A NEURO-FUZZY GRAPHIC OBJECT CLASSIFIER WITH MODIFIED DISTANCE MEASURE ESTIMATOR

R. A. ALIEV, B. G. GUIRIMOV AND R. R. ALIEV

ABSTRACT. The paper analyses issues leading to errors in graphic object classifiers. The distance measures suggested in literature and used as a basis in traditional, fuzzy, and Neuro-Fuzzy classifiers are found to be not suitable for classification of non-stylized or fuzzy objects in which the features of classes are much more difficult to recognize because of significant uncertainties in their location and gray-levels. The authors suggest a neuro-fuzzy graphic object classifier with modified distance measure that gives better performance indices than systems based on traditional ordinary and cumulative distance measures. Simulation has shown that the quality of recognition significantly improves when using the suggested method.

1. The State of the Problem: Overview of Existing Systems

The problem of optical recognition is not new – it has existed for decades. There are many good results and many theories have been developed [1]. Many examples of applications are known – extending from automated digital conversion of handwriting used in banks and offices to vision systems used in robotics and space systems, and diagnostic and expert systems for micro-biology [2,3]. Despite significant successes in the development of artificial intelligence systems, the optical recognition of fuzzy graphic objects still presents significant difficulties to machines. The human way of solving this problem, which seems not to be a problem at all, is very difficult to discover and mimic in artificial brains.

Let's give a simple definition and state in general the classification problem.

Definition 1. Let $X = (x_1, x_2, ..., x_N)$ be a graphic object described by N features ($X \in \Re^N$, a space of object features) and $C = (c_1, c_2, ..., c_S)$ be a set of S classes. Then a classifier is a function $F_{class}(X) : \Re^N \to C$. The classifier partitions the feature space into S mutually exclusive areas.

It is not surprising that the quality of classification depends on how well we can compare any two different objects. We call the tool used to compare graphic objects a Distance Measure Estimator (DME). The DME estimates the distance between (or similarity degree of) two objects. After the training of the classifier, its knowledge base KB (usually based on Neural Networks, Fuzzy Rules, or a combined Neuro-Fuzzy model) stores knowledge of the relations between the classes and the features. For every class there is a set of associated features in KB that uniquely identify the class. Then the classifier can make a decision about the class of the input object based

Invited Paper: Received in December 2003
Key words and phrases : Neuro-Fuzzy technology, Fuzzy logic, IF-THEN rules, Neural Network

on the value of some kind of DME. After the distance values between the input object and every class contained in the KB have been calculated, the class that produces the minimum value of distance with the input object is considered the best match.

To classify fuzzy objects based on DME, we have to implement some metric that will measure normalized fuzzy distance between instance vectors. The most widely used distance measures are the Euclidean and Manhattan metrics [4]:

$$d(x,y)=\sqrt{\sum_{i=1}^{N}(x_i-y_i)^2}\ ; \tag{1}$$

$$d(x,y)=\sum_{i=1}^{N}|x_i-y_i|\ ; \tag{2}$$

where $X=(x_1,x_2,...,x_N)$ and $Y=(y_1,y_2,...y_N)$ are N-dimensional fuzzy feature vectors of two compared objects.

There are also a lot of modifications of traditional metrics [5]. Depending on the type of classes and their features, different metrics should be used.

Since both the Euclidean and Manhattan distances are calculated separately for each dimension (standing for a specific feature), they are not very good distance measures for similarity between objects that have correlated and ordered features [4].

Reference [4] introduces cumulative distance measures for such cases. The cumulative Euclidean distance measure is as follows:

$$d(x,y)=\sqrt{\sum_{i=1}^{N}\left(\sum_{u=1}^{i}x_u-\sum_{u=1}^{i}y_u\right)^2}\ . \tag{3}$$

The cumulative Manhattan distance measure, called also Landmover distance, is calculated as follows:

$$d(x,y)=\sum_{i=1}^{N}\left|\sum_{u=1}^{i}x_u-\sum_{u=1}^{i}y_u\right|. \tag{4}$$

The drawback of the suggested measures is that the features are not considered equal. As can be seen, the lower number features count more in the final value than the higher number ones. This leads to unreasonable results when features are weakly related or not unreasonable equally important [4].

