarities $S_c^{\text{norm}}$ are used by the subsequent competitive layer, namely, the WTA network for classification. The normalization of $S_c$ reduces the sensitivity to different scanners gains (image-to-contrast) as well as to different toners darkness (shade of ink) [23]. The competition between neurons in the WTA network is based on calculating the scalar product between input vectors $S_c^{\text{norm}}$ and synaptic weight vector $W$:

$$O = W \cdot S_c^{\text{norm}} = \sum_{m=1}^{P} W_{ij} \cdot S_c^{\text{norm}}$$

(26)

After calculation of the WTA outputs $O_1, O_2, \ldots, O_n, \ldots, O_P$, ($0 \leq O_i \leq 1$), the class membership is simply established by:
\[ C = \arg \max_{m=1,2,...,P} \{ O_m \} \]  

(27)

Highly and severely degraded characters are always rejected by the SMNN subnet due to their very small relative distance \( \xi \), and consequently in most cases, confused by the WTA classifier. Thus, a lower bound rejection threshold \( R_{LB} \) is introduced in order to avoid this misclassification and hence increases the reliability of the proposed method. For a given character, the relative distance is compared to a first rejection threshold \( R_C \) and then to a second rejection threshold \( R_{LB} \) (see Fig. 2) as follows:

If \( \xi \geq R_C \); process with SMNN subnet.

If \( \xi \geq R_{LB} \); process with WTA.

Rejection of the character by SMNN.

The value of \( R_{LB} \) was determined experimentally for various values of \( R_C \in [0.002,0.024] \).

7. Experimental results

The experimental part is conducted into two subparts; the first one consists in recognizing the ACN where the number of classes is equal to 13 (see Fig. 14(a)). The classes considered consist in ten numeral characters, and three alphabetic characters. The first subpart which concerns the ACN recognition is the main targeted application of this work due to its extreme needs in bank check processing. The second experimental subpart was added in order to show the generalization of the proposed method on data having a higher number of classes. This subpart is an extension of the first one to include twenty two other alphabetic characters from the customer first and last name, as shown in Fig. 14. Thus, the classes considered consist in thirty five; ten numerals and twenty five alphabetic characters. The letters QVWX are omitted because this experimental works was conducted exclusively with Latin transcriptions of Arabic names which do not use these letters. To show the high performance of the proposed approach (CSM net), it is also compared to seven other classifiers, namely, the enhancement of degraded document images (EDDI) method, the highly degraded printed character method (HDPP), the CSM-MLP combination method, the dynamic Bayesian networks (DBN), the gradient pattern recognition (AGP), the sequential combination of context-based classifiers (SCCC) [24] and the Hopfield MLP combination method (Hopf-MLP) [5]. Because the HDPP method is applied in this work to only isolated printed characters, it is implemented without using the blind deconvolution and MRF-based segmentation techniques that are usually used for segmentation problems.

7.1. Data extraction and training phase

The characters data set represents a two fonts grey level characters of size 40-by-40 that were collected from 490 post checks belonging to the same post office. The post checks were scanned at 300 dpi, which avoids the scaling procedure. Hence a frame of 40-by-40 pixels is sufficient to include each sample character.

The SMNN and Hamming subnets used 130 and 350 preprocessed image characters (10 samples from each class) for their memories, or synaptic weights, in the first and second experimental subparts respectively. The parameters of SMNN and the Hamming similarity layers were fixed to: \( P_1 = 130, P_2 = 350 \) and \( n = 1600 \) (40-by-40 pixels), where \( P_1 \) and \( P_2 \) are the numbers of memorized patterns for the first and second experimental subparts. The SMNN parameters \( \beta \) and the model densities \( \beta \) for 13 and 22 models of Fig. 14 are given below.

