Hybrid Fuzzy-Neural Systems in Handwritten Word Recognition

Jung-Hsien Chiang, *Member, IEEE*, and Paul D. Gader, *Member, IEEE*

***Abstract*—Two hybrid fuzzy neural systems are developed and applied to handwritten word recognition. The word recognition system requires a module that assigns character class membership values to segments of images of handwritten words. The module must accurately represent ambiguities between character classes and assign low membership values to a wide variety of noncharacter segments resulting from erroneous segmentations. Each hybrid is a cascaded system. The first stage of both is a self-organizing feature map (SOFM). The second stages map distances into membership values. The third stage of one system is a multilayer perceptron (MLP). The third stage of the other is a bank of Choquet fuzzy integrals (FI's). The two systems are compared individually and as a combination to the baseline system. The new systems each perform better than the baseline system. The MLP system slightly outperforms the FI system, but the combination of the two outperforms the individual systems with a small increase in computational cost over the MLP system. Recognition rates of over 92% are achieved with a lexicon set having average size of 100. Experiments were performed on a standard test set from the SUNY/USPS CD-ROM database.**

***Index Terms*—Fuzzy neural systems, word recognition.**

I. Introduction

The research described here is aimed at developing improved handwritten character classifiers for use in off-line handwritten word recognition. Our improvements are based on the observation that handwritten character classes are fuzzy sets. The goal is to improve word recognition rates by developing algorithms that can accurately compute character class memberships for both character and noncharacter input patterns.

A. Problem Description

Character segmentation is the process of isolating the individual characters in a digital image of handwriting. Ideally, a handwritten word recognition system would use a sequential process in which character segmentation was performed prior to character recognition. Unfortunately, as is often true in image pattern recognition, it is difficult to segment without recognizing and it is difficult to recognize without segmenting.

Manuscript received October 12, 1995; revised November 11, 1996. This work was supported in part by the U.S. Postal Service through the Environmental Research Institute of Michigan (ERIM).

J.-H. Chiang was with the Department of Computer Engineering and Computer Science, University of Missouri-Columbia, Columbia, MO 65211 USA. He is now with the Department of Information Management, Chaoyang Institute of Technology, Wufeng, Taiwan.

P. D. Gader is with the Department of Computer Engineering and Computer Science, University of Missouri-Columbia, Columbia, MO 65211 USA.

Publisher Item Identifier S 1063-6706(97)02838-5.

Fig. 1. Segmenting these images correctly requires word level context and recognition.

Fig. 1 depicts a classic problem. Both images are equally valid representations of "ll" and "u." The first interpretation requires a segmentation into two segments whereas the second requires only one segment. A system cannot ascertain which hypothetical segmentation is correct without using recognition and contextual information. This leads to the principle of having a lexicon driven system that includes tightly coupled segmentation and recognition [1]–[9].

One approach to tightly coupled segmentation and recognition is to generate an "over-segmentation." The image of the word is segmented into primitives. Each primitive is a subimage of the original word and ideally consists of either a character or a part of a character. The matching process computes the best way to assemble the primitives to represent a candidate string in a dictionary or lexicon. The match score between a segment and a character can be obtained by a character classifier. This approach provides a framework for solving the problems illustrated in Fig. 1 because an over-segmentation algorithm should divide both patterns into two pieces. If a dictionary string contains an "ll" then the pieces can be kept separate to match the pair of characters. If a dictionary string contains a "u" then the pieces can be combined to match the single character.

While over-segmentation provides a basis for a solution to the problem of ambiguity between single and multiple characters, it leaves several to be solved. When matching assemblies of primitives to a dictionary string, the system must process pieces of characters, multiple characters, or single characters with parts of other characters attached, etc. The system must be able to reject such assemblies as noncharacters while providing accurate class membership information for assemblies that do represent characters. Another problem not solved by over-segmentation is overlap between classes. A pixel pattern may represent many different characters depending on the context.

An example illustrating these problems is shown in Fig. 2. An erroneous match made by our baseline system of an image of the word "Cowlesville" to the lexicon string "Avenue" is shown. The second segment could be either "v" or "o"

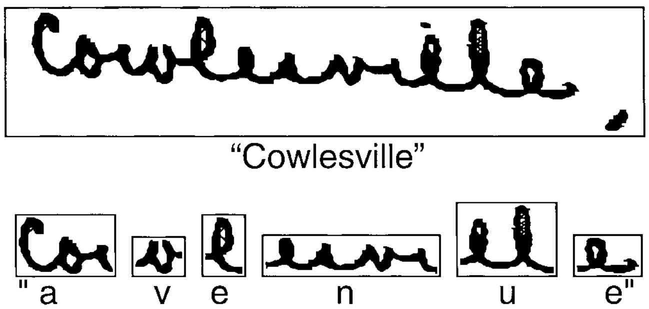

Fig. 2. Example of how character class overlaps and inadequate performance on noncharacter classes can cause erroneous word matches.

and the third segment could be either "e" or "l," illustrating the ambiguity between classes. As noted previously, the fifth segment in this figure could be "u." The first and fourth segments are assemblies of primitives that do not represent characters. However, our baseline system assigns fairly high-class membership values for matching to the classes "a" and "n," respectively, resulting in an erroneous overall match. The original research described here is aimed at solving these problems.

These examples show that handwritten character classes are not crisp sets; they are fuzzy sets and we treat them as such. Thus, instead of assigning a recognition score to each segment for each character class, we view the process as that of assigning a membership of each segment in each fuzzy character class.

B. Related Research

There are many hundreds if not thousands of papers on handwriting recognition. Traditional character recognition design and evaluation methods generally emphasize recognition rate. We have empirical evidence supporting the hypothesis that a high character recognition rate does not imply a high word recognition rate [4]. Some researchers have gone beyond the idea that character recognition rates on isolated characters are the only evaluation criteria and have considered the problem of rejecting noncharacters. Fogelman *et al*. devised a hybrid network for digit recognition consisting of a shared weight network for feature extraction followed by a radial basis function network for classification [10], [11]. They used their hybrid model to perform segmentation-free numeric field recognition. They compared their network structure to the shared-weight architecture of Le Cun *et al*. [12], [13] in terms of ability to reject white noise and alphabetic characters.

There are a variety of methods published for handwritten word recognition, mostly in the 1990's. They fall roughly into two classes: the lexicon-driven dynamic programming-based systems referred to earlier and hidden Markov model-based [14]–[16]. The baseline system described here is a dynamic programming-based system. Our word recognition rates are comparable to rates recently published by leading research groups (see Table V) and our unique methods could be used by these groups to improve their systems.

The general contribution of this work is the development and demonstration of novel methods for measuring and using fuzzy character class memberships within the context of handwritten word recognition. The novel ideas described here can be used to increase performance in other segmentation-based word recognition systems because we have shown how to improve a module that is common to all such systems. This research is relevant to fuzzy sets because it demonstrates that character recognition is not as important as character class membership estimation for handwritten word recognition. In doing so, we demonstrate methods for improving handwritten word recognition systems using fuzzy sets and also identify new practical problems in membership estimation for researchers in fuzzy sets.

In particular, the major contributions are:

1) design, implementation, and demonstration of a hybrid neural network model for character class membership assignment using a cascade of a Kohonen self-organizing feature map (SOFM) and a multilayer perceptron (MLP);
2) design, implementation, and demonstration of a hybrid neural network/fuzzy integral model for character class membership assignment using a cascade of a Kohonen SOFM and Choquet fuzzy integrals (FI's).

Both new models are used to compute fuzzy memberships of input patterns in handwritten character classes. Both have resulted in significant improvements in word-recognition performance and the word-recognition rates we achieve are comparable to recently published results by others, as shown in Section VIII.

We first discuss the variety of techniques involved in the word-recognition system. We then briefly describe the baseline word-recognition system, which has been fully described elsewhere [3]–[5], [14], [18], [19]. The three following sections discuss the two hybrid systems. Section IV discusses the general framework and the following two sections discuss the stages of the cascaded systems. Experiments are then discussed.

II. Basic Techniques and Choquet Fuzzy Integrals

Many techniques are used in the handwritten word-recognition system: image processing, mathematical morphology, feature extraction, MLP's, Kohonen SOFM, k nearest-neighbor algorithm, fuzzy sets, FI's, and dynamic programming. The emphasis in this paper is on the MLP, SOFM, and FI. The MLP and SOFM are well known. Since the FI technique is less well known, we describe it.

We only consider fuzzy measures and integrals on finite sets. Complete definitions can be found in [20]–[25]. Let $X = \{x_1, x_2, \cdots, x_n\}$ be a finite set and let $P(X)$ denote the family of all subsets of X. A set function g: $P(X) \rightarrow [0, 1]$ is called a fuzzy measure if

1) $g(\phi) = 0, g(X) = 1$;
2) if $A, B \in P(X)$ and $A \subset B$, then $g(A) \leq g(B)$.