An improvement for the distance measure in case of not equally important and fuzzy features is to attach a weight parameter to each feature:

$$d(x,y)=\sum_{i=1}^{N}|x_i-y_i|\,w_i \tag{5}$$

and to apply an *equivalent* feature aggregation as in:

$$d(x,y)=\frac{1}{N!}\sum_{u=1}^{N!}\sum_{i=1}^{N}\left|\sum_{v=1}^{i}x_{r_{uv}}-\sum_{v=1}^{i}y_{r_{uv}}\right|, \tag{6}$$

where $R=\left\{(r_{uv}),u=\overline{1,N!},v=\overline{1,N}\right\}$ is the matrix the rows ($u=\overline{1,N!}$) of which are all possible combinations of the series $\{1,2,...,N\}$.

Another problem with classifiers, especially the classifiers implemented on the base of neural networks, is that following the traditional *definition 1*, the classifier partitions the feature space into *S* mutually exclusive areas. In many applications, especially, in case of fuzzy objects, the definition of the feature space is rarely complete. Also the boundaries of the partitions may partially overlap.

Disabling mandatory classification of an object into any class, i.e. applying the so-called distance reject, introduced in [5], can significantly reduce the misclassification risk. In traditional neural networks with the "winner takes all" principle there is no way to reject the input object that does not belong to the allowed object space. For example, the neural network classifier taught to recognize the 26 English letters will classify into the same class set all figures and symbols from Russian and Greek character sets. This is because a neural network's learning set contains only allowable class samples and there is no way to provide any disabled samples because this would make the learning set infinite.

Fuzzy classifiers can provide a more trustworthy distance measure that would allow more quality recognition. Allowing human expertise with viewable and editable rules defining classes, is essential for effective and rapid learning (knowledge one-step conversion) as well as human-like performance. However it would be more loss than profit to ignore the advantages of learning from experience, excellent performance, and the generalization ability all intrinsic to neural networks. The best solution is to mix these two techniques in the most effective way – in a Neuro-Fuzzy classifier [9,10].

2. Neuro-Fuzzy Model: The Fuzzy Rules Representation

We define a neuro-fuzzy system as an interrelated and mutually complementing intelligent combination of neural and fuzzy subsystems. Here the neural subsystem performs, primarily, functions of learning, tuning, and optimization of the parameters of the fuzzy subsystem. The parameters of the fuzzy subsystem are datasets able to represent the fuzzy rules defining the overall system operation. The model is designed such that it supports both fuzzy IF-THEN rules and connectionist-based representations via some explicit conversion procedure that regenerates them from the unique system knowledge base. Such a neuro-fuzzy system can be easily applied to develop graphic object classifiers.

The neuro-fuzzy model considered below was first introduced in [6] and then improved in [7].

Assume that in the environment under consideration we have *N* features to define *M* classes of graphic objects:

$$E = \{C_1, C_2, ..., C_M\}$$

Also for simplicity we suppose we can use only one fuzzy rule to describe every class. So the number of rules is also *M*.

The fuzzy rules for the classification of graphic objects are represented as follows [7]:

IF F_{11}/w_{11} AND F_{12}/w_{12} AND … AND F_{1N}/w_{1N}
THEN $O = C_1$;
IF F_{21}/w_{21} AND F_{22}/w_{22} AND … AND F_{2N}/w_{2N}

$$
\begin{aligned}
&\text{THEN } O = C_2 ;\\
&\qquad\qquad \ldots \ldots \ldots\\
&\text{IF } F_{i1} / w_{i1} \text{ AND } F_{i2} / w_{i2} \text{ AND} \ldots \text{AND } F_{iN} / w_{iN}\\
&\text{THEN } O = C_i ;\\
&\qquad\qquad \ldots \ldots \ldots\\
&\text{IF } F_{M1} / w_{M1} \text{ AND } F_{M2} / w_{M2} \text{ AND} \ldots \text{AND } F_{MN} / w_{MN}\\
&\text{THEN } O = C_M ,
\end{aligned}
\tag{7}
$$

where $F_{ij} := (f_j \;\; IS \;\; x_{ij})$, we denote a term expressing the degree of presence in an object O of some feature (f_j) which makes it be interpreted as the class labeled by index i (C_i); x_{ij} is the constant fuzzy value from the set of features $X = \{x_1, x_2, ..., x_m\}$ used to define the value of feature f_j in the class C_i. F_{ij} / w_{ij} means that an object of class i should have the feature f_j with importance w_{iN}. (For example, a feature of graphic symbol 'A' might be: "the associated objects have a horizontal bar of dark color with the length of about a half the object's width, located somewhere in the middle of the object's height, approximately the same distances from the left and right sides.") .