\[
\begin{align*}
\beta_0 &= 0.33, \beta_1 = 0.21, \beta_2 = 0.29, \beta_3 = 0.30, \beta_4 = 0.29, \beta_5 = 0.31, \\
\beta_6 &= 0.30, \beta_7 = 0.21, \beta_8 = 0.29, \beta_9 = 0.32, \beta_C = 0.34, \beta_E = 0.35 \text{ and } \beta_C = 0.23.
\end{align*}
\]

\[
\begin{align*}
\beta_0 &= 0.0028, \beta_1 = 0.0037, \beta_2 = 0.0031, \beta_3 = 0.0029, \beta_4 = 0.0030, \\
\beta_5 &= 0.0029, \beta_6 = 0.0030, \beta_7 = 0.0037, \beta_8 = 0.0030, \beta_9 = 0.0029, \\
\beta_C &= 0.0028, \beta_E = 0.0027 \text{ and } \beta_C = 0.0035.
\end{align*}
\]

The characters used for learning for the second subnet, or competitive layer, were extracted from 70 post checks, and gave rise to 1960 and 3010 characters for the first and second experimental subparts respectively. For both the CSM net and the Hamming net, this subnet is a competitive neural network based on the WTA. The WTA training parameter \( \varphi \) was set to 0.5, and the iteration number \( T \) was set according to experimental results as it will be presented in the next subsection. The WTA connection weights \( w_i \) are obtained after convergences, and stored for the recognition process. Before they are fed to the CSM net or to the Hamming net, the characters are filtered and binarized. The remaining 420 post checks were used for the testing, which include 11,760 and 17,640 characters that were considered for the first and second experimental subparts respectively.

7.2. Recognition phase

7.2.1. ACN characters recognition

Fig. 15 shows the recognition plots of the CSM net, and the Hamming net at different iteration numbers \( T \), in a “complete rejection mode”. We mean by a complete rejection mode, all the patterns are rejected by the first subnet or similarity layer, namely, SMNN and Hamming subnets, and classified only by WTA-based classifier. The iteration number \( T \) is chosen to give the best performance, and since, as it can be noticed, the recognition rate is maximum at \( T = 2 \) and \( T = 3 \) for CSM net and Hamming net respectively. Fig. 15 shows also the high performance of the CSM net over the Hamming net for all the iteration numbers \( T \). The error-reject plots of the SMNN and Hamming subnets for different values of \( R_C \) (0.1 \( \leq R_C \leq 0.9 \)) and \( R_H \) (0.1 \( \leq R_H \leq 0.9 \)) respectively, are given in Fig. 16. In this figure, the error rate falls for higher \( R_C \) and \( R_H \). The best performance is achieved for lower values of \( R_C \) and \( R_H \). According to Fig. 16, the SMNN subnet outperforms clearly the Hamming subnet for lower values of \( R_C \) (\( R_C \leq 0.4 \)).

Fig. 17 presents the plots of recognition rate for SMNN and Hamming subnets vs their error rates. The recognition rate is proportional to the error rates for both subnets, due to the rejection rate which is inversely proportional to the error rate. According to Fig. 17, the SMNN subnet outperforms clearly the Hamming

![Fig. 14. Preprocessed character samples (a) account check number characters (b) customer first and last name characters.](http://iranpaper.ir)
subnet in term of recognition rate for higher error rates (error > 0.016), corresponding to the lower rejection threshold $R_C (R_C < 0.4)$.