Fuzzy measures are generalizations of probability measures that need not be additive. Since a fuzzy measure is not necessarily additive, the measure of the union of two disjoint subsets cannot be directly computed from the measures of the subsets, in general. Sugeno introduced a class of measures referred to as λ-fuzzy measures satisfying the property that for all $A, B \subset X$ with $A \cap B = \phi$

$$g(A \cup B) = g(A) + g(B) + \lambda g(A)g(B), \qquad \text{for some } \lambda > -1. \tag{1}$$

The value of λ can be found from $g(X) = 1$, which is easy to solve using Newton's method.

The densities of a fuzzy measure are defined to be the measures of the singleton sets and are denoted by $g^i = g(\{x_i\})$. They are extremely important in the case of λ-fuzzy measures as we now show. Let $A_i = \{x_i, x_{i+1}, \cdots, x_n\}$. When g is a λ-fuzzy measure, the values of $g(A_i)$ can be computed recursively as

$$g(A_n) = g(\{x_n\}) = g^n \tag{2}$$

$$g(A_i) = g^i + g(A_{i+1}) + \lambda g^i g(A_{i+1}), \qquad \text{for } 1 \le i < n. \tag{3}$$

Thus, a λ-fuzzy measure is completely determined by its densities.

Let h: $X \to [0, 1]$. The discrete Choquet FI can be computed as follows:

$$e = \sum_{i=1}^{n} [h(x_i) - h(x_{i-1})] g(A_i) \tag{4}$$

where X has been sorted so that $h(x_1) \le h(x_2) \le \cdots \le h(x_n)$ and $h(x_0) = 0$.

Define $\delta_i(g) = g(A_i) - g(A_{i+1})$. Then, the computation can also be represented by

$$e = \sum_{i=1}^{n} h(x_i)\delta_i(g) \tag{5}$$

where $A_{n+1} = \phi$. This is a nonlinear weighted sum since the indexes depend on the ordering of the values of the function h. If the measure g is additive, then the sum is independent of the ordering and is a linear weighted sum. From (2)–(4), we see that the calculation of the Choquet FI with respect to a λ-fuzzy measure only requires the knowledge of the fuzzy densities and the values $h(x_i), i = 1, 2, \cdots, n$.

Fuzzy measures and integrals represent the notion that "the whole is not equal to the sum of the parts." Fuzzy measures and integrals are useful because they provide increased flexibility over probability measures for aggregating partial information. This increased flexibility leads to more difficult design problems. For λ-fuzzy measures, this can be translated to the problem of finding optimal densities.

In this paper, one of our novel methodologies uses FI's to assign character class membership values to individual segments of images of handwritten words. Subsets of nodes of self-organizing feature maps will provide partial information toward the class membership of a segment. The importance of each node for a certain class will be represented by a density value. The scaled activation values at the nodes will be the values of the function h. These notions will be made precise in Section VII.

III. Baseline Handwritten Word Recognition System

The baseline system has been completely described elsewhere [14]. The new hybrid character class memberships assignment (CCMA) modules that are the focus of this paper are used as modules within the baseline system, replacing the existing (baseline) CCMA module. In the results section, we compare the results obtained using the hybrid CCMA modules to those obtained using the baseline CCMA module.

Fig. 3 depicts a block diagram of the baseline word recognition system and illustrates how the baseline CCMA module and the two new CCMA modules fit into the system. The inputs to the system are a binary image of a word and a lexicon of candidate strings. This system first segments the word image into primitives. A picture of the primitives found by our algorithm for a particular example is shown in Fig. 4. In the next stage, a match score is computed between each string in a lexicon and the word using the primitives. Match values between primitives or unions of primitives and characters are taken to be character class memberships values, which we denote by $\mu(S, C)$, where S denotes a primitive or union of primitives and C denotes a character class. Our focus in this paper is on improving the computation of these character class membership values; thus, our research involves investigating methods for computing extremely complex fuzzy set memberships for realistic problems in pattern recognition.

Estimating $\mu(S, C)$ is a difficult membership function estimation problem, because the domain is very high dimensional. The baseline system uses MLP's to implement $\mu(S, C)$. The MLP's are trained to produce fuzzy set memberships using a fuzzy k nearest-neighbor algorithm [4], [26]. Each network uses features extracted from a segment as input. There are two feature types: the bar and transition features, fully described in [4], [5], [19]. One feature type is 100 dimensional and the other is 120 dimensional.

There are four networks: for each feature type, there is one network for upper case characters and one for lower case characters. Each network has either 120 or 100 inputs and 28 outputs. The outputs include 26 outputs for character classes and two for noncharacter classes. One noncharacter class consists of segments of words that are larger than characters, e.g., two characters or pieces of two characters. The other noncharacter class consists of segments of words that are smaller than characters, e.g., pieces of characters. Each network has two hidden layers with 65 units in the first hidden layer and 39 in the second.

IV. New Character Class Membership Assignment Models

The hybrid fuzzy neural systems both use the three-stage framework shown in Fig. 5. There is one instance of each model for each feature type and for each case (upper and lower). The SOFM is used to create prototypes. Each prototype represents an allograph. An allograph is loosely defined as a shape that can be used to represent a character (e.g., cursive and printed styles of the same character). Each allograph is thought of as a fuzzy set. Allograph memberships are computed from distances to prototypes. These membership

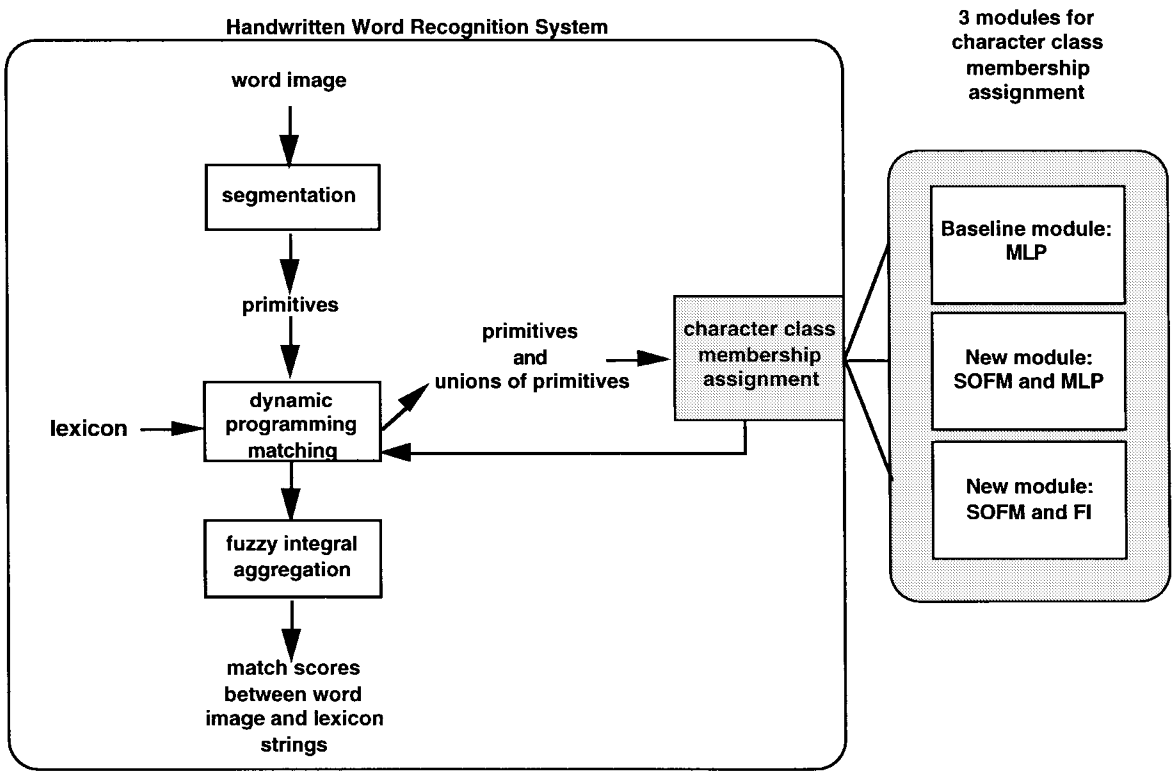

Fig. 3. Block diagram of the baseline word recognition system.

