On-line Handwriting Recognition with a Neuro-Fuzzy Method

Sung-Bae Cho

ATR Human Information Processing Research Laboratories
2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-02, Japan
Tel: +81-7749-5-1076 Fax: +81-7749-5-1008 E-mail: sbcho@hip.atr.co.jp

***Abstract*— This paper describes an efficient neuro-fuzzy method for recognizing on-line handwriting characters. The basic idea is to train a number of network classifiers and aggregating them with fuzzy logic. The method combines the outputs of separate networks with importance of each network, which is subjectively assigned as the nature of fuzzy logic. We demonstrate the superior performance of the presented method and compare with conventional methods like voting and averaging by thorough experiments on a difficult on-line handwriting recognition problem.**

I. Introduction

On-line handwriting recognition means that the machine recognizes the writing while the user writes. It naturally requires a transducer that captures the writing as it is written. Due to the recent hardware advance the interest in on-line handwriting recognition was renewed, and there have been proposed quite many methodologies.

Some recognition methods rely on prior analysis of the characters of the alphabet. Features (ascenders, cusps, closures, etc.) can be alphabet specific. Sequences of coded zones can also be alphabet specific if the zones are chosen based on properties of the alphabet. Other methods, such as most of the signal processing ones, are essentially independent of the alphabet. These methods are well introduced in the recent survey made by Tappert *et al* [1].

In this paper, we present another novel method based on neuro-fuzzy mechanism and show the results of extensive experiments. This method produces final decision from several neural network modules aggregated by fuzzy logic, especially fuzzy integral proposed by Sugeno [2]. In the fuzzy integral both objective evidence supplied by various sources and the expected worth of subsets of these sources are considered in the fusion process; It combines objective evidence for a hypothesis with the system's expectation of the importance of that evidence to the hypothesis. This approach may provide a possibility for incorporating any *a priori* knowledge regarding the underlying problem to improve the ability of the networks to generalize.

The rest of this paper is organized as follows. Section 2 introduces the overview of the on-line handwriting recognition. In Section 3, we present the proposed method based on the fuzzy integral. Explained in Section 4 are results of the preliminary experiments. Finally, Section 5 discusses the summary of the paper and the future research.

II. Handwriting recognition

In this paper, handwriting characters are inputted to the computer (SUN workstation) by an LCD tablet of of Photron FIOS-6440 which samples 80 dots per second. The tasks are to classify the Arabic numerals, the uppercase letters, and the lowercase letters which were collected from 13 writers. The writers are told to draw the numerals and letters into prepared square boxes in order to facilitate segmentation.

An input character consists of a set of strokes, each of which begins with a pen-down movement and ends with pen-up movements. Several preprocessing algorithms were applied to successive data points in a stroke to reduce quantization noises and fluctuations of the writer's pen motion. The processes used are as follows: wild point reduction, dot reduction, hook analysis, three point smoothing, peak preserving filtering, and N point normalization.

Wild point reduction can replace or eliminate an occasional spurious point, usually caused by a hardware problem. Dot reduction reduces dots to single points. Hook analysis algorithm eliminate hooks that can occur at the beginning, but more frequently at the end of strokes. Smoothing averages a point with its neighbors which are two neighbors in this case. Filtering eliminates duplicate data points and reduces the number of points. Size normalization adjusts the character size to a standard.

A sequence of preprocessed data points is approximated by a sequence of 8-directional straight-line segments [3]. The procedure for collecting handwriting data is schematically presented in figure 1.

III. Neuro-Fuzzy Recognizer

A. Overall Structure

The basic idea of the presented scheme is to develop n independently trained neural networks with relevant

0-7803-2461-7/95/$4.00 © 1995 IEEE

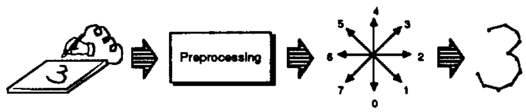

Figure 1: Schematic diagram of the process for data collection.