According to Fig. 18, the Hamming net, that combines Hamming subnet and WTA-based classifier, outperforms clearly the Hamming subnet for lower, medium, and particularly for higher rejection threshold values. The curve describing the combination of Hamming subnet and WTA is obtained by choosing the iteration number $T = 3$ and various $R_H (0.1 \leq R_C \leq 0.9)$. In Fig. 19, the proposed method, namely, CSM net that combines SMNN subnet and WTA-based classifier, outperforms clearly the SMNN subnet for lower, medium, and particularly for higher rejection threshold values. The curve describing the combination of SMNN and WTA is obtained by choosing the iteration number: $T = 2$, and various $R_C (0.1 \leq R_C \leq 0.9)$. This combination (CSM net) tends, in reality, to boost the performance of the SMNN classifier. The result is a robust multiple classifier for all the rejection threshold $R_C$. Tables 2 and 3 summarizes the results of combining SMNN and Hamming subnets with the WTA network. This combination improves both the multiple classifiers, namely, the CSM net and the Hamming net. However with the SMNN–WTA combination (CSM net) we obtain higher performances, that are better than Hamming-WTA (Hamming net) ones as shown in Table 4. The plots of Fig. 20 show the high performance of the CSM net over the Hamming net for all the rejection threshold ($R_H$ and $R_C$). Table 4 summarizes the best performances achieved by the four classifiers, namely, the SMNN subnet, the CSM net, the Hamming subnet, and the Hamming net for $R_H = 0.1$ and $R_C = 0.1$, and $R_B = 0.014$. The extremely small difference in recognition rate between CSM net and Hamming net, becomes progressively important when the number of the data set characters increases. A difference in sensitivity is of 0.48% [99.28 (CSM net)–98.80 (Hamming net)] when applied, for example, to 600 post checks, corresponds to about 18,000 characters (30 characters/check); this results in 86.4 (18,000 x 0.48%) recognized characters, more than the Hamming net. This value could correspond to a gain difference of 86.4/600 = 0.48% in check recognition rate. Hence this difference in character recognition rate could be mapped, with a considerable check recognition rate progress interval as follows:

![Fig. 15. CSM net and Hamming net recognition results in rejection mode at different values of T.](image1)

![Fig. 16. Error vs reject plots for the two similarity subnets; SMNN and Hamming using various rejection thresholds values for $R_C$ and $R_H$.](image2)

![Fig. 17. Recognition vs error plots for the two similarity subnets; SMNN and Hamming using various rejection thresholds values for $R_C$ and $R_H$.](image3)

![Fig. 18. Recognition results of the Hamming subnet and Hamming net for various rejection thresholds $R_H$.](image4)
0.48% (gain in character recognition rate)

→ [0.48, 14.4]% (gain in check recognition rate)

The values of: 0.48% and 14.4%, represent the best and the worst gains in check recognition rates, respectively. In practice, the value of 0.48% is far from being reachable, because the system cannot make errors on all the characters of the same check (30 errors per check).

If we assume that our test set is composed of high quality characters, the CSM net and the Hamming net would give certainly the same recognition rate. However with a low quality test set the recognition rate of the CSM would be higher than the one of Hamming net. The percentage of low quality characters in the test set determines clearly the difference in the recognition rates between the two systems. The higher is this percentage, the higher is the difference in the recognition rate. This confirms that our database includes degraded characters, but not enough to give a big difference in the recognition rates. This point will be considered next, in the application of this work on the ACN and the alphabetic characters, that represent the customer first and last names.

### ACN and alphabetic characters recognition

Fig. 21 shows the recognition plots of the CSM net, and the Hamming net, at different iteration numbers \( T \) in a complete rejection mode. According to this figure the recognition rate is maximum at \( T = 6 \) and \( T = 4 \) for CSM net and Hamming net respectively. It shows also the high performance of the CSM net over the Hamming net for all the iteration numbers \( T \). The error-reject plots of the SMNN and Hamming subnets for different values of \( R_C = 0.1 \) and \( R_R = 0.014 \), respectively, are given in Fig. 22. In this figure, the error rate falls for higher \( R_C \) and \( R_R \). The best performance is achieved for the lower values of \( R_C \) and \( R_R \). According to Fig. 22 the SMNN subnet outperforms clearly the Hamming subnet for all the values of \( R_C \) (\( R_R \) > 0.1). Fig. 23 presents the plots of recognition rates for SMNN and Hamming subnets vs their error rates. From this figure, the SMNN subnet outperforms clearly the Hamming subnet in term of recognition rate for the higher error rate corresponding to the higher rejection threshold \( R_T \) (\( R_T \) > 0.1). The plots of Fig. 24 show the high performance of the CSM net over the Hamming net for all the rejection threshold (\( R_R \) and \( R_T \)).