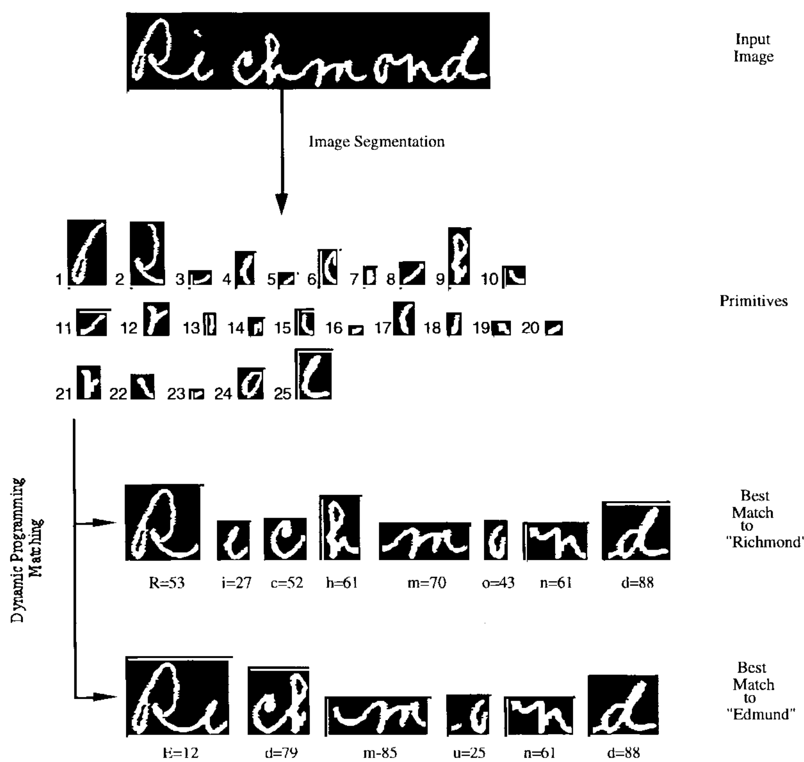

Fig. 4. Example of the segmentation approach to handwriting recognition.

values are stored in the SOFM. They are aggregated to compute character class memberships. Thus, the SOFM with allograph memberships is a feature set for character class membership assignment.

We investigated two models to implement this general framework: the SOFM/MLP and SOFM/FI. Both models use the same SOFM. The SOFM/MLP model uses sigmoid membership functions in the second stage while the SOFM/FI

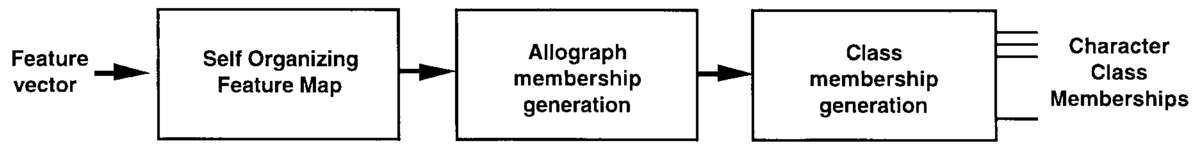

Fig. 5. Block diagram of general hybrid model for character confidence assignment.

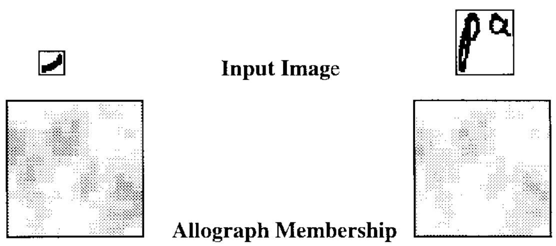

Fig. 6. SOFM generated allograph memberships provide a uniform representation of noncharacters.

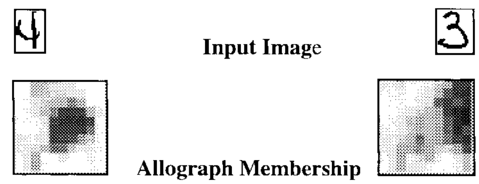

Fig. 7. Each digit class has a region or set of regions in this SOFM that characterizes the class.

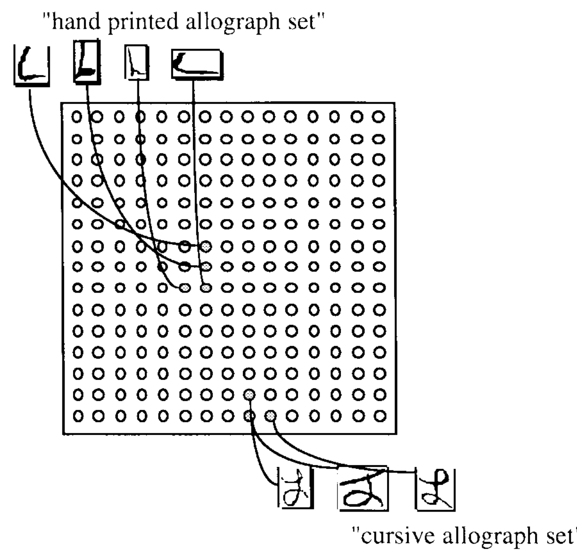

Fig. 8. An example of the feature map and the representation of allographs obtained from the class "L."

model uses inverse distance-based membership functions. We describe these in Section VI. In the third stage, the SOFM/MLP model uses a MLP to assign class memberships, whereas the SOFM/FI model uses a set of Choquet FI's. We describe the third stage in Section VII.

V. Self-Organizing Feature Maps

The SOFM's are trained using Kohonen's program available via anonymous FTP [27]. The inputs are either the bar or transition feature vectors. One aspect of our application is novel and critical to our success: the criteria used for selecting a map. We discuss the feature map selection criteria in Appendix A.

The SOFM provides a uniform representation of noncharacters. Most noncharacters are far from the prototypes. Thus, the activation values of the SOFM can be mapped into low membership values. This is different from most representations. For example, the transition features for the noncharacters in Fig. 6 are very different from one another, but the outputs from a SOFM trained with characters are similar. This standard representation makes it easier to classify them both as noncharacters.

Neighborhood information is useful for class membership estimation. Allograph membership generation can yield a "topologically ordered" array. Prototypes that are close in the array are generally similar. Thus, neighborhood information can be integrated to estimate class memberships, as described in Section VII. Furthermore, as discussed in [28], if suitable learning rate and neighborhood size are used the SOFM provides robust clustering that resists the influence of outliers.

An example is a SOFM trained with handwritten digits. The allograph memberships for two patterns are shown in Fig. 7. Each class has a region or set of regions which "fire" when that class is input.

VI. Allograph Memberships

We formalize the notion of allograph by defining an allograph to be a fuzzy set associated with a node in a feature map trained with handwritten characters. A collection of allographs is referred to as an allograph set.

An example is shown in Fig. 8. All training patterns from class "L" were presented to the map and nodes that won at least once were identified. Those nodes are highlighted in Fig. 8. Images picked arbitrarily from the training patterns that won at each node are shown. In this example, the class "L" has two allograph sets—the "hand printed" set, with four allographs and the "cursive" set with three allographs.

In the SOFM/MLP, allograph membership values are computed using sigmoid functions of the form

$$f(d;\alpha,\beta) = \frac{1}{1+e^{\alpha(d-\beta)}} \tag{6}$$

where d is the distance between a prototype and an input pattern and are used to compute allograph memberships. The scaling parameters α and β are found using gradient descent.

The SOFM/FI uses the following formula to compute the membership of input pattern S_k in allograph i:

$$h(S_k, i) = \frac{1}{1 + \left(\frac{d_{ik}}{\eta_i}\right)} \tag{7}$$

where η_i is the average distance between all of the winning patterns in the ith cluster and the ith cluster center v_i and d_{ik} is the distance between pattern S_k and v_i.

VII. Character Class Memberships via the SOFM/MLP and SOFM/FI Models

In this section, we describe the third stage of each system. The third stage of the SOFM/MLP system is a fully connected MLP that uses the topologically ordered array of allograph membership values as input features. This method does not use the topological ordering of the array directly (the topological ordering appears to be related to the robustness of the prototype generation process). The rationale for using MLP's is that they have performed well for character recognition applications already and should perform even better with the more standard representation afforded by the allograph memberships as feature inputs. The specific architecture and learning parameters that we used in our experiments is described in Section VIII.

The third stage of the SOFM/FI system is less straightforward to describe (although more computationally efficient) but does use the topological ordering of the SOFM directly. The general idea is to identify allograph sets as connected regions in the SOFM for each character class. A fuzzy measure is then defined on each allograph set. The character class memberships for an input pattern are estimated by computing a Choquet integral over each allograph set and taking the max of the integral values.

The feature map is the only trainable part of this system. An off-line design process is used to construct the domains of integration (allograph sets) and associated fuzzy measures for each class. For each character class we first identify core regions by thresholding the sum of the allograph membership values over that class in the feature map. The result of thresholding can be viewed as a binary image. The core regions are defined to be the connected components in the binary images. As a result of the spatial organization of the SOFM, connected core nodes represent similar patterns in feature space. We further divide the connected components into connected subregions. These subregions are allograph sets; they are used to define fuzzy measures and domains of integration for FI's using a methodology described later in this section.