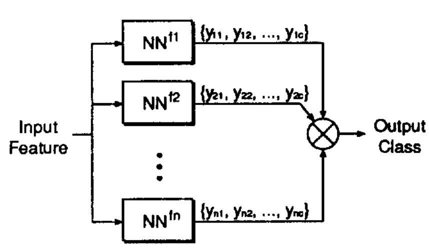

Figure 2: The multiple network architecture with fusion method.

features, and to classify a given input pattern by utilizing combination methods to decide the collective classification [4, 5] (see Fig. 2). Then it naturally raises the question of obtaining a consensus on the results of each individual network or expert. We can consider two kinds of methods based on voting technique and fusion technique.

Voting technique: Several literature have reported the usefulness of the voting procedures in classification area. The methods based on voting technique consider the result of each network as an expert judgement. A variety of voting procedures can be adopted from group decision making theory: unanimity, majority, plurality, Borda count, and so on. In particular, we will introduce the two of them: majority voting and Borda count.

The majority voting rule chooses the classification made by more than half the networks. When there is no agreement among more than half the networks, the result is considered an error. To appreciate the network performance, let's assume that all neural networks arrive at the correct classification with a certain likelihood $1-p$ and that they make independent errors. The chances of seeing exactly k errors among n copies of the network is then

$$\binom{n}{k} p^k (1-p)^{n-k} \tag{1}$$

which gives the following likelihood of the majority rule being in error

$$\sum_{k>n/2}^{n} \binom{n}{k} p^k (1-p)^{n-k}. \tag{2}$$

It can be shown by induction for odd n (or separately for even n) that provided $p < 1/2$, (2) is monotonically decreasing in n. In other words, if each network can get the correct answer more than half the time, and if network responses are independent, then the more networks used, the less the likelihood of an error by a majority decision rule. In the limit of infinite n, the coordinated error rate goes to zero.

For any particular class c, the Borda count is the sum of the number of classes ranked below c by each network; Let $B_j(c)$ be the number of classes ranked below the class c by the jth network. Then, the Borda count for class c is $B(c) = \sum_{j=1}^{n} B_j(c)$. The final decision is given by selecting the class label whose Borda count is the largest.

Fusion technique: Because the outputs of neural networks are estimates of Bayesian *a posteriori* probabilities [6], the classification of an input X is actually based on a set of real value measurements:

$$P(\omega_i|X), \quad 1 \le i \le c. \tag{3}$$

They represent the probabilities that X comes from each of the c classes under the condition X. In the multiple network scheme, each network k estimates by itself a set approximations of those true values as follows:

$$P_k(\omega_i|X), \quad 1 \le i \le c, \quad 1 \le k \le n. \tag{4}$$

One simple approach to combine the results on the same X by all n networks is to use the following average value as a new estimation of combined network:

$$P(\omega_i|X) = \frac{1}{n}\sum_{k=1}^{n} P_k(\omega_i|X), \quad 1 \le i \le c. \tag{5}$$

We can think of such a combined value as an averaged Bayes classifier. This estimation will be improved if we give the judge the ability to bias the outputs based on *a priori* knowledge about the reliability of the networks:

$$P(\omega_i|X) = \sum_{k=1}^{n} r_k P_k(\omega_i|X), \quad 1 \le i \le c, \tag{6}$$

$$\text{where} \sum_{k=1}^{n} r_k = 1. \tag{7}$$

Another alternative is to use the maximum value of $P_k(\omega_i|X)$ denoted by $P_m(\omega_i|X)$, to replace the correspondent average value. Since $\sum_{i=1}^{c} P_m(\omega_i|X) \neq 1$, we use the following normalized values as the new estimations:

$$P(\omega_i|X) = \frac{P_m(\omega_i|X)}{\sum_{j=1}^{c} P_m(\omega_j|X)}, \quad 1 \le i \le c. \quad (8)$$

B. Fuzzy Integral for Network Fusion

The fuzzy integral is a nonlinear functional that is defined with respect to a fuzzy measure, especially g_λ-fuzzy measure introduced by Sugeno [2]. The ability of the fuzzy integral to combine the results of multiple sources of information has been established in several previous works [7, 8, 9]. In the following we shall introduce some definitions of it and present an effective method for combining the outputs of multiple networks with regard to subjectively defined importances of individual networks. For further details on the method, see the recent publication made by the authors [10].