Table 5 summarizes the best performances achieved by the four classifiers, namely, the SMNN subnet, the CSM net, the Hamming

![Recognition results of the SMNN subnet and CSM net for various rejection thresholds](image1)

**Fig. 19.** Recognition results of the SMNN subnet and CSM net for various rejection thresholds \( R_C \).

**Fig. 20.** Recognition results comparison for the CSM net and the Hamming net for various rejection thresholds \( R_C \) and \( R_R \).

<table>
<thead>
<tr>
<th>Classifiers</th>
<th>Reject (%)</th>
<th>Error (%)</th>
<th>Reliability (%)</th>
<th>Recognition (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hamming subnet (( R_R = 0.1 ))</td>
<td>1.65</td>
<td>0.35</td>
<td>99.65</td>
<td>98.00</td>
</tr>
<tr>
<td>SMNN (( R_C = 0.1 ))</td>
<td>1.24</td>
<td>0.17</td>
<td>99.83</td>
<td>98.59</td>
</tr>
<tr>
<td>Hamming net (( R_R = 0.1 ))</td>
<td>0</td>
<td>1.20</td>
<td>98.80</td>
<td>98.80</td>
</tr>
<tr>
<td>CSM net (( R_C = 0.1 ), ( R_R = 0.014 ))</td>
<td>0.24</td>
<td>0.48</td>
<td>99.52</td>
<td>99.28</td>
</tr>
</tbody>
</table>

**Table 4** Best performance for the four classifiers applied to ACN characters (\( R_C = 0.1 \) and \( R_R = 0.014 \), and \( R_T = 0.1 \)).

![Recognition plots of the SMNN and Hamming subnets for different values](image2)
subnet and the Hamming net for \( RH = 0.1 \) and \( RC = 0.1 \), and \( RLB = 0.014 \). However with the CSM net, we obtain higher performances than the Hamming net, and better results than those given in Table 4. The difference in recognition rate between CSM net and

![Fig. 21. CSM net and Hamming net recognition results in rejection mode at different values of \( T \).](image)

![Fig. 22. Error vs reject plots for the two similarity subnets; Hamming and SMNN using various rejection thresholds values for \( RC \) and \( RH \).](image)

![Fig. 23. Recognition vs error plots for the two similarity subnets; Hamming and SMNN using various rejection thresholds values for \( RC \) and \( RH \).](image)

![Fig. 24. Recognition results comparison for CSM net and Hamming net for various rejection thresholds \( RC \) and \( RH \).](image)

![Fig. 25. Reject results comparison of the first and second experimental subparts for various rejection thresholds \( RC \).](image)

![Table 5. Best performance for the four classifiers applied to ACN and alphabetic characters \((RH = 0.1 \text{ and } RLB = 0.014, \text{ and } RH = 0.1)\).](image)
Hamming net is greater than the one obtained in the first experimental subpart. This difference is quite evident, because the second test set has more degraded characters than the first one. This could be noticed from the plots of reject vs rejection threshold ($R_C$) of Fig. 25. From this figure, the SMNN subnet, when applied to the second test set, rejects more characters, in term of percentage, than in the first test set. This confirms clearly, that in the second test set, there are more degraded characters than in the first one. Hence, by using this test set, the difference in recognition rate between the CSM net and the Hamming net is greater than the one of the first experiment. This support and confirms clearly the statement of the last paragraph of the Section 7.2.1.

To test the robustness of the proposed method with poor quality characters, the proposed method was first applied to the training set (350 models) represented by its classes shown in Fig. 14, corrupted with impulse noise, then to real, broken and incomplete characters. The training set was corrupted with salt and pepper impulse noise, and a recognition rate of 100% was achieved for 200/1600 = 12.5% (200 pixels altered from 1600), and for 400/1600 = 25.0%, for both CSM net and Hamming net as shown in Fig. 26. It can be noticed from this figure that for impulse noise higher than 25%, the CSM net maintains the recognition rate at 100% and decreases slightly to 95.43% for 75% of impulse noise. Whereas, in the Hamming net, the recognition rate falls rapidly beyond the value of 25% of impulse noise, and reaches the value of 28.57% at 75% of impulse noise. It can be concluded from these tests, that for an impulse noise up to 62.5%, the 350 noisy characters are successfully recognized by the CSM net, as it can be seen in Table 6.