The feature weights w_{ij} are intended to consider different importance of particular features in recognition of different classes. This helps significantly improve quality in case the environment contains classes with similar pictures having slight distinctions disseminated over small regions. For example, in Fig. 1 the circled areas (i.e. the related features) must be paid more attention than others when classifying the object into 'C' or 'O'.

Figure 1. Areas with higher importance (circled)

Note that a feature is a combination of fuzzy constants expressing both locations and gray-levels (or colors). The same feature can be used in description of more than one class. The minimization of the number of features is part of the learning procedure and may be required in order to optimize the system knowledge base and to improve overall performance and interpretability.

A fuzzy constant defining the location has a membership function $\mu_D(d_x, d_y)$ of two orthogonal coordinates (Fig. 2, only one coordinate shown).

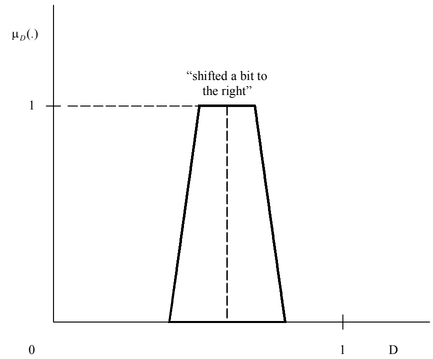


Figure 2. A location constant

A gray-level constant for a fixed location (d_X, d_Y) is represented by a membership function $\mu_G(g)$ of an argument g representing the gray-level normalized into [0,1] (Fig. 3).

The truth value to which a particular feature i exists in the input object, T_{fi}, is calculated as:

(8) $$T_{fi} = \max_{(d_x, d_y)}\{\min_2\{\mu_{Di}(d_x, d_y), \mu_{Gi}(g[d_x, d_y])\}\}\ .$$

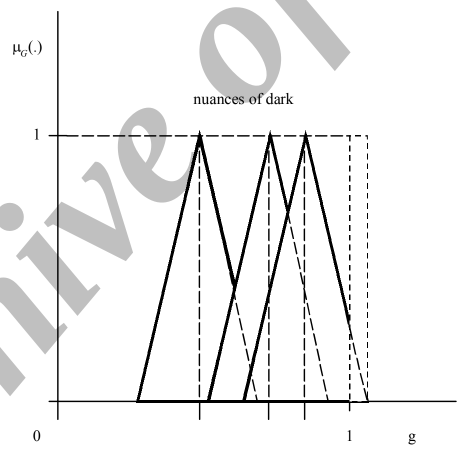


Figure 3. Constants of gray-level

The truth values of all features are aggregated to obtain an overall truth value using a kind of t-norm. The appropriate choice of this t-norm is very crucial for quality classification. As can be seen, this aggregation (conjunction) operation is closely related in the meaning to the distance measure concept discussed in the first section. We will show in the next section how we implement the DME for a Neuro-Fuzzy classifier.

The output value O, determining the classified object, obtained from all rules is represented as a fuzzy set:

(9) $$O = \{C_1/T_1, C_2/T_2, ..., C_i/T_i, ..., C_M/T_M\},$$

where T_i is the calculated truth value of the object to be treated as class C_i.

The final goal is to obtain the appropriate class label. We mentioned above that the classifier must reject the objects having no associated class in the considered environment.

(10) $$C = O_{defuz} = \begin{cases} L\left(\arg\max_i \{T_i\}\right), & \textit{if } \max_i \{T_i\} > T_{\min} \\ \textit{rejected}, & \textit{elsewise} \end{cases}$$

where $T_{\min}$ is the minimum allowable truth value for classes in the environment E and $L(c)$ is the label for the class with the index c.