Definition 1: A set function $g : 2^Y \to [0,1]$ is called a fuzzy measure if

1) $g(\emptyset) = 0,\ g(Y) = 1,$
2) $g(A) \le g(B)$ if $A \subset B,$
3) If $\{A_i\}_{i=1}^{\infty}$ is an increasing sequence of measurable sets, then

$$\lim_{i\to\infty} g(A_i) = g(\lim_{i\to\infty} A_i).$$

From this definition, Sugeno introduced the g_λ-fuzzy measure satisfying the following additional property:

$$g(A \cup B) = g(A) + g(B) + \lambda g(A) g(B),$$

for all $A, B \subset X$ and $A \cap B = \emptyset$, and for some $\lambda > -1$.

Example 1: Consider the following case of $Y = \{y_1, y_2, y_3\}$ together with density values $g^1 = 0.34$, $g^2 = 0.32$, and $g^3 = 0.33$. Using the equation 14 (which will be introduced below), the Sugeno measure g must have a parameter λ satisfying $0.0359\lambda^2 + 0.3266\lambda - 0.001 = 0$. The unique root greater than -1 for this equation is $\lambda = 0.0305$, which produces the following fuzzy measure on the power set of Y:

Subset A	$g_{0.0305}(A)$
$\emptyset$	0
$\{y_1\}$	0.34
$\{y_2\}$	0.32
$\{y_3\}$	0.33
$\{y_1, y_2\}$	0.6633
$\{y_2, y_3\}$	0.6532
$\{y_1, y_3\}$	0.6734
$\{y_1, y_2, y_3\}$	1.0

As expected, the subset of criteria $\{y_1, y_3\}$ is more important for confirming the hypothesis than either subsets $\{y_1, y_2\}$ or $\{y_2, y_3\}$.

Definition 2: Let Y be a finite set and $h : Y \to [0,1]$ a fuzzy subset of Y. The fuzzy integral over Y of the function h with respect to a fuzzy measure g is defined by

$$\begin{aligned} h(y) \circ g(\cdot) &= \max_{E \subseteq Y} \left[\min\left(\min_{y\in E} h(y), g(E) \right) \right] \\ &= \max_{\alpha\in[0,1]} [\min(\alpha, g(F_\alpha))] \end{aligned} \quad (9)$$

where

$$F_\alpha = \{y | h(y) \ge \alpha\}. \quad (10)$$

To get some intuition for the fuzzy integral we consider the following interpretation. $h(y)$ measures the degree to which the concept h is satisfied by y. The term $\min_{y\in E} h(y)$ measures the degree to which the concept h is satisfied by all the elements in E. Moreover, the value $g(E)$ is a measure of the degree to which the subset of objects E satisfies the concept measured by g. Then, the value obtained from comparing these two quantities in terms of the min operator indicates the degree to which E satisfies both the criteria of the measure g and $\min_{y\in E} h(y)$. Finally, the max operation takes the biggest of these terms. One can interpret the fuzzy integral as finding the maximal grade of agreement between the objective evidence and expectation.

The calculation of the fuzzy integral when Y is a finite set is easily given. Let $Y = \{y_1, y_2, \ldots, y_n\}$ be a finite set and let $h : Y \to [0,1]$ be a function. Suppose $h(y_1) \ge h(y_2) \ge \ldots \ge h(y_n)$, (if not, Y is rearranged so that this relation holds). Then a fuzzy integral, e, with respect to a fuzzy measure g over Y can be computed by

$$e = \max_{i=1}^{n} [\min(h(y_i),\ g(A_i))] \quad (11)$$

where $A_i = \{y_1, y_2, \ldots, y_i\}$.