Experiments were also conducted on real, broken and incomplete characters images. Fig. 27(a) shows many of these characters, and in spite of the important degradation of the characters, the proposed method was able to recognize even highly degraded printed characters. Fig. 27(b) shows also samples of the degraded characters that were confused by the proposed method; their recognized class is shown on the right of each character image. From Fig. 27(c) we can see the rejected characters ($\xi < R_C = 0.1$) and their corresponding relative distances below each character. These characters were correctly recognized by the WTA-based classifier, showing their class on the right of each character. Fig. 27(d) shows characters that were confused by the SMNN subnet, but fortunately rejected due to their relative distance which is less than $R_C = 0.1$. The confused class is given by the first letter on the right of each character. The WTA subnet was able to recognize these characters and the correct class is given by the second letter on
the right of each character. The characters presented in Fig. 14 were manually corrupted, as shown in Fig. 28. These highly degraded characters were correctly recognized by the proposed method, which confirms clearly the robustness of the CSM net for the recognition of degraded characters. Their correct class is shown below each character image.

To show the high performance of the CSM net, it is compared to seven other existing character recognition methods. Fig. 29 summarizes the results obtained on the data set using eight classifiers for degraded characters recognition. The parameters corresponding to these classifiers were chosen to give the highest recognition rate. For the proposed method we chose $R_C = 0.1$, $R_{lb} = 0.014$ and $T = 6$. The parameter of the CSM-MLP method was set to $R_M = 0.8$ using the centroid dithering. The parameters of the SCC method were set to $R_N = 0.8$ and $R_S = 0.4$, and the parameters of the Hopf-MLP method were set to $R_H = 0.58$ and $R_M = 0.50$. As can

Fig. 28. Manually corrupted characters correctly recognized by the CSM net.

Fig. 29. Error vs reject showing comparison results of eight classifiers.

Fig. 30. Recognition vs error showing comparison results of eight classifiers.

Table 7

Best performance comparisons of the eight classifiers: the CSM net that constitutes the proposed work, the EDDI, the HDPD, the CSM-MLP, the DBN, AGP, SCCC and Hopf-MLP. The eight classifiers were tested using the 17640 printed characters.

<table>
<thead>
<tr>
<th>Classifiers</th>
<th>Reject (%)</th>
<th>Error (%)</th>
<th>Reliability (%)</th>
<th>Recognition (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSM net ($R_C = 0.1$, $R_{lb} = 0.014$)</td>
<td>0.15</td>
<td>0.39</td>
<td>99.61</td>
<td>99.46</td>
</tr>
<tr>
<td>Hopf-MLP</td>
<td>0.20</td>
<td>0.74</td>
<td>99.26</td>
<td>99.06</td>
</tr>
<tr>
<td>SCCC</td>
<td>0.21</td>
<td>0.79</td>
<td>99.21</td>
<td>99.00</td>
</tr>
<tr>
<td>EDDI</td>
<td>0.30</td>
<td>0.78</td>
<td>99.22</td>
<td>98.92</td>
</tr>
<tr>
<td>HDPD</td>
<td>0.45</td>
<td>0.85</td>
<td>99.15</td>
<td>98.70</td>
</tr>
<tr>
<td>CSM-MLP</td>
<td>0.33</td>
<td>0.65</td>
<td>99.35</td>
<td>99.02</td>
</tr>
<tr>
<td>DBN</td>
<td>0.28</td>
<td>0.75</td>
<td>99.25</td>
<td>98.97</td>
</tr>
<tr>
<td>AGP</td>
<td>0.50</td>
<td>0.73</td>
<td>99.27</td>
<td>98.77</td>
</tr>
</tbody>
</table>