There are four main reasons for considering this model.

1) By integrating over localized regions in the feature map, we take advantage of the spatial organization.
2) The Choquet integral can provide a degree of robustness against outliers [29], which reduces the effect of isolated high confidence values in an allograph set.
3) This membership aggregation method is significantly more computationally efficient than the MLP. We will present a comparison of the number of operations required for each technique in Appendix B.
4) The two methods for aggregating the information from the map (i.e., the MLP's and FI's) are complementary; they make different mistakes and, therefore, can be used jointly to increase the overall recognition rate. We will provide empirical results supporting this claim.

A. Design Methodology for the SOFM/FI Model

The first steps of the design are to train a SOFM and compute the parameters of the allograph membership functions. The allograph sets and fuzzy measures are then defined using a three step process: finding core regions for each class, defining the allograph sets, and defining a λ-fuzzy measure for each allograph set.

Step A—Finding the Core Regions: The procedure to determine the core nodes is as follows.

Let sum (i, C) represent the sum of the allograph membership values at node i for class C

$$sum(i, C) = \sum_{S_k \in C} \frac{1}{1 + \left(\frac{d_{ik}}{\eta_i}\right)} \tag{8}$$

and let m_i represent the mean of the allograph membership values at node i over all classes

$For\ class\ C = 1, 2, \cdots, 26\ Do$
    $For\ node\ i = 1, 2, \cdots, n\ Do$
        $If\ sum(i,C) \geq 2\ m_i\ then$
            $label\ node\ i\ as\ a\ core\ node$
            $for\ class\ C.$
        $Endif$

The threshold $2m_i$ was heuristically determined. The core regions are connected components of core nodes.

Step B—Defining the Allograph Sets by Refining Core Regions: We have observed that allograph sets should be roughly the same size for this method. Therefore, we include steps in our design process to increase the size of core regions consisting of a single node and to divide the large core regions into smaller regions. Core regions consisting of a single node are called singular regions. We expand singular regions as follows:

$For\ each\ class\ C = 1, 2, \cdots, 26\ Do$
    $For\ each\ singular\ region\ associated\ with$
    $class\ C,$
        $let\ i\ be\ the\ node\ in\ the\ singular\ region$
        Do
        $For\ each 4\text{-}neighbor\ j\ of\ i\ Do$
            $If\ act_j^c >= 1.5\ m_i\ then$
                $add\ j\ to\ the\ singular$
                $region.$
            $Endif$

The intuition behind this algorithm is that if a node is isolated, we can relax the threshold from $2m_i$ to $1.5m_i$ at the neighbors of the node to try to create a larger region.

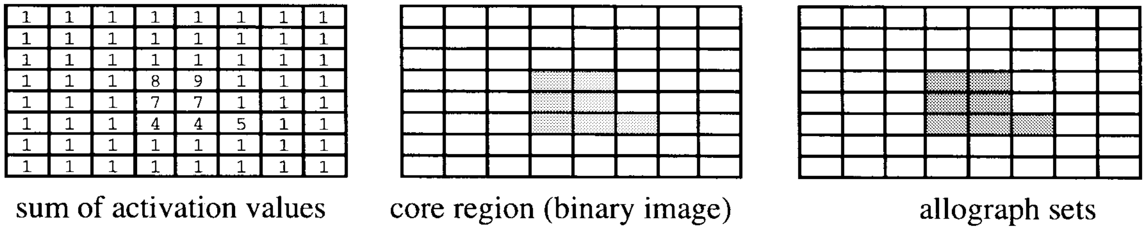

Fig. 9. Illustration of the design process using the 8 × 8 feature map example.

We divide large regions into smaller regions using similarity of activation values. This is similar to histogram-based thresholding of gray-level images, but in this case was performed by visual inspection. An illustration of dividing an artificial 8 × 8 feature map is shown in Fig. 9. The numbers shown on the left map are sums of the activation values. The means of the activation values are assumed to be 2.0 across the whole map. The core region was identified by *Step A* and is shown in the middle map (shaded area). In *Step B*, the allograph sets were defined by refining the core region, which are shown in the right map.

Step C—Defining A λ-Fuzzy Measure for Each Allograph Set: For each class C, let $A_C = \{X | X$ be an allograph set for class $C\}$. For each allograph set $X \in A_C$, we define a λ-fuzzy measure $g_{X,C}$. Let

- N_c^i number of times class C wins at node i;
- N^i number of times node i wins;
- B_c^i number of nodes at which class C wins at least once and that are four neighbors of node i.

The density values for X are defined as follows:

$$\begin{aligned} &For\ class\ C = 1, 2, \cdots, 26\ Do \\ &\qquad For\ each\ allograph\ set\ X \in A_C \\ &\qquad\qquad For\ each\ allograph\ i \in X \end{aligned}$$

$$g^i_{X,C} = \frac{N^i_C}{N^i \cdot B^i_c}. \tag{9}$$

The density value of a node reflects the degree of importance of the node in terms of relative winning frequency. As with most applications of FI's, the system is sensitive to the choice of densities.

These design steps are performed off line. The steps required to compute the character class memberships of an input pattern once the system is designed are described in the next section.

B. Calculating Character Class Memberships with the SOFM/FI Model

We show how to compute the memberships in class C for an input pattern S. For each allograph set $X \in A_C$, let x_i denote the ith node of X and $h_X(S, x_i)$ be the allograph membership value at node x_i, and let $g_{X,C}$ denote the λ-fuzzy measure defined on X by the densities $g^i_{X,C}$. The character class membership of S in $C, \mu(S, C)$ is defined to be the maximum Choquet integral value over all allograph sets in A_c, i.e.,

$$\mu(S, C) = \max_{X \in A_C} \left\{ \sum_{x_i \in X} h_A(S, x_i) \delta_i(g_{X,C}) \right\} \tag{10}$$

where the values $h_X(S, x_1) \leq h_X(S, x_2) \leq \cdots \leq h_X(S, x_n)$ are sorted allograph membership values and the $\delta_i(g_{X,C})$ are as defined in (5).

An illustration of the SOFM/FI character membership assignment is shown in Fig. 10. First, the distances between the input pattern and the prototypes are computed in the SOFM. The inverse distance mappings are then used to compute allograph memberships. For each allograph set, the Choquet integral values are computed from the allograph memberships and densities at the nodes labeled with integrals. Each class membership is the maximum Choquet integral value over all allograph sets for that class at the output nodes.

As an example, we show how to integrate the allograph memberships for character class "v" in Table I. There are two allograph sets for class "v" with three and two allographs, respectively. The class membership for class "v" is the max of the integral values obtained from the two allograph sets, which is 0.31.

TABLE I
AN EXAMPLE OF THE COMPUTATION OF THE CLASS MEMBERSHIPS USING CHOQUET FUZZY INTEGRAL

	Allograph	h(x)	$g^i(x)$	Choquet integral
Allograph set #1	v11	.32	.13	.31
	v12	.25	.17	
	v13	.41	.25	
Allograph set #2	v21	.24	.32	.25
	v22	.27	.12	

VIII. EXPERIMENTS

We first describe the data and then the experiments. We present character and word level results. Character level results are presented as character recognition rates. These results are of interest because they support the conjecture that character recognition rates are not indicators of word recognition performance. It is an open problem to find criteria for predicting the utility of a character classifier for word recognition.