Note that when g is a g_λ-fuzzy measure, the values of $g(A_i)$ can be determined recursively as

$$\begin{aligned} g(A_1) &= g(\{y_1\}) = g^1 && (12) \\ g(A_i) &= g^i + g(A_{i-1}) + \lambda g^i g(A_{i-1}), \quad \text{for } 1 < i \le n && (13) \end{aligned}$$

λ is given by solving the equation

$$\lambda + 1 = \prod_{i=1}^{n} (1 + \lambda g^i) \quad (14)$$

where $\lambda \in (-1, +\infty)$, and $\lambda \ne 0$. This can be easily calculated by solving an $(n-1)$st degree polynomial and finding the unique root greater than -1. Thus the calculation of the fuzzy integral with respect to a g_λ-fuzzy measure would only require the knowledge of the density function, where ith density, g^i, is interpreted as the degree of importance of the source y_i towards the final evaluation. In this definition, we can easily see that if g^1 increases then $g(A_i)$ also increases and hence

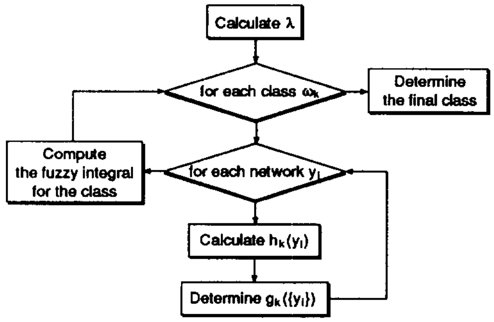

Figure 3: Algorithm of the network fusion by the fuzzy integral.

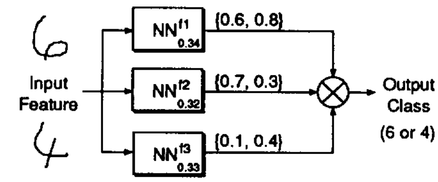

Figure 4: A simple example of network outputs for a two class problem.

the new fuzzy integral must be greater than or equal than the previous value. In general, if g^k increases for some $1 < k \leq n$, then $g(A_i)$, $i = 1, \ldots, k-1$, decreases and $g(A_i)$, $i > k$ increases. Therefore, if the function h or measure g increases then the integral increases.

Let $\Omega = \{\omega_1, \omega_2, \ldots, \omega_c\}$ be a set of classes of interest. Note that each ω_i may, in fact, be a set of classes by itself. Let $Y = \{y_1, y_2, \ldots, y_n\}$ be a set of neural networks, and A be the object under consideration for recognition. Let $h_k : Y \rightarrow [0,1]$ be the partial evaluation of the object A for class ω_k, that is, $h_k(y_i)$ is an indication of how certain we are in the classification of object A to be in class ω_k using the network y_i, where a 1 indicates absolute certainty that the object A is really in class ω_k and 0 implies absolute certainty that the object A is not in ω_k.

Corresponding to each y_i the degree of importance, g^i, of how important y_i is in the recognition of the class ω_k must be given. These densities can be subjectively assigned by an expert, or can be induced from data set. The g^i's define the fuzzy density mapping. Hence λ is calculated using (14) and thereby the g_λ-fuzzy measure, g, is constructed. Now, using (11) to (14), the fuzzy integral can be calculated. Finally, the class ω_k with the largest integral value is chosen as the output class. Fig. 3 illustrates the details of how the consensus is formed.

Example 2: Using the Example 1, how the consensus decision is performed by the fuzzy integral can now be described for a two class problem, which discriminates handwriting characters 6 and 4. Suppose that we obtain the network outputs for an input image as shown in Fig 4: $h(y_1) = 0.6$, $h(y_2) = 0.7$, and $h(y_3) = 0.1$, for class 1. For class 2, $h(y_1) = 0.8$, $h(y_2) = 0.3$, and $h(y_3) = 0.4$. The following table shows how the consensus is formed, where $H(E) = min(h(y_i), g(A_i))$.