be seen in Fig. 29, the proposed method clearly outperforms the seven other classifiers. The plots corresponding to the EDDI, the HDPD, the CSM-MLP, DBN, the AGP, the SCC and the Hopf-MLP almost reach the error rate value of the proposed method for very high rejection rates. On the other hand, for low rejection rates, the combined approach performs better than all these classifiers. The recognition-error plot of Fig. 30 shows clearly the high performance achieved by the CSM net compared to these classifiers. Our approach clearly outperforms the other classifiers for very low and high error rates, and still has better performance for medium error rates. This can be explained by the fact that the combined method accepts all the incoming patterns that are recognizable than the mentioned classifiers for the same error rates, which results in higher recognition rates. Table 7 summarizes the recognition results achieved by the eight classifiers, namely, the CSM net, EDDI, HDPD, CSM-MLP, DBN, AGP, SCC and Hopf-MLP. From Table 7, we can clearly see that the results obtained from our method are much better than those given by all the seven other classifiers in terms of reliability and recognition rate.

A standard reference in term of signal to noise ratio (SNR) is provided, in order to stabilize the amount of noise tolerated for the recognition of degraded characters with the CSM net. This was achieved using various additive Gaussian noise (AGN) levels applied to the set of account and alphabetic printed characters presented in Fig. 14. Fig. 31 shows the corruption of the sample characters “0”, “4” and “R” with AGNs from 4 to −14 dB of SNR by step of 2 dB, and their corresponding quality parameters $\xi$, below each character image. The plot of Fig. 32 shows the SNR for each noisy sample character correctly recognized by the CSM net. In spite of the highly degradation that affects the characters, our proposed method was able to recognize this type of printed characters. This is due to the appropriate combination, that is based on the right choice of the two subnets, namely, the SMNN and the WTA subnets, with efficient rejection thresholds $R_C$ and $R_{LB}$. It can be concluded from these results, that for a SNR up to $-8$ dB ($\sigma = 2.51$), the 35 noisy characters are successfully recognized as this can be seen in Table 8.

### Table 8

<table>
<thead>
<tr>
<th>SNR (db)</th>
<th>Reject (%)</th>
<th>Error (%)</th>
<th>Reliability (%)</th>
<th>Recognition (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>−2</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>−4</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>−6</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>−8</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>−10</td>
<td>1</td>
<td>2</td>
<td>94.28</td>
<td>91.43</td>
</tr>
<tr>
<td>−12</td>
<td>3</td>
<td>4</td>
<td>88.57</td>
<td>80.00</td>
</tr>
<tr>
<td>−14</td>
<td>7</td>
<td>3</td>
<td>91.43</td>
<td>71.43</td>
</tr>
</tbody>
</table>

8. Conclusion

For characters recognitions we have presented the cascaded combination architecture of two noise insensitive neural networks, namely, the SMNN and the WTA subnets. This combination gives a strong noise robustness to the tasks supported. The CSM method is analyzed and presented in the form of a feed forward network neural network (SMNN), used as a memory network. A relative distance is introduced, as a quality measurement tool, to measure the quality of the degraded characters in order to accept or reject the incoming characters. Experiments were conducted on isolated printed characters collected from post checks, and a comparative study of our method was achieved with individual classifiers and Hamming net, on one hand, and with seven other classifiers found in the literature, on the other hand. It was demonstrated that the CSM net outperforms all the other classifiers. Experiments were also conducted on noisy images for various impulsive noises rates and real, broken and incomplete characters, it shows an effective robustness. It shows also clearly that our recognition system produces promising results for even highly degraded printed characters recognitions. A standard reference, in term of signal noise ratio (SNR) for the recognition of degraded characters with the
CSM net, was provided, to estimate the amount of additive noise that could be processed by our system. The proposed CSM net could be also addressed to solve the problem of multi font recognitions by increasing the number of classes.

Acknowledgments

The authors thank the anonymous reviewers whose valuable and consistent comments along the revision process made this manuscript more useful and clear.

References