3. Neuro-Fuzzy Model: The Network Representation

The connectionist structure of the proposed Neuro-Fuzzy classifier is shown in Fig. 4. The network consists of six layers. Each layer performs different functions and uses specific processing units (neurons).

The first layer is the receptive field. It just receives a single object in 2-color binary matrix form.

In the second layer the locations (regions) responsible for particular features are retrieved. Each neuron in the second layer is related to a particular location constant. The output of these neurons contains information about the gray-level inside the region defined by the constant.

The third layer neurons have accumulated in their parameters the information about the features used to classify the objects. These neurons actually implement the DME function (6) where instead of membership degrees we use error (or mismatch) values:

$$E = w_{(i,j)} \min_{(D_x, D_y)} \left\{\max_2 \left\{\varepsilon_{\min(i,j)}(D_x, D_y), \varepsilon_{f(i,j)}(D_x, D_y)\right\}\right\}$$

$$d(x,y) = \frac{1}{N!} \sum_{u=1}^{N!} \sum_{i=1}^{N} \sum_{v=1}^{i} E_{r_{uv}} .$$

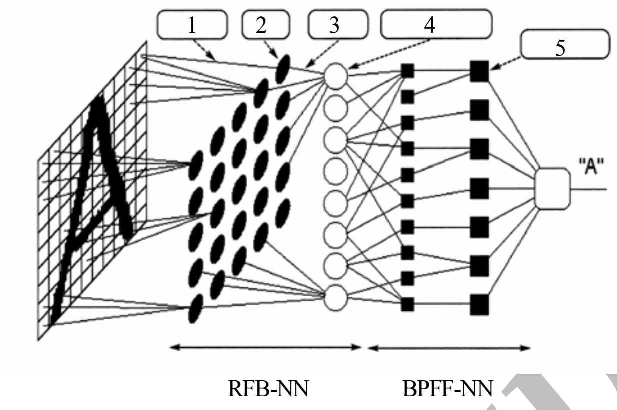


Fig. 4. The Neuro-Fuzzy classifier (1 – location constants; 2 – gray-level values; 3 – gray-level constants; 4 – overall mismatch; 5 – final class mismatch)

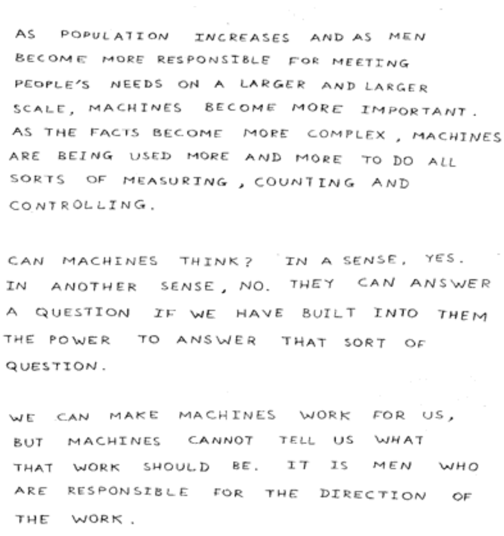
A neuron in this layer outputs the distance (mismatch) value between the input object and the class the neuron is related to. The remaining layers are for the purpose of defuzzification and are implemented as a Back-Propagation Feed-Forward Neural Network (BPFF-NN).

A more detailed description of the neural network and learning algorithms can be found in [7,8]. Here we just note that the learning is done by both supervised and unsupervised algorithms. The proposed neuro-fuzzy system also supports human expertise, revision and one-step knowledge conversion.

4. Software Implementation and Simulation Results

On the basis of the suggested Neuro-Fuzzy classification system a software package has been developed [7,8]. The software system was intended for recognition of human handwriting characters. A sample text is shown in Fig. 5. In our further modification of this software we tried the modified DME suggested in this paper (6).

We have compared the performance indices of several recognition systems on the sample text (Fig. 5). Figure 6 show the output texts from 3 widely used recognition application software (a – Fine Reader Version 4; b – Fine Reader Version 6; c – HP Precision Scan). Figure 7 shows the system's output screen.


AS POPULATION INCREASES AND AS MEN
BECOME MORE RESPONSIBLE FOR MEETING
PEOPLE'S NEEDS ON A LARGER AND LARGER
SCALE, MACHINES BECOME MORE IMPORTANT.
AS THE FACTS BECOME MORE COMPLEX, MACHINES
ARE BEING USED MORE AND MORE TO DO ALL
SORTS OF MEASURING, COUNTING AND
CONTROLLING.