Class	$h(y_i)$	$g(A_i)$	$H(E)$	max
1	0.7	$g(\{y_2\}) = 0.32$	0.32	
	0.6	$g(\{y_2, y_1\}) = 0.66$	0.6	√
	0.1	$g(\{y_2, y_1, y_3\}) = 1.0$	0.1	
2	0.8	$g(\{y_1\}) = 0.34$	0.34	
	0.4	$g(\{y_1, y_3\}) = 0.67$	0.4	√
	0.3	$g(\{y_1, y_3, y_2\}) = 1.0$	0.3	

Finally, the class 1 is selected as output. In the meantime, in case that we use the weighted average instead of the fuzzy integral, the class 2 is chosen as the correct class because the class 2 yields 0.5 ($0.34 \times 0.8 + 0.32 \times 0.3 + 0.33 \times 0.4$) whereas the class 1 produces 0.46 ($0.34 \times 0.6 + 0.32 \times 0.7 + 0.33 \times 0.1$). This example shows how the minute differences of g^i conspire so as to dramatically change the performance compared to simple averaging.

IV. Experimental Results

To evaluate the performance of the proposed method, we implemented three different networks, each of which is a two-layered neural network having a different number of input neurons and 20 hidden neurons. NN_1, NN_2 and NN_3 have 10, 15, and 20 input neurons, respectively. In each case, the network makes a decision based on its resolution. For example, NN_1 uses sparsely sampled inputs, and in doing so is able to overcome variations in input noise. NN_3, on comparison, uses a finer view of the input image. The selection of the features is largely adhoc and no attempt was made to find an optimal coding scheme although this is an important issue in character recognition schemes. Our objective here is to evaluate and compare different fusion methods through an example which has a certain complexity and practical significance.

Each of the three networks was trained by the EBP algorithm with 40 samples per class, validated with another 500 samples, and tested on 10 sets of additional samples collected from different 10 writers: The training process was stopped when the recognition rate over the validation set was optimized. This process and early stopping mechanism were adopted mainly for preventing networks from overtraining. The initial parameter values used for training were: Learning rate

Table 1: Fuzzy densities and the corresponding λ's.

Subject	g^1	g^2	g^3	λ
Number	0.3450	0.3349	0.3249	−0.0149
Upper	0.3447	0.3312	0.3240	0.0003
Lower	0.3370	0.3321	0.3312	−0.0009

is 0.4 and momentum parameter is 0.6. From the result of Eaton *et al.*, we have selected a value of the learning rate to be decreased gradually to allow stable convergence of training. An input vector is classified as belonging to the output class associated with the highest output activation. Each of the following experiments consisted of 10 trials in which the different data were made from different writers.

We assigned the fuzzy densities g^i, the degree of importance of each network, based on how good these networks performed on validation data, though there are a lot of possibilities to obtain these densities. We computed these values as follows:

$$g^i = \frac{p_i}{\sum_j p_j} \cdot d\mathrm{sum}, \tag{15}$$

where p_i is the performance of network NN_i for the validation data and $d$sum is the desired sum of fuzzy densities. The real values of these densities with the corresponding λ are shown in table 1. Table 2 reports the results of network fusion using the fuzzy integral on three different networks for numerals. In this table the value in the parentheses mean the confidence of the evaluation result. As can be seen, cases 2 and 3 were misclassified by NN_3 and NN_2, respectively. However, in the final evaluations they were correctly classified. In cases 5 and 17, one network with strong evidence overwhelmed the other networks, producing correct classification. Furthermore, in case 15, the fuzzy integral made a correct decision despite that the partial decisions from the individual neural networks were completely inconsistent. The effect of misclassification by the other networks has given rise to small fuzzy integral values for the correct classification in this case.

Table 3 shows the simulation results obtained on numerals, uppercase letters, and lowercase letters, respectively. All results are averaged over ten different sets of the data. In this table, NN_1 to NN_3 represent the three individual networks, and NN_{fi} a combined network of the three networks with the fuzzy integral. Although the network learned the training set almost perfectly in all three cases, the performances on the test sets are quite different.

A comparison of the proposed method with the conventional methods is given in table 4. As can be seen, the overall classification rates for the fuzzy integral were higher than those for the average, the weighted average, and the maximum method. In addition, the

Table 2: Results of network fusion using the fuzzy integral on three different networks for numerals.