We present word recognition rates for the baseline system and the two hybrid models. We also present results from combining these classifiers. Both hybrid systems significantly outperform the baseline system and the SOFM/MLP system outperforms the SOFM/FI system. The combination of the two new systems results in better performance than the individual modules. Although it is well known that multiple algorithms sometimes yield better results, the latter conclusion is interesting for two reasons: 1) both modules use the same features but different membership computation methods; this implies that each module is able to make use of different information that is available in the features and 2) the increase in computational

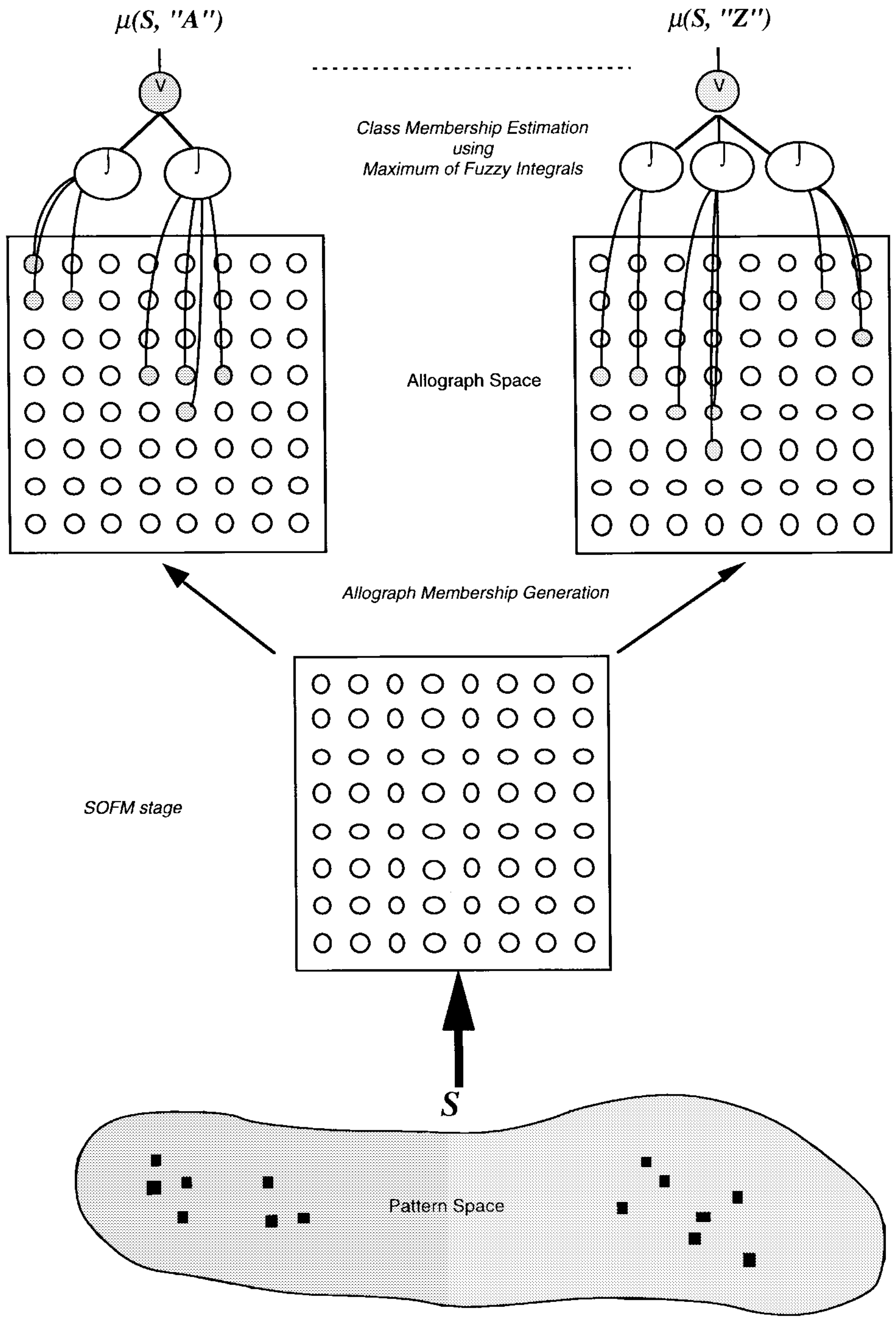

Fig. 10. Illustration of class membership assignment using the hybrid SOFM/FI model.

cost from using both modules over using just the SOFM/MLP system is very small.

A. Isolated Character Data

Isolated characters were extracted from handwritten word images which were extracted from images of addresses from U.S. postal service mail (they were extracted independently of this study). We used two data sets: the BHA and parts of the BD data sets, which are defined in [19]. The BHA data set is completely distinct from the BD data set. Ideally, we would use all BHA data (since the word images came from the BD data as described in the next section). However, some classes were poorly represented; there were over 900 “a’s” but fewer than 10 “j’s.” Poorly represented classes were augmented with characters from the BD data set. We generated training and testing sets with 250 characters per class each. For those classes with less than 250 total samples (using all the BD and BHA characters), we constructed sets of 250 per class by randomly resizing the existing characters. We refer to these data sets as the BHA250 training and testing data. These characters are unconstrained. Samples of noncharacters were also extracted from handwritten word images.

TABLE II
THE CHARACTER-RECOGNITION RATES FOR THE DIFFERENT CHARACTER CLASS MEMBERSHIP ASSIGNMENT MODULES

	Bar feature		Transition feature	
	UPPER CASE	lower case	UPPER CASE	lower case
BASELINE	79.6 %	75.4 %	81.2 %	79.2 %
SOFM/MLP	75.4 %	72.3 %	74.3 %	71.4 %
SOFM/FI	73.8 %	70.1 %	73.6 %	70.7 %

B. Handwritten Word Data

The handwritten word data base for our experiments came from the U.S. postal service sponsored *CD-ROM* image database produced by the Center of Excellence for Document Analysis and Recognition (CEDAR), State University of New York (SUNY) at Buffalo [30]. We used the 317 word BD city word test set for our tests. The testing images come with three lexicon sets. Each lexicon set was generated by a procedure designed to simulate a real address interpretation algorithm. The three lexicon sets have average lengths of 10, 100, and 1000, respectively. We present results for the lexicon sets with average lengths 100 and 1000.

C. Character Level Training and Testing—SOFM Construction

We constructed feature maps using isolated characters extracted from the BHA250 data sets. We used 50 character samples per class for training and 250 per class for feature map selection. The training set is referred to as the SOFM50 training set. The SOFM50 training set was extracted from the BHA250 training set using the SOFM algorithm as a clustering scheme. For each class, a 5×5 feature map was constructed from the BHA250 training set. The top two best matches from each of the 25 nodes were used as the elements of the SOFM50 training set. The BHA250 test set was used for feature map selection.

Feature maps of size 15×15 were generated from the transition and bar features. The maps were trained in two phases. The first phase is the "ordering phase." In this phase, the neighborhood size is initialized as the diameter of the map (15 here) and decreases to one during training. The parameters were the number of learning cycles: 5000; initial learning rate: 0.05; and neighborhood size: 15. In the second phase, the reference vectors are fine tuned. The neighborhood size is smaller and the learning cycle is longer. The parameters were number of learning cycles: 50 000; initial learning rate: 0.02; and neighborhood size: 5. After the feature maps construction, the parameters of the allograph membership functions were learned.

D. Character Level Training and Testing—Character Class Memberships

The MLP of the SOFM/MLP consists of 225 input units, 28 output units, and two hidden layers with 65 and 39 units. It was trained with back propagation using the transition and the bar features. We used the SOFM50 training set and the

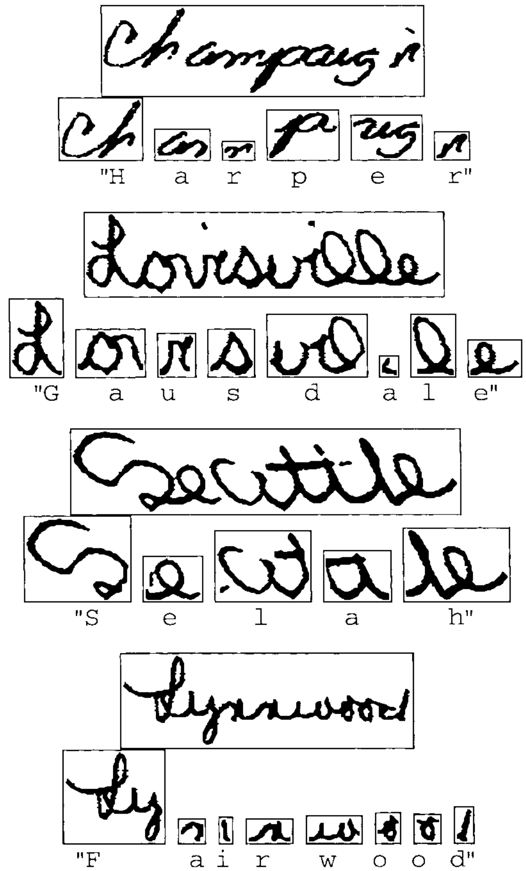

Fig. 11. Samples of word images that are incorrectly recognized by the baseline system, but correctly recognized by a hybrid model. The incorrect matches made by the baseline system are shown.

BHA250 test set. Both the transition feature and bar feature networks were trained on the same data with the same learning parameters using fuzzy desired outputs. Both were trained to an rms error rate of 0.07. The baseline model was also trained with the same learning parameters.

The design of the SOFM/FI was performed using the BHA250 training set. The indexes of the allograph sets were stored in a text file. The Choquet integral density values were computed off line from (7).

Character-recognition rates for the baseline system and hybrid models are shown in Table II. The recognition rate is the percentage of characters for which the highest output was associated with the true class. The noncharacter classes are not included in the computation of the recognition rate. We point out that the results for the baseline system are different from those reported in [5]. This is because our earlier results used a special-purpose board for training neural networks that is no longer available to us. Both hybrid models have lower character-recognition rates than the baseline but perform better on word recognition, as shown in the next section. Thus, higher character-recognition rates do not guarantee higher word-recognition rates.