CAN MACHINES THINK? IN A SENSE, YES.
IN ANOTHER SENSE, NO. THEY CAN ANSWER
A QUESTION IF WE HAVE BUILT INTO THEM
THE POWER TO ANSWER THAT SORT OF
QUESTION.

WE CAN MAKE MACHINES WORK FOR US,
BUT MACHINES CANNOT TELL US WHAT
THAT WORK SHOULD BE. IT IS MEN WHO
ARE RESPONSIBLE FOR THE DIRECTION OF
THE WORK.

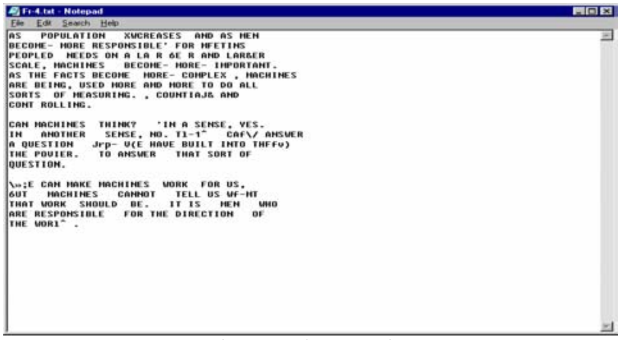
Fig. 5. Sample handwritten text to test the system

Table 1 shows the comparison results shown by five systems. FR (Fine Reader for handwritten character recognition) is the leading commercial system used for optical character recognition. DME1 and DME2 are versions of the software implemented on the basis of DME implemented by formula (5) and (6), respectively. As can be seen the use of DME2 gave a noticeable improvement in the quality of recognition. Note that the performance index given in this table is calculated as the ratio of the number of correctly recognized characters to the number of all of the characters in the text.

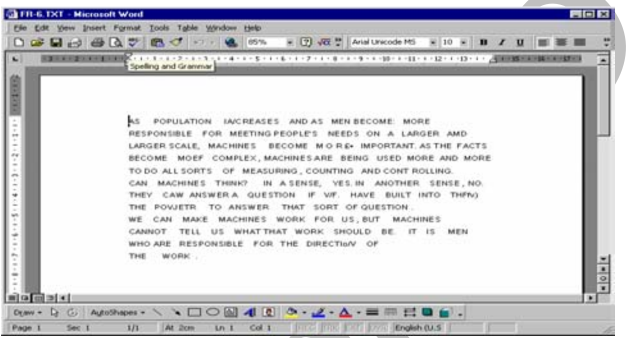
Table 1. Comparison of Performance of Three Software Systems

Package	**Performance %**
Fine Reader Version 4	93.9
Fine Reader Version 6	95.8
HP Precision Scan	93,3
Uncon DME1	95.04
Uncon DME2	97.2

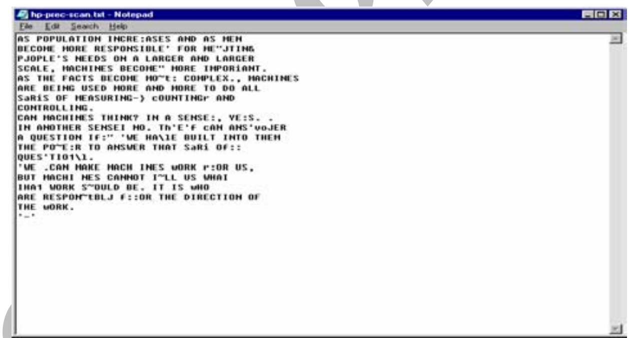
Also take into account that in the fuzzy rule base there was used only one rule for every character class. Increasing the number of rules would give further improvement in quality.



a. Fine Reader Version 4



b. Fine Reader Version 6



c. HP Precision Scan

Figure 6. Recognition results from three application software packages imported to text editors