D. I.	A. C.	Partial decision NN_1	NN_2	NN_3	Fuzzy decision
1	5	5 (0.98)	5 (0.89)	5 (0.99)	5 (0.95)
2	6	6 (0.99)	6 (0.99)	5 (0.33)	6 (0.68)
3	8	8 (0.99)	0 (0.00)	8 (0.99)	8 (0.66)
4	2	2 (0.99)	2 (0.99)	2 (0.99)	2 (0.99)
5	7	8 (0.01)	8 (0.00)	7 (0.96)	7 (0.32)
6	9	8 (0.00)	7 (0.11)	7 (0.47)	7 (0.19)
7	7	6 (0.01)	7 (0.99)	7 (0.99)	7 (0.66)
8	7	2 (0.13)	7 (0.96)	7 (0.00)	7 (0.32)
9	1	1 (0.99)	1 (0.99)	1 (0.99)	1 (0.99)
10	0	6 (0.71)	6 (0.40)	8 (0.80)	6 (0.37)
11	7	1 (0.17)	8 (0.00)	1 (0.00)	1 (0.06)
12	4	4 (0.99)	4 (0.99)	9 (0.99)	4 (0.66)
13	9	9 (0.99)	9 (0.99)	8 (0.07)	9 (0.66)
14	3	3 (0.99)	3 (0.99)	3 (0.99)	3 (0.99)
15	7	8 (0.03)	0 (0.04)	7 (0.48)	7 (0.16)
16	5	5 (0.93)	3 (0.13)	3 (0.62)	5 (0.32)
17	2	8 (0.34)	2 (0.99)	8 (0.20)	2 (0.33)
18	8	8 (0.98)	0 (0.96)	8 (0.68)	8 (0.83)
19	0	1 (0.03)	4 (0.00)	4 (0.00)	1 (0.01)
20	9	9 (0.75)	9 (0.37)	9 (0.95)	9 (0.69)
21	0	8 (0.80)	0 (0.99)	0 (0.99)	0 (0.66)
22	4	4 (0.99)	9 (0.91)	4 (0.98)	4 (0.70)
23	9	9 (0.99)	1 (0.99)	9 (0.99)	9 (0.67)
24	4	4 (0.99)	4 (0.99)	4 (0.99)	4 (0.99)
25	8	8 (0.99)	0 (0.98)	8 (0.83)	8 (0.62)

Table 3: Means and standard deviations of recognition rates of the three individual networks and the fuzzy integral (%).

	Numerals		Uppercases		Lowercases	
Net	Mean	S.D.	Mean	S.D.	Mean	S.D.
NN_1	82.69	6.36	68.10	8.95	73.95	7.73
NN_2	81.25	7.16	68.63	9.14	71.80	8.86
NN_3	81.05	7.15	66.26	10.60	72.15	9.30
NN_{fi}	88.11	7.14	76.16	9.85	80.35	7.24

integral evaluation need not sum to one, so that lack of evidence and negative evidence can be distinguished. It is also seen from table 4 that the mean recognition rate of the proposed method is higher than that of the other conventional methods. The following test, the paired t-test, can further support to determine whether the fuzzy integral method is superior to the conventional method or not.

For a given test problem, let f_i^a denote the solution at convergence for method a using test data i. To test whether methods a and b have the same mean solution

Table 4: Comparison of the fuzzy integral with other fusion methods on the average recognition rates (%).

Method	Number	Uppercase	Lowercase
Majority	84.90	74.00	74.60
Borda	86.80	77.40	79.10
Average	86.94	75.27	78.25
W. Avg.	87.06	75.58	78.75
Maximum	85.72	71.37	76.90
Fuzzy	88.11	77.60	80.35

value, we compute the following statistic:

$$t = \frac{\sqrt{n}\bar{x}}{\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}} \tag{16}$$

where $n = 10$, $x_i = f_i^a - f_i^b$, and $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$. (In this case the method b is of the fuzzy integral.) From this value we can reject the null hypothesis that $H_0 : \bar{x} \leq 0$ in favor of the alternative that $\bar{x} > 0$ with significance level α, where $\alpha = \Phi(t)$ and $\Phi(t)$ can be obtained from the table of percentage points of the t-distribution.