E. Word-Recognition Results

In all word-recognition experiments, the baseline model is used to segment a word image into primitives, form unions

TABLE III
WORD-RECOGNITION TEST RESULTS VIA DIFFERENT CONFIDENCE ASSIGNMENT MODELS

	Recognition Rate for Ave Lexicon Length = 100	Recognition Rate for Ave Lexicon Length = 1000
BASELINE	79.8 %	61.2 %
SOFM/MLP	89.6 %	77.0 %
SOFM/FI	86.4 %	74.8 %

TABLE IV
WORD-RECOGNITION TEST RESULTS VIA THE BORDA COUNT COMBINATION OF WORD RECOGNIZERS

	Recognition Rate for Ave Lexicon Length = 100	Recognition Rate for Ave Lexicon Length = 1000
BASELINE + SOFM/MLP	89.0 %	76.7 %
BASELINE + SOFM/FI	87.4 %	73.5 %
SOFM/FI + SOFM/MLP	92.1 %	81.7 %

TABLE V
RECENT RESULTS FROM THE LITERATURE

Reference	Source and type of handwritten word data	Lex size	Recognition
Govindaraju et. al. [32]	USPS unconstrained	100	90%
Govindaraju et. al. [32]	USPS unconstrained	100	84%
Govindaraju et. al. [32]	USPS unconstrained	100	92% (Combination)
Govindaraju et. al. [32]	USPS unconstrained	1000	78%
Govindaraju et. al. [32]	USPS unconstrained	1000	68%
Govindaraju et. al. [32]	USPS unconstrained	1000	81% (Combination)
Kim & Govindaraju[1]	USPS unconstrained	100	87%
Kim & Govindaraju[1]	USPS unconstrained	1000	72%
Guillivec and Suen [17]	Check unconstrained	32	72%
Olivier [33]	Check unconstrained	27	70%
Aref et. al. [34]	local, 25 writers	40	85%
Yocoubi, et. al. [35]	French Postal Service (Constrained Cursive)	100	94%
Yocoubi, et. al. [35]	French Postal Service (Constrained Handprinted)	100	75%
Yocoubi, et. al. [35]	French Postal Service (Constrained Cursive)	1000	80%
Yocoubi, et. al. [35]	French Postal Service (Constrained Handprinted)	1000	52%
Cho & Kim [36]	USPS unconstrained	100	95%
Cho & Kim [36]	USPS unconstrained	1000	79%
Kimura & Shridhar [2]	USPS unconstrained	100	91%
Kimura & Shridhar [2]	USPS unconstrained	1000	85%

of primitives, and extract bar and transition features. The character class membership values are computed in three ways: the baseline neural networks, the SOFM/MLP, and the SOFM/FI. Dynamic programming and FI aggregation are then used to match to lexicon strings, as shown in Fig. 3.

The results of the individual systems are shown in Table III. The percentages represent the number of times the correct string was ranked highest of all the strings in the lexicon divided by 317. As can be seen, the hybrid SOFM/MLP and SOFM/FI models both achieved much higher recognition rates.

The Borda count is a common method for combining word-recognition outputs and is described in many places including [31]. The word-recognition test results obtained using the Borda count are shown in Table IV. The combination of the SOFM/MLP and the SOFM/FI achieves the best result. Note that combining the BASELINE and the SOFM/MLP does not increase performance. Therefore, it is not necessarily true that combination of algorithms yields higher performance.

Our handwritten word recognition results are among the best published results. Table V shows results reported recently in the literature. It is difficult to compare results on different data, but the table shows that are our results are competitive with the best in the literature.

Fig. 11 shows some word images that were incorrectly recognized by the baseline system but correctly recognized by either the SOFM/MLP or SOFM/FI. The segmentations of the incorrect matches are shown. Each segmentation contains noncharacters which no longer cause errors.

IX. CONCLUSION

We have shown that methods based on the notion that handwritten character classes are fuzzy sets can lead to high handwritten word-recognition rates. Although the approach

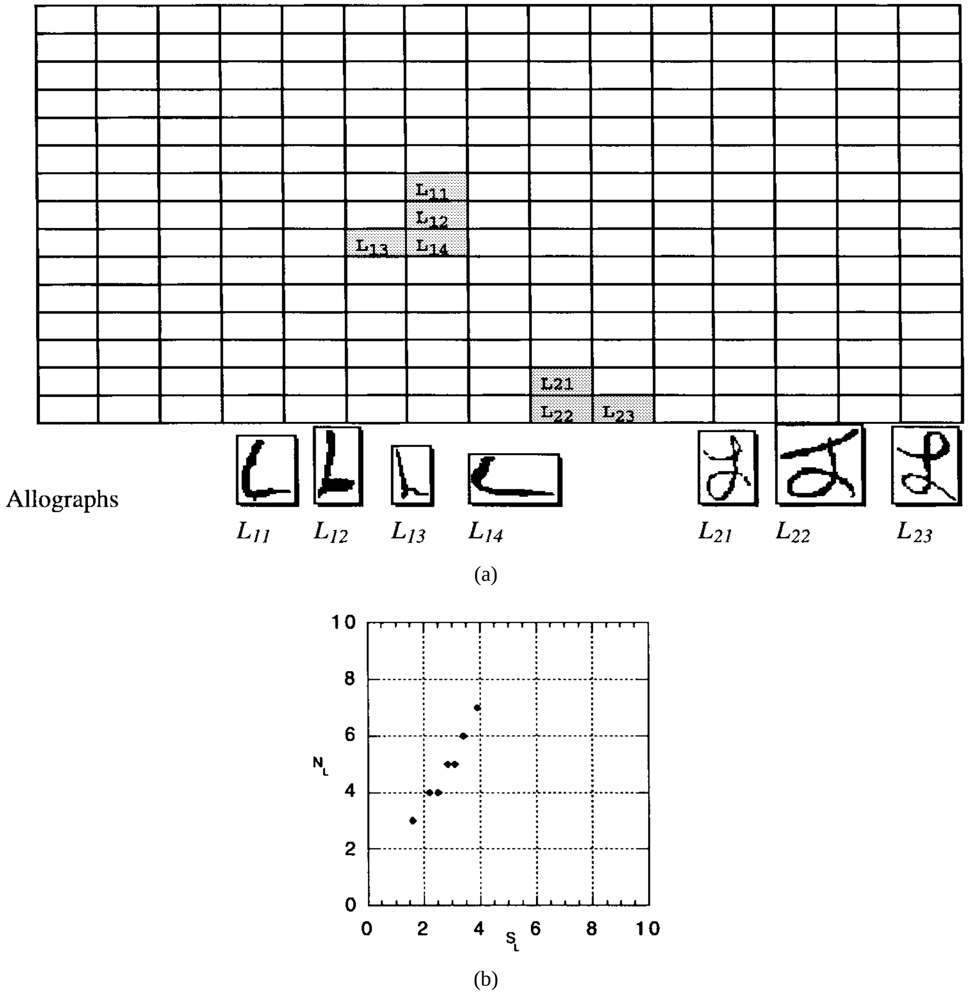

Fig. 12. (a) An example of a good feature map and two allograph sets for class "L." (b) The crisp and fuzzy sizes plot of the good feature map shown above.

based on SOFM/MLP may seem straightforward, it would be difficult to predict that one could achieve such a large increase in performance using this hybrid model. Furthermore, the success of the method depends upon our ability to create good feature maps, which we evaluated using our feature map-selection criteria.

Our experiments suggest and/or support several ideas. One is that clustering methods such as the SOFM, are useful for creating features that support rejection of noncharacters and accurate class-membership estimation in handwriting recognition systems. Other similar methods, e.g., fuzzy C-means or possibilistic clustering, may work as well or better.

Another idea is that high character-recognition rates do not imply high word-recognition rates. We believe that the important factor is accurate representation of all character-class memberships. Unfortunately, successful representations of handwritten characters are high dimensional. Thus, an important problem in fuzzy sets is how to accurately compute fuzzy set memberships of high-dimensional elements such as handwritten characters. These problems represent interesting challenges to the fuzzy set community.

We have also demonstrated a novel approach for using the Choquet FI to aggregate features computed by a SOFM that uses the spatial organization properties of the SOFM. Although the MLP approach achieves higher recognition rates, the FI provides complementary information with a very small increase in computational cost.