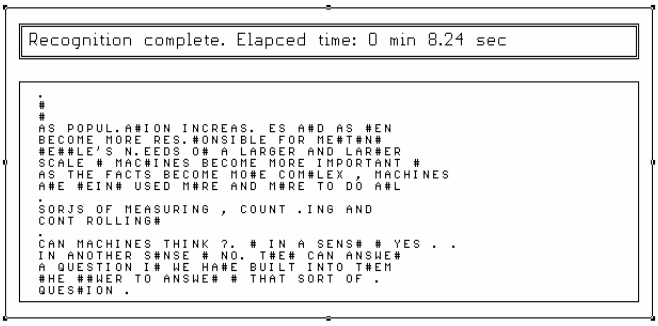


Fig. 7. Handwritten recognition system's output screen

5. Conclusion

The researches of various classification techniques and practical experiments show that the quality of recognition of fuzzy graphic objects can be improved by choosing appropriate feature aggregation method or a DME by other words. The suggested Neuro-Fuzzy Object Classifier with modified DME has shown better quality of recognition compared with existing application systems when applied to handwritten text.

REFERENCES

[1] J. Bezdek and S. Pal (ed.), *Fuzzy models for pattern recognition,* New York: IEEE Press (1992).

[2] X. Ye, C.Suen and M.Cheriet, *A generic system to extract and clean handwritten data from business forms*, in Prof. Int. Workshop on Frontiers in handwriting Recognition, Amsterdam (2000) 63-72.

[3] R. A. Aliev and R. R. Aliev, *Soft Computing and its applications*, World Scientific Publishing Co. Pte. Ltd (2001) p. 444.

[4] K. Saastamoinen, V. Könönen, and P. Luukka, *A classifier based on the fuzzy similarity in the Lukasiewicz structure with different metrics*, in proceedings of IEEE International Conference on Fuzzy Systems, FUZZ-IEEE'02, Vol. **1** (2002) 363-367.

[5] H. Bandemer and W. Näther, *Fuzzy data analysis*, Theory and Decision Library, Series B: Mathematical and Statistical Methods, Vol. **20**, Cluwer Academic Publishers (1992) 67-71.

[6] L. Mascarilla and C. Frélicot, *Combining rejection-based pattern classifiers*, in 19th International Conference of the North American Fuzzy Information Processing Society – NAFIPS, PeachFuzz 2000 (2000) 114-118.

[7] R Aliev and B. Guirimov, *Handwritten image recognition by using neural and fuzzy approaches*, Intelligent Control and Decision Making Systems, No. **1**, Thematic Collected Articles, Baku, Publishing House of Azerb. State Oil Academy (1997) 3-7.

[8] R. Aliev, B. Guirimov, K. Bonfig, and Steinmann, *A neuro-fuzzy algorithm for recognition of non-stylized images*, in proceedings of Fourth International Conference on Application of Fuzzy Systems and Soft Computing, ICAFS'2000, Siegen, Germany (2000) 238-241.

[9] S. Halgamuge and M. Glesner, *A fuzzy neural approach for pattern classification with generation of rules based on supervised learning*, in proceedings of Nuro Nimes 92 (1992) 165-173.

[10] V. Uebele, S. Abe, and M. Lan, *A neural - network based fuzzy classifier*, IEEE Transactions on Systems, Man, and Cybernetics, Vol. **23,** No. **3** (1995) 353-361.

[11] R. Yager, *A general approach to rule aggregation in fuzzy logic control*, Appl. Intelligence, **2** (1992) 333-351.

[12] R. Yager, *On a general class of fuzzy connectives*, Fuzzy Sets and Systems, **4** (1980) 235-242.

R. A. ALIEV*, MEMBER IEEE, DEPARTMENT OF COMPUTER-AIDED CONTROL SYSTEMS, AZERBAIJAN STATE OIL ACADEMY, BAKU, AZERBAIJAN
E-mail address: **raliev@iatp.az**

B. G. GUIRIMOV, DEPARTMENT OF COMPUTER-AIDED CONTROL SYSTEMS, AZERBAIJAN STATE OIL ACADEMY, BAKU, AZERBAIJAN
E-mail address: **guirimov@hotmail.com**

R. R. ALIEV, EASTERN MEDITERRANEAN UNIVERSITY, NORTH CYPRUS
E-mail address: **rashad.aliyev@emu.edu.tr**

* CORRESPONDING AUTHOR