Since it follows t-distribution, an α point can be computed as the threshold t_α, where α could be 95, 99.95 or 99.9%, Then, if

$$|t| > t_\alpha \tag{17}$$

the null hypothesis is rejected at a $100\% - \alpha$ level of significance, i.e., the fuzzy integral method is superior to the conventional method. Otherwise, the null hypothesis is accepted, i.e., we cannot say the fuzzy integral method improves the performance significantly.

Table 5 shows the results of the test with $n = 10$ for all three tasks. In this comparison, f_i^b is of the fuzzy integral, and f_i^a of the average method as mentioned in the equation (5) or the weighted average method in (6). These methods were chosen for comparisons because they had produced the best results among the conventional methods mentioned in this paper. In this comparison, the degree of freedom is $(n-1) = 9$, and the threshold t_α with $\alpha = 95, 97.25$, and 99% is 1.833, 2.262 and 2.821, respectively. It is seen from table 5 that all the values of t, except for the weighted average of the uppercase letter task, are greater than t_α with $\alpha = 95\%$. Therefore, for all the cases except that, "no-improvement" hypothesis is rejected at a 5% level of significance. Similarly, other cases can be tested. This is a strong evidence that the proposed method is superior to the conventional methods.

Table 5: The paired t-test. (degree of freedom = 9, $t_{0.05} = 1.833$, $t_{0.025} = 2.262$, $t_{0.01} = 2.821$. "A" stands for Average method, and "WA" for Weighted Average. "Yes" indicates that the hypothesis is rejected for the task at the associated level of significance.)

Task	t	Significance Level		
		5 %	2.5 %	1 %
Numerals (A)	−2.908	Yes	Yes	Yes
Numerals (WA)	−2.575	Yes	Yes	No
Uppercase (A)	−2.193	Yes	No	No
Uppercase (WA)	−1.300	No	No	No
Lowercase (A)	−3.806	Yes	Yes	Yes
Lowercase (WA)	−3.361	Yes	Yes	Yes

V. Concluding Remarks

This paper presented a neuro-fuzzy method which is efficient for recognizing on-line handwriting character recognition. The experimental results show that it improves the generalization capability significantly. However, a serious theoretical investigation is needed as a further study for justifying the proposed method.

References

[1] C.C. Tappert, C.Y. Suen and T. Wakahara, "The state of the art in on-line handwriting recognition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **12** (8), 787–808, 1990.

[2] M. Sugeno, "Fuzzy measures and fuzzy integrals: A survey," *Fuzzy Automata and Decision Processes*, Amsterdam: North Holland, 89–102, 1977.

[3] H. Freeman, "Computer processing in line drawing images," *Computing Survey*, **6**, 57–98, March 1974.

[4] L.K. Hansen, and P. Salamon, "Neural network ensembles," *IEEE Trans. Pattern Analysis and Machine Intelligence*, **12**, 993–1001, 1990.

[5] S. Shlien, "Multiple binary decision tree classifiers," *Pattern Recognition*, **23**, 757–763, 1990.

[6] M.D. Richard, and R.P. Lippmann, "Neural network classifiers estimate Bayesian *a posteriori* probabilities," *Neural Computation*, **3**, 461–483, 1991.

[7] K. Leszeynski, P. Penczek, and W. Grochulskki, "Sugeno's fuzzy measures and fuzzy clustering," *Fuzzy Sets and Systems*, **15**, 147–158, 1985.

[8] H. Tahani, and J.M. Keller, "Information fusion in computer vision using the fuzzy integral," *IEEE Trans. on Systems, Man, and Cybernetics*, **20**, 733–741, 1990.

[9] R.R. Yager, "Element selection from a fuzzy subset using the fuzzy integral," *IEEE Trans. on Systems, Man, and Cybernetics*, **23**, 467–477, 1993.

[10] S.-B. Cho, and J.H. Kim, "Combining multiple neural networks by fuzzy integral for robust classification," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 25, no. 1, 1995.