APPENDIX A
FEATURE MAP-SELECTION CRITERIA

The prototypes found by the SOFM algorithm depend on the order that the training patterns are presented, initial values of the reference vectors, the neighborhood kernel function, and the rate of decrease of neighborhood size. We used two measures for the quality of a feature map. For each class we construct the list of winning nodes. Each winning node represents a cluster for that class. We compute the membership of each pattern in the class in each cluster using (7). These memberships are used to define the two measures for each cluster C_i:

1) the crisp size N_i number of patterns in cluster C_i (patterns with $\mu(S_k, C_i) > \gamma$);
2) the fuzzy size $F_i = \Sigma_{S_k \in C_i} \ \mu(S_k, C_i)$ sum of the memberships in C_i for all patterns in cluster C_i (i.e., for all patterns $S_k \in C_i$ with $\mu(S_k, C_i) > \gamma$).

Ideally, $S_i = N_i$. We seek to minimize the deviation from the ideal while maintaining large clusters. Our use of

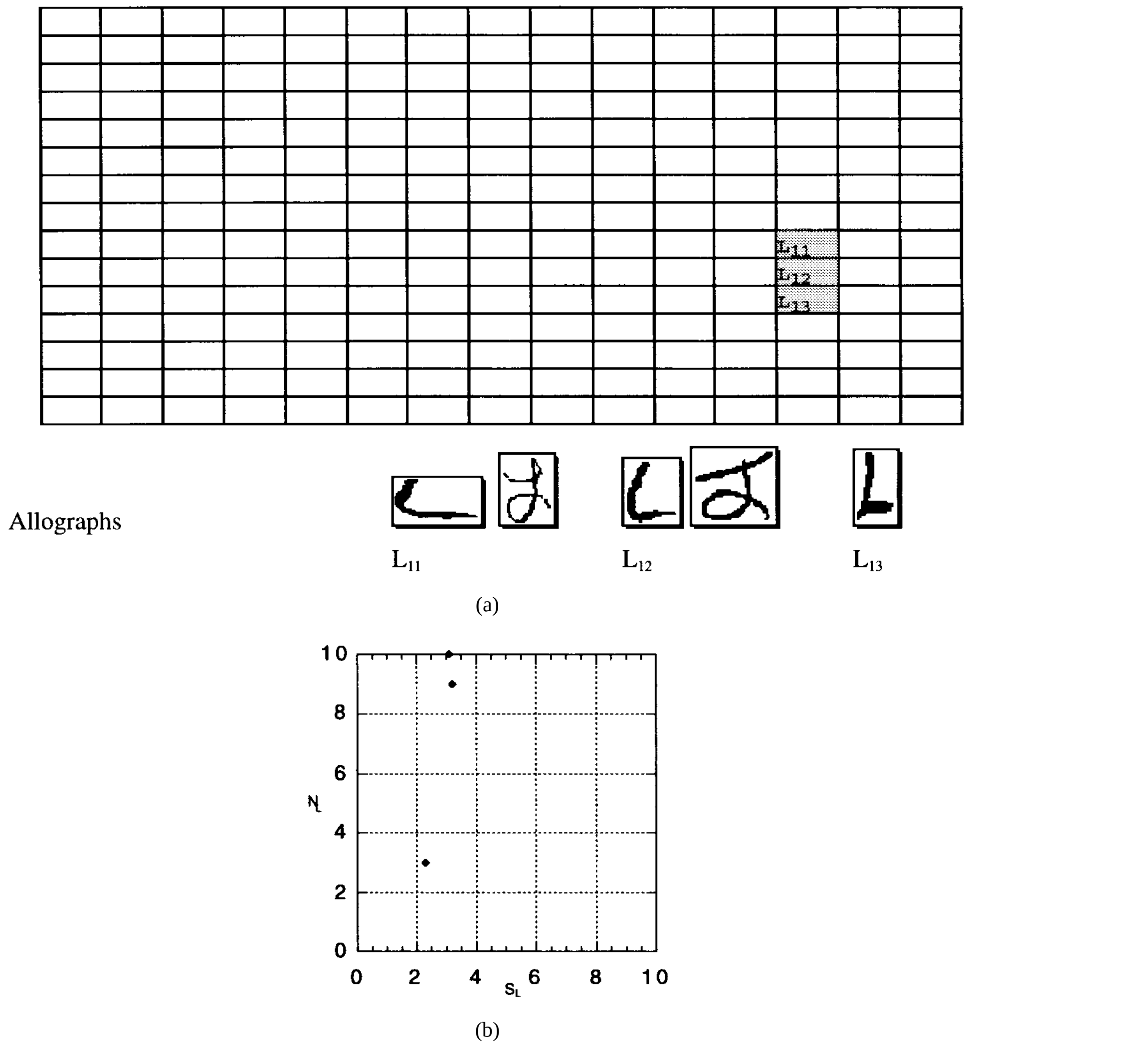

Fig. 13. (a) An example of a bad feature map and one allograph set from class "L." (b) The crisp and fuzzy sizes plot of the bad feature map shown above.

these measures is heuristic; we repeat this procedure for each class and choose a feature map according to the plots of the above measures. We can formalize our goal as follows. Let $\{(N_i, F_i) | i = 1, 2, \cdots w\}$ be the set of pairs of measures obtained from a class C. For each class we would like to

$$\text{Minimize} \quad \sum \frac{(N_i - F_i)}{\|(N_i, F_i)\|}. \tag{A.1}$$

We show two examples using class "L." Fig. 12(a) shows two allograph sets obtained from a good feature map. Each set contains allographs representing similar styles. The crisp and fuzzy sizes plot obtained from this feature map is shown in Fig. 12(b).

In contrast, Fig. 13 shows a bad feature map and the corresponding plots. The feature map has only one allograph set. The representatives of both allographs L_{11} and L_{12} are composed of different styles of patterns. Two of the allographs L_{11} and L_{12} have large crisp sizes but relative low membership values. The value of the objective function in (A.1) is 0.37 from the good map and 0.57 from the bad map.

Appendix B
Computational Analysis

In this section, we compare the complexity of the MLP and FI methods for computing character class memberships. We first consider the FI case; we consider the worse case; we consider comparisons and all arithmetic operations as equal. Let m be the maximum number of allograph sets for a class and n be the maximum number of elements in an allograph set. The calculation of a Choquet integral over an allograph set requires that we sort n numbers, compute n weights, and compute a weighted sum of n terms. Sorting requires $n \log_2 n$ steps. The computation of each weight requires two additions, one subtraction, and two multiplications, as given by (3) and (5), so $5n$ operations are required for a single allograph set. The weighted sum requires $2(n-1)$ operations. Since there are 26 classes, computing the character class memberships from a scaled SOFM using this approach requires at most

$$\begin{aligned} &26\, m\, [5n + 2(n-1) + n \log_2 n] \\ &\quad = 26\, m[n \log_2 n + 7n - 2] \text{ operations.} \end{aligned}$$

In our experiments, we have $m<4$ and $n<4$. Using $m = n = 4$ yields

$$\begin{aligned}26\, m[n \log_2 n + 7n - 2] &= 26 \times 4 \times (8 + 26)\\ &= 3536 \text{ operations.}\end{aligned}$$

The number of operations required for our MLP-based method, which uses a network with 225 input units, 28 output units, and two hidden layers with 65 and 39 units, and which has a bias at each noninput node is

$$\begin{aligned}&(225 \times 65) + (65 \times 39) + (39 \times 28) + 4(65 + 39 + 28)\\ &\quad = 18\,928 \text{ (operations).}\end{aligned}$$

It would require a much smaller MLP network for the MLP to require a number of operations similar to the FI. We have had difficulty training smaller networks to perform well.

ACKNOWLEDGMENT

The authors would like to thank several researchers who participated in the construction of the baseline system, including M. Mohamed, University of Missouri, Columbia, MO, and A. Gillies, D. Hepp, M. Ganzberger, M. Whalen, and others from the Environmental Research Institute of Michigan (ERIM), Ann Arbor. They would also like to thank Dr. J. Tan and C. O'Connor for their past support.

REFERENCES

[1] G. Kim and V. Govindaraju, "Handwritten word recognition for real-time applications," in *Proc. 3rd Int. Conf. Document Anal. Recognition*, Montreal, CA, Aug. 1995, pp. 24–28.

[2] F. Kimura, M. Shridhar, and N. Narasimhamurthi, "Lexicon directed segmentation-recognition procedure for unconstrained handwritten words," in *Proc. 3rd Int. Workshop Frontiers Handwriting Recognition*, Buffalo, NY, Nov. 1993, pp. 122–132.

[3] P. D. Gader, M. A. Mohamed, and J. M. Keller, "Dynamic programming based handwritten word recognition using the Choquet fuzzy integral as the match function," *J. Electron. Imaging*, vol. 5, no. 1, pp. 15–25, 1996.

[4] P. Gader, M. Mohamed, and J. Chiang, "Comparison of crisp and fuzzy character neural networks in handwritten word recognition," *IEEE Trans. Fuzzy Syst.*, vol. 3, pp. 357–364, Aug. 1995.

[5] ——, "Handwritten word recognition with character and inter-character neural networks," *IEEE Trans. Syst. Man Cybern.*, to be published.

[6] E. Lecolinet and J. Crettez, "A grapheme-based segmentation technique for cursive script recognition," in *Proc. 1st Int. Conf. Document Anal. Recognition*, Saint Malo, France, Sept. 1991, pp. 740–748.

[7] C. Nohl, C. Burges, and J. Ben, "Character-based handwritten address word recognition with lexicon," in *Proc. U.S. Postal Service Advanced Technol. Conf.*, Washington, DC, Nov. 1992, pp. 167–180.

[8] B. Plessis, "Isolated handwritten word recognition for contextual analysis reading," in *Proc. U.S. Postal Service Advanced Technol. Conf.*, Washington, DC, Nov. 1992, pp. 579–593.

[9] M. Shridhar and F. Kimura, "Handwritten address interpretation using word recognition with and without lexicon," in *Proc. IEEE Conf. Syst., Man, Cybern.*, Vancouver, BC, Canada, Oct. 1995, pp. 2341–2346.

[10] S. F. Fogelman and T. A. Sligos, "Integrating neural networks for real world applications," in *Int. Conf. Neural Networks IEEE Comput. Soc.*, J. Zurada, R. M. II, and C. Robinson, Eds., Orlando, FL, July 1994, pp. 396–405.

[11] S. F. Fogelman, B. Lamy, and E. Viennet, "Multi-modular neural network architectures for pattern recognition: Applications in optical character recognition and human face recognition," *Int. J. Pattern Recognition Artificial Intell.*, vol. 7, no. 4, pp. 721–756, 1993.

[12] Y. L. Cun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Back-propagation applied to handwritten zip code recognition," *Neural Computat.*, vol. 1, no. 4, pp. 541–551, 1989.

[13] Y. L. Cun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, L. D. Jackel, and H. Baird, "Constrained neural network for unconstrained handwritten digit recognition," in *Proc. 1st Int. Workshop Frontiers Handwriting Recognition*, Concordia Univ., Montreal CA, Apr. 1990, pp. 145–151.

[14] M. Mohamed and P. D. Gader, "Handwritten word recognition using segmentation-free hidden Markov modeling and segmentation-based dynamic programming techniques," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, no. 5, pp. 548–554, May 1995.

[15] A. Gillies, "Cursive word recognition using hidden Markov models," in *Proc. U.S. Postal Service Advanced Technol. Conf.*, Washington, DC, Nov. 1992, pp. 557–563.

[16] M. Y. Chen, A. Kundu, and J. Zhou, "Off-line handwritten word recognition using hidden Markov model type stochastic networks," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, pp. 481–496, May 1994.

[17] D. Guillevic and C. Y. Suen, "Cursive script recognition applied to the processing of bank cheques," in *Proc. 3rd Int. Conf. Document Anal. Recognition*, Montreal CA, Aug. 1995, pp. 15–19.

[18] P. D. Gader, M. P. Whalen, M. J. Ganzberger, and D. Hepp, "Hand-printed word recognition on a NIST data set," *Machine Vision Applicat.*, vol. 8, pp. 31–40, 1995.

[19] A. M. Gillies, D. Hepp, M. Ganzberger, R. Rovner, and P. D. Gader, "Handwritten address interpretation," Rep. ERIM U.S. Postal Service, Office Adv. Technol., 1993.

[20] M. Grabisch, H. T. Nguyen, and E. A. Walker, *Fundamentals of Uncertainty Calculi with Applications to Fuzzy Inference*. Dordrecht, Germany: Kluwer, 1995.

[21] M. Grabisch, "On equivalence classes of fuzzy connectives—The case of fuzzy integrals," *IEEE Trans. Fuzzy Syst.*, vol. 3, pp. 96–110, Jan. 1995.

[22] J. M. Keller, P. D. Gader, H. Tahani, J.-H. Chiang, and M. Mohamed, "Advances in fuzzy integration for pattern recognition," *Fuzzy Sets Syst.*, vol. 65, pp. 273–283, July 1994.

[23] T. Murofushi and M. Sugeno, "A theory of fuzzy measures: Representations, the Choquet integral, and null sets," *J. Math Anal. Applicat.*, vol. 159, pp. 532–549, 1991.

[24] M. Sugeno, "Fuzzy measures and fuzzy integrals—A survey," in *Fuzzy Automata and Decision Processes*, M. M. Gupta, G. N. Saridis, and B. R. Gaines, Eds. New York: North-Holland, 1977, pp. 89–102.

[25] H. Tahani and J. Keller, "Information fusion in computer vision using the fuzzy integral," *IEEE Trans. Syst., Man, Cybern.*, vol. 20, no. 3, pp. 733–741, May 1990.

[26] J. Keller, M. Gray, and J. Givens, "A Fuzzy K Nearest Neighbor Algorithm," *IEEE Trans. Syst., Man, Cybern.*, vol. 15, no. 4, pp. 580–585, July 1985.

[27] T. Kohonen, J. Kangas, and J. Laaksonen, "SOMPAK: The self-organizing map program package version 1.2," Helsinki Univ. Technol., Helsinki, Finland, Nov. 1992.

[28] J. Chiang, "Hybrid fuzzy neural systems for robust handwritten word recognition," Ph.D. dissertation, Univ. Missouri-Columbia, Columbia, MO, 1995.

[29] J. Keller and Z. Chen, "Learning in fuzzy neural networks utilizing additive hybrid operators," in *Proc. 2nd Int. Conf. Fuzzy Logic Neural Networks*, Iizuka, Japan, July 1992, pp. 85–87 (invited paper).

[30] J. J. Hull, "A database for handwritten text recognition research," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, pp. 550–554, May 1994.

[31] T. K. Ho, J. J. Hull, and S. N. Srihari, "Decision combination in multiple classifer systems," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, pp. 66–75, Jan. 1994.

[32] V. Govindaraju and S. N. Srihari, "System for reading handwritten documents," in *Proc. IEEE Conf. Syst., Man, Cybern.*, Vancouver, BC, Canada, Oct. 1995, pp. 2347–2352.

[33] C. Olivier, T. Paquet, M. Avila, and Y. Lecourtier, "Recognition of handwritten words using stochastic models," in *Proc. Int. Conf. Document Anal. Recognition*, Montreal, CA, Aug. 1995, pp. 19–23.

[34] W. G. Aref, P. Vallabhaneni, and D. Barbara, "On training hidden Markov models for recognizing handwritten text," in *Proc. 4th Int. Workshop Frontiers Handwriting Recognition*, Taipei, Taiwan, Dec. 1994, pp. 275–284.

[35] A. E. Yacoubi, J.-M. Bertille, and M. Gilloux, "Toward a more effective handwritten word recognition system," in *Proc. 4th Int. Workshop Frontiers Handwriting Recognition*, Taipei, Taiwan, Dec. 1994, pp. 378–386.

[36] W. Cho and J. Kim, "Off-line recognition of cursive words with network of hidden Markov models," in *Proc. 4th Int. Workshop Frontiers Handwriting Recognition*, Taipei, Taiwan, Dec. 1994, pp. 410–417.

Jung-Hsien Chiang (S'94–M'95) received the B.Sc. degree in electrical engineering from the National Taiwan Institute of Technology, Taiwan, in 1989, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Missouri, Columbia, in 1991 and 1995, respectively.

He was employed as a Senior Engineer by Chung-Hsin Elec. Inc., Taiwan, and as Researcher in the Computer and Communication Lab/Industrial Technology Research Institute, Taiwan. He is currently an Associate Professor in the Department of Information Management at the Chaoyang Institute of Technology, Taiwan. His current research interests include neural networks, fuzzy systems, pattern recognition, and the modeling of decision-making problems.

Paul D. Gader (M'87) received the Ph.D. degree from the University of Florida, Gainesville, FL, in 1986.

He is an Associate Professor in the Department of Computer Engineering and Computer Science at the University of Missouri-Columbia, Columbia, MO. Previously, he worked at the Environmental Research Institute of Michigan, Ann Arbor, and at Honeywell Systems and Research Center, Minneapolis, MN. He is an Associate Editor of the *Journal of Mathematical Imaging and Vision* and the *Journal of Electronic Imaging*. He has been active in handwriting recognition research since 1989 and has performed research on handwritten numeral, character, and word recognition as well as document image processing and field segmentation. He has also been involved in research of fuzzy pattern recognition and image processing, image algebra and morphological image processing, obstacle detection, medical image analysis, vehicle identification, scene analysis, and applied mathematics.

Dr. Gader co-chaired the SPIE Conference on Image Algebra and Morphological Image Processing from 1990 to 1994.